University of Crete
Department of Computer Science

FO.R.T.H.
Institute of Computer Science

Location-sensing using the IEEE 802.11 Infrastructure and the Peer-to-peer Paradigm for Mobile Computing Applications

(M.Sc. Thesis)

Anastasia Katranidou

Heraklion

February 2006

Department of Computer Science
University of Crete

Location-sensing using the IEEE 802.11 Infrastructure and the Peer-to-peer Paradigm for Mobile Computing Applications

Submitted to the
Department of Computer Science
in partial fulfillment of the requirements for the degree of
Master of Science

February 20, 2006

Author: _____
Anastasia Katranidou
Department of Computer Science

Board
of enquiry:

Supervisor _____
Maria Papadopouli
Assistant Professor

Member _____
Apostolos Traganitis
Professor

Member _____
Panagiotis Trahanias
Professor

Accepted by:

Chairman of the
Graduate Studies Committee _____
Dimitris Plexousakis
Associate Professor

Heraklion, February 2006

In my family

Στην οικογένειά μου

Location-sensing using the IEEE 802.11 Infrastructure and the Peer-to-peer Paradigm for Mobile Computing Applications

Anastasia Katranidou

M.Sc. Thesis

Department of Computer Science

University of Crete

Abstract

During the last years, there is a growing interest in location-based applications. For example, vehicles with navigation tools and location-based services, health monitoring and assistive technology, smart spaces, virtual reality applications, wearable computing, and games are becoming widely spread. For the support of location-dependent services, the positioning becomes critical.

One such location-sensing mechanism is the Cooperative Location-sensing System (*CLS*) that exploits the mobile peer-to-peer (*p2p*) paradigm and the wireless communication infrastructure. Each device uses a grid representation of the environment and exchanges location estimates with each other. CLS runs iteratively and identifies the more likely areas in which the device resides using this information. We evaluated CLS with extensive simulations using *ns-2*. Specifically, we analyzed the impact of range error, density of landmarks (i.e., hosts that know *a priori* their position), density of wireless devices, and mobility on the accuracy of the position estimations. For example, considering 10% of hosts to be landmarks, average connectivity degree equal to 10, and range error equal to 10% of the transmission range, 80% of the hosts can position themselves with a location error of at most 1.5 meters, when all hosts are stationary. 50% of all hosts can position themselves with a location error of at most 3.6 meters, when 60% of the them are mobile.

We evaluated the range error (i.e., distance estimation error) in ICS-FORTH by taking real-life signal strength measurements in this environment. 90% of the total area corresponds to a range error of at most 10% of the transmission range.

To improve the accuracy of the positioning, we extended CLS by incorporating addi-

tional information, such as a signal strength map, information about the environment (e.g., floorplan), and mobility information. We designed heuristics that use this information to identify the areas in which it is more likely the device to be located. Considering the aforementioned simulation scenario, 80% of the hosts can now position themselves with a location error of at most 0.8 meters, compared to 1.5 meters in the case of simple CLS. The same percentage of hosts can locate themselves with a location error of at most 2.6 meters, when 60% of them are mobile.

To summarize, the main contributions of this thesis are the extensive evaluation of the CLS and its enhancement.

Εντοπισμός Θέσης χρησιμοποιώντας την Υποδομή του IEEE 802.11 και το Δυομότιμο Μοντέλο για Κινητές Υπολογιστικές Εφαρμογές

Αναστασία Κατρανίδου

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών

Πανεπιστήμιο Κρήτης

Περίληψη

Τα τελευταία χρόνια, παρατηρείται ένα αυξανόμενο ενδιαφέρον για εφαρμογές και υπηρεσίες που παρέχουν πληροφορία βάσει της θέσης του χρήστη. Παραδείγματος χάριν, οχήματα εφοδιασμένα με όργανα πλοήγησης, η στενή παρακολούθηση της υγείας των ασθενών, τα έξυπνα σπίτια, εφαρμογές εικονικής πραγματικότητας, υπολογιστές που «φοριούνται», και παιχνίδια βασισμένα στον εντοπισμό θέσης διαδίδονται ευρέως.

Για την υποστήριξη αυτών των υπηρεσιών, ο προσδιορισμός θέσης αποτελεί ένα μείζον θέμα. Ένας τέτοιος μηχανισμός που έχει ως στόχο τον εντοπισμό θέσης μιας ασύρματης συσκευής είναι το συνεργατικό σύστημα εντοπισμού θέσης, Cooperative Location-sensing System (CLS) που εκμεταλλεύεται το δυομότιμο μοντέλο (peer-to-peer paradigm) και την υπάρχουσα υποδομή της ασύρματης επικοινωνίας. Κάθε ασύρματη συσκευή χρησιμοποιεί μια αναπαράσταση του περιβάλλοντος του με τη μορφή ενός πλέγματος, και ανταλλάσσει εκτιμήσεις θέσης με τις άλλες συσκευές. Το CLS τρέχει επαναληπτικά σε κάθε συσκευή και προσδιορίζει τις πιο πιθανές περιοχές στις οποίες βρίσκεται η συσκευή, χρησιμοποιώντας αυτές τις εκτιμήσεις.

Αξιολογήσαμε το CLS με εκτενείς προσομοιώσεις χρησιμοποιώντας τον προσομοιωτή δικτύου *ns-2*. Συγκεκριμένα, αναλύσαμε την επίδραση του λάθους εκτίμησης απόστασης (range error), της πυκνότητα των συσκευών που γνωρίζουν εκ των προτέρων τη θέση τους (landmarks), της πυκνότητας των ασύρματων συσκευών, και της κίνησης των συσκευών στην ακρίβεια των εκτιμήσεων θέσης. Για παράδειγμα, βρήκαμε ότι αν θεωρήσουμε το 10% των συσκευών να είναι landmarks, το μέσο βαθμό συνδετικότητας να είναι ίσο με 10, και το

range error ίσο με 10% της εμβέλειας εκπομπής σήματος, το 80% των συσκευών μπορεί να υπολογίσει τη θέση του με λάθος μικρότερο των 1,5 μέτρων, στην περίπτωση που όλες οι συσκευές θεωρούνται ακίνητες. Αντίστοιχα, το 50% των συσκευών μπορεί να υπολογίσει τη θέση του με λάθος μικρότερο των 3,6 μέτρων, όταν 60% από αυτές κινούνται.

Αξιολογήσαμε το range error στο περιβάλλον του ΙΠ-ΙΤΕ λαμβάνοντας πραγματικές μετρήσεις έντασης σήματος. Το 90% της συνολικής περιοχής αντιστοιχεί σε ένα range error μικρότερο του 10% της εμβέλειας εκπομπής σήματος.

Για να βελτιώσουμε την ακρίβεια του προσδιορισμού θέσης, επεκτείναμε το CLS ενσωματώνοντας επιπρόσθετες πληροφορίες, όπως έναν χάρτη έντασης σήματος, πληροφορίες για το περιβάλλον (π.χ., floorplan), και πληροφορίες για την κίνηση των συσκευών. Σχεδιάσαμε ευριστικές μεθόδους που χρησιμοποιούν αυτές τις πληροφορίες για να προσδιορίσουν τις περιοχές στις οποίες είναι πιθανότερο να βρίσκεται κάποια συσκευή. Λαμβάνοντας υπ΄ όψιν το προαναφερόμενο σενάριο προσομοίωσης, παρατηρήσαμε ότι το 80% των συσκευών μπορεί να υπολογίσει τη θέση του με λάθος μικρότερο των 0,8 μέτρων, σε αντίθεση με τα 1,5 μέτρα στην περίπτωση του απλού CLS. Το ίδιο ποσοστό συσκευών μπορεί να υπολογίσει τη θέση του με λάθος μικρότερο των 2,6 μέτρων, όταν το 60% από αυτές κινούνται.

Συνοψίζοντας, οι κύριες συνεισφορές αυτής της μεταπτυχιακής εργασίας είναι η εκτενής αξιολόγηση του CLS και η βελτίωση του.

Ευχαριστίες

Η συγκεκριμένη μεταπτυχιακή εργασία εκπονήθηκε στο εργαστήριο Δικτύων και Τηλεπικοινωνιών στο Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας. Ο καιρός που χρειάστηκε για να ολοκληρωθεί η εργασία ήταν από τις πιο δημιουργικές φάσεις στη ζωή μου καθώς ήταν η πρώτη φορά που ασχολήθηκα ερευνητικά με την περιοχή των τηλεπικοινωνιών.

Πρώτα από όλα θέλω να ευχαριστήσω την ακαδημαική μου σύμβουλο, επίκουρο καθηγήτρια κα. Μαρία Παπαδοπούλη για όλη τη βοήθεια και στήριξη που πάντα ήταν πρόθυμη να μου προσφέρει ώστε να επιτύχω τον αρχικό μου στόχο. Ελπίζω τελικά να φάνηκα αντάξια των προσδοκιών της. Επίσης, θέλω να ευχαριστήσω τον καθηγητή και πρόεδρο του εργαστηρίου μου κ. Απόστολο Τραγανίτη καθώς και τον καθηγητή κ. Παναγιώτη Τραχανιά που συμμετείχαν στην εξεταστική επιτροπή της εργασίας μου. Τέλος, θέλω να ευχαριστήσω τον αναπληρωτή καθηγητή κ. Παναγιώτη Τσακαλίδη ο οποίος ήταν ο πρώτος μου ακαδημαικός σύμβουλος και ο πρώτος που με υποδέχτηκε στο Πανεπιστήμιο Κρήτης.

Για την ολοκλήρωση της εργασίας μου με βοήθησαν επίσης ορισμένα άτομα, είτε με τις γνώσεις τους και το επιστημονικό τους υπόβαθρο είτε με την ψυχολογική και ηθική τους στήριξη. Αρχικά, θέλω να ευχαριστήσω όλα τα παιδιά από το εργαστήριο μου, που ο καθένας με το δικό του τρόπο, βοήθησαν στην επίτευξη του στόχου μου. Συγκεκριμένα, θέλω να ευχαριστήσω τον 'γκουρού' των δικτύων Σ. Παπαδάκη, τον Β. Αγγελάκη, τον Γ. Φωτιάδη και τον Γ. Σταματάκη για όλες τις επιστημονικές απορίες που μου έλυναν κατά καιρούς, τον 'διπλανό' μου Γ. Τζαγκαράκη που μου εξηγούσε τις μαθηματικές μου απορίες, την Κ. Καραδήμου που ήταν πάντα πρόθυμη να με βοηθήσει με το Latex επιδεικνύοντας απίστευτη υπομονή μαζί μου, τον Π. Αλαφούζο για τις στιγμές γέλιου και για τα muffins που μου έφερνε, καθώς και όλα τα παιδιά από το mobile group.

Στη διάρκεια αυτών των χρόνων, είμαι τυχερή που γνώρισα την φιλενάδα μου, πια, Δέσποινα Ευαγγελάτου και τον Λευτέρη Σιδηρουργό. Ίσως χωρίς τη βοήθεια τους να μην κατάφερνα τίποτα. Επίσης, ευχαριστώ τη Βανέσα Ευαγγελάτου, την Ρόη Φλουρή, την Ελευθερία Τζάμαλη, τον Ιάσονα Οικονομίδη και φυσικά τον Άγγελο για την όμορφη παρέα τους.

Τέλος, θέλω να ευχαριστήσω τους γονείς μου, τον Κώστα και την Χαρούλα, και τον αδελφό μου Αλέξανδρο για όλη την ηθική, ψυχολογική και οικονομική τους στήριξη αλλά κυρίως γιατί είναι πάντα δίπλα μου, ό,τι και αν συμβεί.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The immense progress in information technology has opened up exciting research challenges in personal and pervasive computing, transcending the traditional computing paradigms. The aim of pervasive computing is to create systems that are embedded in the environment, almost invisible and constantly available enhancing the information access without distracting the users from their main tasks. Among the emerging technologies expected to prevail in pervasive computing environments are wearable computers, sensors in cars and appliances, smart homes and smart buildings, that can make our lives easier and more productive.

The great technological development in wireless communications had a dramatic impact on the deployment of pervasive computing. In addition, the proliferation of high-speed wireless local area networks has enabled users to access the Internet while moving. The goal here is to enable users to interact effectively with their physical surroundings, such as print a document on the closest printer or locate oneself.

The determination of physical position is known as *localization* or *location-sensing*. Devices in pervasive computing environment will maintain information about their locations to support different location dependent applications.

Location-sensing is important for wireless networks since many applications depend on knowing the location of an object. For instance, there is growing interest in the transportation industry to equip vehicles with navigation tools and location-based services. In the medical community, location-based applications are used for patient monitoring and assistive technology. In addition, location-based routing protocols can save significant energy by eliminating the need for route discovery. Security can also be enhanced by

location awareness, for example, by preventing wormhole attacks. Location awareness is also applied in the entertainment industry (virtual reality applications and games), and in emergency situations for disaster relief.

In outdoor settings, GPS [1] has been used in many commercial applications. However, GPS does not work everywhere. In particular, the technology typically breaks down near obstacles, such as trees and buildings, and in indoor environments.

There have been several location-sensing systems in literature for mobile computing. They can be classified according to their dependency and use of an infrastructure, signal modalities, methodology for estimating distance and accuracy requirements [2]. Several location sensing systems use radio, infrared, sonar, or vision to infer the position of a mobile device. They may need a special infrastructure to operate, such as RF tags, ultrasound, and sensors. Recently, IEEE 802.11 networks have become widely available in universities and corporations providing wireless Internet access. More and more such networks are becoming available in airports, hospitals, shopping centers, and other public areas [3, 4]. Its low cost and the complexity advantages of using it for both communication and positioning, makes IEEE 802.11 an attractive choice. There are location sensing systems [5, 6] that use the infrastructure of IEEE 802.11 access points (APs), without the need of any specialized hardware. They apply probabilistic inference of position mechanism using the radio frequency (RF) emission from APs measured by the IEEE 802.11 cards to track the device. They build a map of the environment by sampling the space and gathering data at various predefined checkpoints of the indoor environment. Other works, such as [7, 8], combine different signal modalities, such as ultrasound from deployed sensors and radio from the IEEE 802.11 infrastructure for location estimation.

There are some recent research efforts on location-sensing systems that operate in an ad-hoc (i.e., infrastructureless) network ([9, 10, 11, 12, 13]). These systems assume cooperative devices that operate in a self-organizing manner. Hosts cooperate and exchange positioning information to estimate their location.

1.1 Challenges

Early location-sensing systems were based exclusively on signal strength measurements and simple triangulation methods. However, the use of only signal strength data and simple

triangulation methods for positioning can be limited, especially due to the interference and transient characteristics of the radio propagation. The dynamic characteristics of the environment, due to the interference and radio propagation, impose important challenges for the design of a scalable, easily deployable, and computationally inexpensive location-sensing system.

In many situations, due to environmental constrains, cost, maintenance, and regulatory barriers, the deployment of a specialized infrastructure for location sensing is not feasible. Our main goal is to design a location-sensing system, that does not depend on a specialized hardware. Given the widely deployment of the IEEE 802.11 communication infrastructure, its use for both communication and positioning becomes a very attractive choice. There are several location-sensing systems that are based on the IEEE 802.11 infrastructure.

RADAR [5] is a system that inspired this work. It uses an infrastructure of IEEE 802.11 access points (APs). In RADAR, IEEE 802.11 cards measure the signal strength from those APs. Based on this data, they build a signal strength map of the environment. Bahl and Padmanabhan [5, 14] report that 90% of the hosts can be located with at most 3 meters of location error. The error distance for tracking a mobile user is at most 3.5 meters. Our aim is to improve this performance.

1.2 Motivation

We aim to build a location-sensing system for mobile computing applications that can provide position estimates, within a few meters accuracy, without the need of specialized hardware and extensive training. Instead, it will take advantage of the available communication infrastructure. Our location-sensing system should operate both on indoors and outdoors environments.

We are interested in exploring the impact of the mobile peer-to-peer (*p2p*) paradigm, the knowledge of the environment and mobility on the positioning. The mobile p2p paradigm has not been explored in the wireless networking community.

1.3 Thesis statement

This thesis is based on an existing location-sensing system, called Cooperative Location-sensing System (*CLS*) [15]. CLS enables devices to cooperate with each other in a self-organizing manner. Hosts gather positioning information periodically from other cooperative devices and estimate their distance from their neighboring peers. They can refine their measurements iteratively as they incorporate new information from their neighbors. The main differences of CLS from other location-sensing systems that operate in ad-hoc networks [9, 10] are:

- A grid-based representation of the environment that allows the system to easily incorporate external information (such as GPS, application-dependent semantics, floorplan, signal strength maps) to improve the location estimation.
- A positioning estimation algorithm that aims to incorporate positioning information from hosts in the network and enable them to estimate their position in a self-organizing and adaptive manner.

CLS is easy and cost effective to deploy and maintain, yet it remains competitive in its accuracy compared to the best published results on location-sensing in mobile computing. It was designed by M. Papadopouli and H. Fretzagias. They evaluated its performance via simulations focusing on the case of stationary nodes. Fretzagias implemented CLS in his own simulation platform. In this thesis, we re-implemented CLS in a popular networking simulation platform, *ns-2*. We selected *ns-2* because it provides tools for simulating wireless networks and mobility.

Firstly, through an extensive performance analysis, we evaluated the existing implementation of CLS. In particular, we studied the impact of several parameters, such as the density of landmarks, the density of wireless devices, the range error and the mobility on the accuracy of the position estimation, through several simulation scenarios. Secondly, to enhance its accuracy, we extended CLS by incorporating additional information, such as a signal strength map, information about the environment (e.g., floorplan) and the mobility of nodes.

We integrated these measurements in the simulations. In addition, we proposed some extensions of the algorithm that aim to enhance the performance of CLS under mobility.

Furthermore, we studied the *range error* (i.e., distance estimation error), based on real-life signal strength measurements that had taken place in the Institute of Computer Science (ICS) of the Foundation for Research and Technology - Hellas (FORTH).

To summarize, the main contribution of this thesis are the extensive evaluation of the CLS and its enhancement. For example, we found that:

- 80% of the hosts can position themselves with a location error of at most 0.8 meters, when all of them are stationary.
- 80% of the hosts can locate themselves with a location error of at most 2.6 meters, when 60% of them are mobile.

1.4 Thesis outline

The present thesis is organized as follows: Chapter 2 presents a definition of the location-sensing systems and a basic classification of these systems. In addition, the most important location-sensing techniques that these systems use are presented. In Chapter 3, we describe the most representative location-sensing systems of the location-sensing computing and the robot localization area. In Chapter 4, we describe the algorithm of *Co-operative Location-sensing System* (*CLS*), that is, the *Communication Protocol* and the *Voting Process*. In Chapter 5, we analyze the performance analysis of CLS via simulations when all the hosts are stationary and there is no mobility. Specifically, we investigate the impact of range error (i.e., distance estimation error), ST and LECT values, and the connectivity degree and percentage of landmarks on the performance of the system. In Chapter 6, we discuss and evaluate a CLS extension, namely, the use of a signal strength map of the environment, information of the envirnoment and heuristics methods. Chapter 7 examines the impact of mobile nodes in CLS and then presents some extensions, that enhance CLS performance, in case there are mobile nodes. In Chapter 8, we present an experimental analysis of range error using actual measurements in ICS-FORTH. In Chapter 9, we compare our work with the most representative location-sensing systems that are referred in the related work (Chapter 3). Finally, in Chapter 10, we discuss our main conclusions and future work plans.

Chapter 2

Background on Location-sensing

In this Chapter, we describe the basic operation of location-sensing systems and we present a classification of these systems based on the location-sensing properties. In particular, we classify the location-sensing systems based on the type of metric, use of hardware, location of the infrastructure, and description of the position. In addition, we refer the location-sensing methods, that is, the triangulation, proximity, and scene analysis. Finally, we describe the robot localization which differs from the mobile computing localization.

2.1 Location-sensing systems

The basic operation of a location system is to gather information about the position of a mobile node that operates in a geographical region and process this information in order to estimate its position.

A popular approach, known as radiolocation, determines the relative direction, the position or the movement of an object using the propagation characteristics of radio waves (e.g. change of phase, roundtrip delay, orientation of arrival) [16]. The precision in the localization depends on various factors, such as the carrier frequency of the radio wave, noise and interference (e.g., reflections, diffractions, temperature of environment), and topology of transceivers. Thus, one should exploit the resources properly, in order for these factors to have the minimum impact in the localization process.

2.2 Classification of location sensing systems

The way that the location sensing systems are classified is not unique, but it is varied. Further down, we present some of their categories and the basic characteristics of each category:

2.2.1 Type of metric

The basic classification of the location sensing systems is based on the method that a radio frequency can be conducted and include systems that use the following methods:

- Received signal strength indication (RSSI)
- Direction resolve (Angle of Arrival)
- Distance resolve (Time of Arrival, Time Difference of Arrival)

The signals that are going to be used in the localization process, in each of the above systems, could be either radio frequencies (RF) from the IEEE 802.11 wireless network, infrared (IR) or ultrasound.

Systems based on signal strength

The Received signal strength indication (RSSI) is a measurement of the strength (not necessarily the quality) of the received signal in a wireless environment, in arbitrary units. Location-sensing that uses signal strength of the received signal uses a known mathematical model which describes the path loss attenuation of the signal with distance. The measurement of signal strength provides a distance estimate between the mobile object and the base station. Consequently, the mobile object must lie on a circle which has as a center the base station and as radius the distance between them. The distance can be measured by calculating the path loss. The path loss can be found if the mobile object knows the power transmitted from the base station and the received power.

In general, the signal strength that a mobile object receives from a base station attenuates as the object goes away from the base station. For instance, in Fig. 2.1, the signal strength ($SS1$) that host $h1$ receives from the base station BS is greater than the signal strength ($SS2$) that host $h2$ receives from the same base station, since host $h1$ is closer to

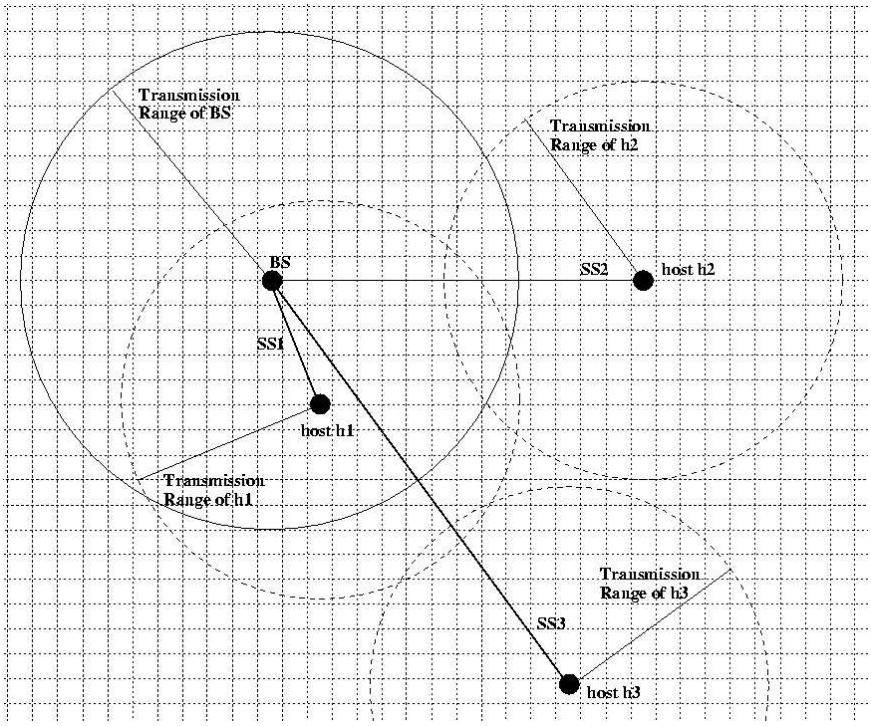


Figure 2.1: Location-sensing based on signal strength.

the BS than host $h2$. On the other hand, host $h3$ cannot 'hear' the BS since it is outside of its transmission range.

If we use three base stations or more, the position of an object can be determined using the technique of triangulation that will describe further down. For signal strength based location systems, the primary source of error is multipath fading and shadowing. The errors due to shadow fading can be combated by using pre-measured signal strength maps centered at the base stations, which describe the changes of the signal strength for a given space. However, this presupposes a stable topological environment and enough preparatory work. In any case, the results that we take depend intensely on the variability of the environment.

Systems based on direction

The systems based on the direction of a signal estimate the mobile object's location by first measuring the Angle of Arrival (AoA) of a signal from a mobile object at several base stations through the use of antenna arrays. The intersection of the lines that define the direction of the signal is the position of the mobile object (Fig. 2.2). Scattering near and around the mobile object and the base station will alter the measured AoA. In the absence

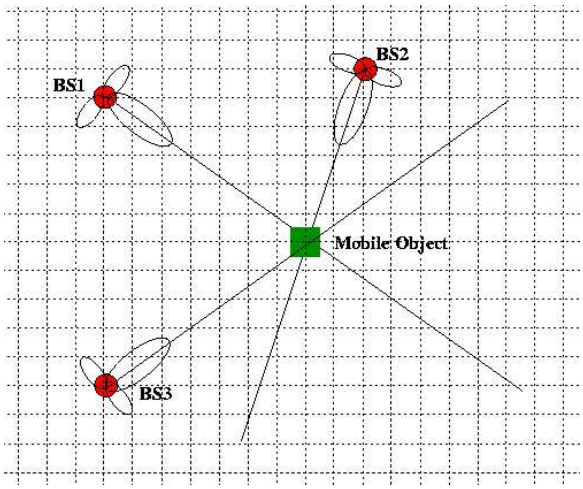


Figure 2.2: Location-sensing based on Angle of Arrival (AoA).

of a line-of-sight (LOS) signal component, the antenna array will take a reflected signal that may not be coming from the direction of the mobile object. Even if a LOS component is present, multipath will still interfere with the angle measurement. The accuracy of the AoA method diminishes with increasing distance between the mobile object and the base station due to fundamental limitations of the devices used to measure the arrival angles. In some measurements that have taken place, this technique gives good results when it is used in macrocells, since the signals arrive with a relatively narrow AoA spread at the base stations. For microcells, this technique gives worse results.

Systems based on distance

The systems based on distance estimate the position by measuring the absolute distance, or the difference of the distances, of the mobile object from the base stations. The

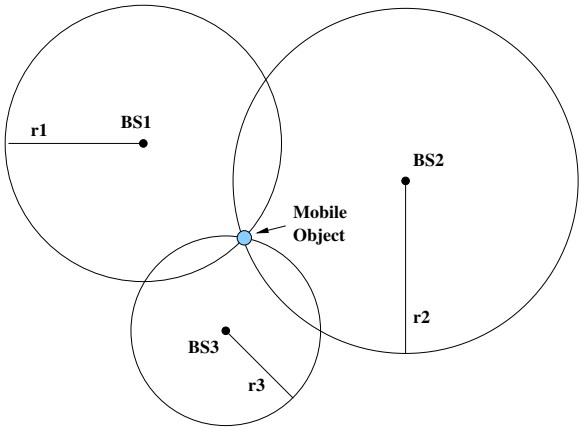


Figure 2.3: Location-sensing based on Time of Arrival (ToA).

distance between a mobile object and a base station is measured by finding the Time of Arrival (ToA) which is the one-way propagation time between them, assuming that the transmission time is known. Geometrically, this is depicted with a circle, centered at the base station, on which the mobile object must lie. By using at least three base stations to resolve ambiguities, the position of the mobile object is at the intersection of the circles (Fig. 2.3).

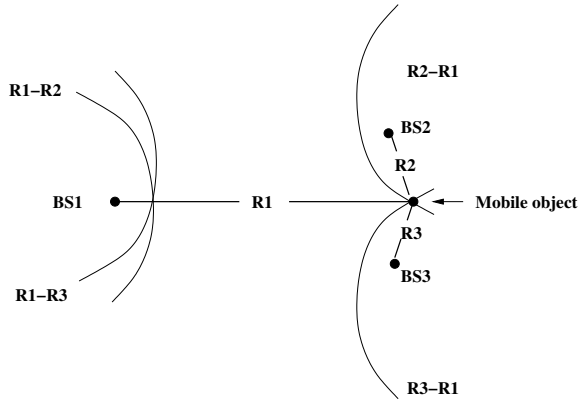


Figure 2.4: Location-sensing based on Time Difference of Arrival (TDoA).

A basic precondition for the system to function is the absolute synchronization of the mobile object and the base station. When this is not true, instead of the absolute times, the Time Difference of Arrival (TDoA) is used, that is, the differences of the times of arrival in every base station, since it is much easier for the base stations to be synchronized. Since the hyperbola is a curve of constant time difference of arrival for two base stations, the time differences define hyperbolae, on which the mobile objects must lie. Hence, the location of the mobile object is at the intersection of the hyperbolae, as it is depicted in Fig. 2.4.

2.2.2 Use of hardware

Location-sensing systems can also be classified in several categories based on their properties [2]. Firstly, we classify the location sensing systems, regarding the use of hardware.

Infrastructure vs. ad-hoc

- *Systems based on infrastructure*: The location-sensing systems that belong in this category take advantage of their environment's infrastructure. For instance, a system can receive position information from an access point which is wired connected with the Internet.

- *Ad-Hoc systems*: An ad-hoc network is an infrastructureless, self-configuring network of mobile hosts connected by wireless links, forming an arbitrary topology. The hosts are free to move randomly and organize themselves arbitrarily. Such a network may operate in a standalone fashion. In a purely ad-hoc location sensing system, all of the entities estimate their locations by cooperating with other nearby objects. In this way, objects of a cluster of the ad-hoc network are located relative to one another or absolutely if some objects in the cluster occupy known locations.

The first category comprises a more realistic phenomenon since the companies and the institutes are based in an infrastructure. On the other hand, the ad-hoc approach is applied in military or research purposes.

Specialized vs. non-specialized hardware

- *Specialized hardware*: These systems are tightly connected to specialized hardware (such as tags, cameras, ultrasound receivers) to locate a wireless device.

- *Non-specialized hardware*: There are location-sensing systems that have no need of a specialized hardware, but they based on existing infrastructure, such as the IEEE 802.11 infrastructure.

2.2.3 Location of the infrastructure

Another classification of the location-sensing systems is based on the location of the infrastructure:

Terrestrial vs. satellite

We can classify the location-sensing systems, regarding the location of the infrastructure:

- *Terrestrial*: The terrestrial systems (e.g., APs and sensors) are placed on the ground.

- *Satellite*: Satellite location systems use satellites as reference points to calculate positions accurate to within meters. With advanced forms of satellite navigation, measurements can be made down to the centimeter level. The main advantages of these systems are economy and better services. The most popular satellite location

system is the Global Positioning System (GPS), which is described in the following chapter.

Auto-localized vs. remotely localized

Depending on whether the measurements take place locally or remotely, we can classify the location-sensing systems into two categories:

- *Auto-localization systems*: In these systems, the measurements are taken place in the device that is to be localized itself, as in GPS.
- *Remote localization systems*: In remote localization systems, the measurements are taken place in a different place other than the device that is to be localized.

2.2.4 Description of the position

The location-sensing systems can be classified regarding the description of the position.

Physical vs. symbolic

From the point of view of the type of information available, location-sensing systems can be characterized as either physical or symbolic.

- *Physical*: Physical information provides the position of a location on a physical coordinate system (e.g., global coordinates).
- *Symbolic*: Symbolic information employs textual descriptions of location (e.g., next to the laboratory) and may be either geographic symbolic, or place symbolic.

Absolute vs. relative

Whether physical position or symbolic location is used, the information provided may be either absolute or relative.

- *Absolute*: Absolute position implies a location system that employs a shared reference grid for all located objects, for example latitude, longitude and altitude.
- *Relative*: A location system that uses relative positions can have a distinct frame of reference, for example three meters from the traffic lights.

Furthermore, there are location-sensing systems that use multiple modalities. For instance, they use both RF and ultrasound tags. Although, they can provide better location accuracy, using more than one modality, they need more hardware. In addition, the location-sensing systems can be categorized based on their cost and whether they are scalable or not. Finally, the privacy is another parameter that location-sensing systems should take into consideration.

2.2.5 Accuracy vs. precision

The examples of the previous sections raise naturally the question of accuracy. If more resolution is required, for example to pinpoint an object the size of a car or a person, then a more accurate system should be used.

- **Accuracy**: The property of the smaller distance that a system can differentiate is called accuracy of the system.
- **Precision**: The percentage of the times that the prescribed accuracy is achieved is called precision of the system.

Accuracy and precision are often the two axes of a trade-off: less accuracy may be traded for more precision. Thus, using only one of the two attributes of spatial location is not a suitable measure for comparison of location sensing systems. Rather, location systems should be assessed on the basis of the error distribution incurred when locating objects, taking into account any dependencies, for example the required density of infrastructure (for example satellites, base stations, radio frequency readers and so on).

2.3 Location-sensing methods

The most common techniques in location-sensing are triangulation, proximity and scene analysis [2].

2.3.1 Triangulation

The triangulation technique uses the geometric properties of triangles to compute object locations. It can be done via lateration, which uses multiple distance measurements

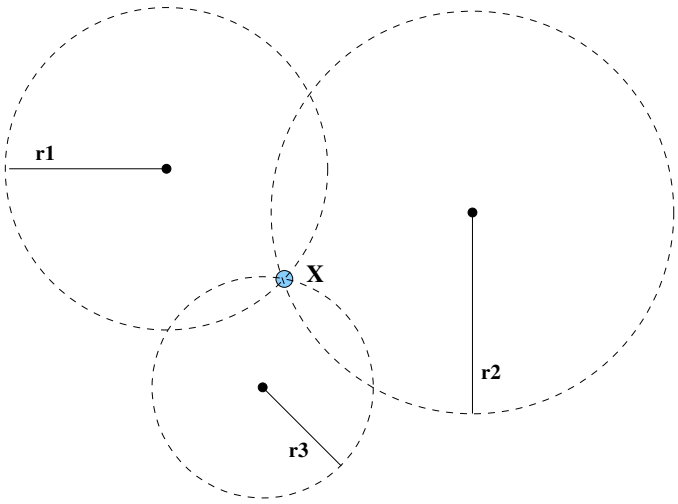


Figure 2.5: Determination of 2D position using lateration. It requires distance measurements between the object 'X' and 3 non-collinear points.

between known points, or via angulation, which measures angle or bearing relative to points with known separation.

Lateration

The technique of lateration computes the position of an object by measuring its distance from multiple reference positions. In two-dimensions, an object's position can be computed using the distance measurements from 3 non-collinear points, as it is depicted in Fig. 2.5. In three dimensions, an object's position requires the distance measurements from 4 non-coplanar points. The distance measurements can be calculated using three general approaches:

- **Direct**. Direct measurements of distance uses a physical action or movement.
- **Time-of-Flight**. Measuring distance from an object to some point P using time-of-flight means measuring the time it takes to travel between the object and point P at a known velocity.
- **Attenuation**. The intensity of an emitted signal decreases as the distance from the emission source increases. The decrease relative to the original intensity is the attenuation. For example, a free space radio signal emitted by an object will be attenuated by a factor proportional to $\frac{1}{r}$ when it reaches point P at distance r from the object. Attenuation is less accurate than time-of-flight.

Angulation

Angulation is similar to lateration but instead of distances it uses angles to determine an object's position. In general, two dimensional angulation requires two angle measurements and one length measurement, such as the distance between the reference points, as it is depicted in Fig. 2.6. In three dimensions, a precise position can be computed using one length measurement, one azimuth measurement and two angle measurements.

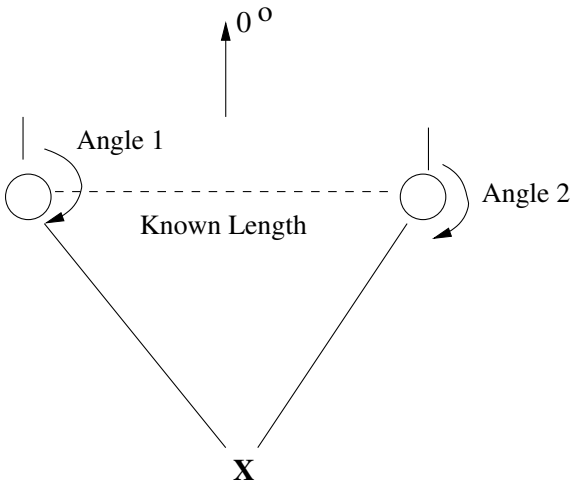


Figure 2.6: 2D position using angulation. It requires at least two angle measurements and one distance measurement.

2.3.2 Proximity

The proximity technique measures nearness to a known set of points. There are three general approaches to sense proximity:

- **Detecting physical contact**. Technologies for sensing physical contact include pressure sensors, touch sensors and capacitive field detectors.
- **Measuring wireless cellular access points**. Another implementation of the proximity location technique is monitoring when a mobile device is in range of one or more access points in a cellular network.
- **Observing automatic ID systems**. A third implementation of the proximity technique uses automatic identification systems, such as credit card point-of-sale terminals, computer login histories, land-line telephone records and electronic card lock logs.

2.3.3 Scene analysis

The scene analysis technique uses features of a scene observed from a particular vantage point to draw conclusions about the location of the observer. In *static* scene analysis, observed features are looked up in a predefined dataset that maps them to object locations. In contrast, *differential* scene analysis tracks the difference between successive scenes to estimate location. Differences in the scenes will correspond to movements of the observer. The scene itself can consist of visual images or any other measurable physical phenomena, such as the electromagnetic characteristics that occur when an object is at a particular position and orientation.

The advantage of scene analysis is that the location of objects can be inferred using passive observation and features that do not correspond to geometric angles or distances. The disadvantage of scene analysis is that the observer needs to have access to the features of the environment against which it will compare its observed scenes. Furthermore, changes to the environment in a way that alters the perceived features of the scenes may necessitate reconstruction of the predefined dataset or retrieval of an entirely new dataset.

2.4 Robot localization

Robot localization is a well studied problem and it is a key component in many successful autonomous robot systems [17, 18]. Robot localization is the process of having an estimate of a robot's location with respect to its environment, given a representation of this environment and some sensing ability within the environment. This means that the robot has to find out its location in relation to the environment. When we talk about location, pose, or position we mean the x and y coordinates and the heading direction of a robot in a global coordinate system. If a robot does not know where it is, relatively to the environment, it is difficult to decide what to do. The robot needs to have some idea of where it is, in order to be able to operate and act successfully [19]. By some authors the robot localization problem has been stated as the "most fundamental problem to providing robots truly autonomous capabilities" [20].

In our case, we can consider any wireless device as a mobile robot. A particularly challenging case of localization is when there is no *a priori* estimate of the robot's location (*global localization problem*). Similarly, in our case, a wireless device has no information

about its position before it starts communicating with the networks base stations. Furthermore, it has to continuously refine the estimations of its pose (*pose maintenance*).

Much progress has been made in developing localization techniques since the problem first appeared in the literature. Dead reckoning, which means acquiring relative measurements, can be used for pose maintenance, but requires some initial knowledge of location. Some of the simplest methods for global localization include landmark-based localization and triangulation. Probabilistic techniques, such as Kalman filtering [21, 22], and later, Bayesian analysis [23], were developed to address noisy measurements in these systems [24].

2.4.1 Robot vs. mobile computing localization

Location-sensing in robotics differs from that in mobile computing. For instance, in mobile computing the movement of a user is less controlled and automated than the movement of a robot. Furthermore, many robots do not have any energy problems or power restrictions since they can carry a battery and a powerful processor, whereas a user can only carry a laptop or a PDA which have limited power and battery supplies. Thus, a robot can solve complex mathematical computations. In addition, a robot may have extra sensors or hardware for additional positioning information.

However, a robot needs a training phase to have a notion of its environment. Moreover, different sensors can also give inconsistent or incorrect sensor readings.

2.5 Summary

Table 2.1 indicates the location-sensing properties, as we described in the above Sections.

Different location sensing systems that use lateration have different means to estimate the required distances. The simpler way to do this is when the distance is directly available. However, it is only in special cases (e.g., robotics) when distance information is available directly. Radio flight-of-time is the approach used by GPS which in order to achieve good accuracy has to maintain the clocks of its satellites within 10^{-13} of a sec within each other. A system that uses attenuation in a wireless LAN environment to estimate location is Active Campus [25]. Visual scene analysis is frequently used in robotics and signal strength profiling has been used in the RADAR [5] system.

	Location-sensing properties
Type of metric	SS AoA ToA, TDoA
Use of hardware	infrastructure vs. ad-hoc specialized vs. no specialized hardware
Location of the infrastructure	auto-localized vs. remotely localized terrestrial vs. satellite
Description of the position	physical vs. symbolic absolute vs. relative
	multiple modalities vs. one
Location sensing methods	triangulation proximity scene Analysis

Table 2.1: Location-sensing properties.

The coverage area of a RF-based system can be extended from a single room to the whole building by deploying more RF readers. However, the cost of the extended deployment might be considerable. Thus, when considering location sensing from the point of view of scale, an appropriate measure is the number of locatable objects (for example GPS receivers or RF tags) in the coverage area per unit of infrastructure.

In some cases it is also possible to move between relative and absolute position. It is straightforward to transform an absolute position to a relative one, if a second reference point is available and also in many cases it is possible to use multiple relative readings to retrieve an absolute position when the absolute positions of the reference points are known. The distinctions along the absolute/relative and physical/semantic split should not be seen as inherent system capabilities but rather as a useful abstraction for the identification of the types of information available at any one system.

A final consideration for location sensing systems is their limitations. Since the majority of them depend on the propagation properties of radio through space their effectiveness and efficiency is correlated to the environment. For example, GPS receivers are unable to operate effectively indoors where the signals are very weak.

Chapter 3

Related Work

In this chapter, we present an overview of research in localization systems. More specifically, we focus on recent studies that deal with the determination of a mobile computing device or a robot. In the former case, we refer to it as *Location-sensing computing*, whereas in the latter case, we reckon with *Robot localization*. We refer to GPS, Ekahau project, NearMe, and E911 which use a different technology. We describe these studies, in detail, and contrast them with CLS (Ch. 9).

3.1 Location-sensing computing

We divide the location sensing systems in the following indicative categories, based on the location-sensing properties we examined in Ch. 2, and describe some representative location-sensing systems of each category.

3.1.1 Systems based on distance

Early location-sensing systems which targeted on in-building localization, required specialized hardware to determine a device's location. For instance, the Active Badge location system [26] and the more recent Active Bat system [27] are two of the first systems in this field which are based on distance measurements in order to locate an object. Active Badge uses specialized tags which emit diffuse infrared pulses detected by ceiling-mounted sensors. Its accuracy granularity is equal to a room size but it needs one base station in every room. The Badge communicates with the base station every 10 sec. One of its main limitations is the sunlight and fluorescent interference with the infrared pulses.

Active Bat uses an ultrasound time-of-flight technique to provide accurate physical positioning. Specifically, users and objects must carry Bat tags. These tags emit an ultrasonic pulse to a grid of ceiling-mounted receivers and a simultaneous 'reset' signal over a radio link. Each ceiling sensor measures the time interval from 'reset' to ultrasonic pulse arrival and computes its distance from the Bat tag. The Active Bat has location accuracy of nine centimeters in 95% of the measurements. It needs one base station in every 10 square meters. The Bat makes 25 computations in every sec for every room. However, it requires large infrastructure (ceiling sensor grids) and maintenance cost.

The Cricket Location Support System [8, 7] uses both specialized ultrasound and radio frequency receivers to detect signals transmitted by fixed ultrasound emitters. The additional radio frequency signals are used for synchronization of the time measurement and to distinguish ultrasound signals that stem from multi-path effects and ignore them. The mobile object performs triangulation computations relative to the beacons. Cricket has simpler hardware and infrastructure than the aforementioned systems, but worse accuracy. Specifically, it does not require a grid of ceiling sensors with fixed locations but returns an estimation of the users position with a possible error of a four foot by four foot region.

Finally, in Spot-On system [11], special tags use radio signal attenuation to estimate distance between tags. In this system, there are multiple base stations, which provide signal strength measurements and a central server which aggregates this data. The aim is to localize wireless devices relative to one another, rather than to fixed base stations, allowing for ad-hoc localization. Its accuracy depends on the cluster size, and every cluster must have at least two tags. However, radio signal attenuation is less accurate than the time-of-flight.

Finally, EasyLiving [28] uses motion tracking cameras to determine the distance of an object in a home environment. It needs three cameras in every room and its precision varies. However, vision location systems need high processing power and must continuously struggle in order to maintain analysis accuracy as scene complexity increases.

3.1.2 Systems based on direction

Other location-sensing systems were based on the capability of the nodes to sense the direction from which a signal is received, in order to determine a device's location. One representative work is that of Niculescu and Badri Nath [29]. They designed and evaluated

a cooperative location-sensing system that operates in an ad-hoc network. They examined two algorithms, the 'DV-Bearing', which allows each node to get an angle estimation to a landmark, and 'DV-Radial', which allows a node to get an angle estimation to a landmark. It uses primarily angle estimations and assumes specialized hardware that allows a host to calculate the angle between two hosts. This can be done through antenna arrays or ultrasound receivers. Hosts gather data, compute their solutions, and propagate them throughout the network. Specifically, nodes immediately adjacent to a landmark get their angle estimations directly from the landmark. Assuming that a node has some neighbors with orientation for a landmark, it will be able to compute its own orientation with respect to that landmark, and forward it further into the network. The main disadvantage is that if a node uses imprecise information to solve its location, it is likely that its solution will contain more error than if the information used had accurate readings directly from landmarks that know their position exactly.

3.1.3 Systems based on signal strength

Later systems for location sensing-computing used off-the-shelf wireless networking hardware, measuring radio frequency (RF) signal intensity to determine a mobile object's location. The RADAR system [5, 14] was one of the first tracking systems based on the IEEE 802.11 wireless networking technology. RADAR measures at the base station the signal strength and signal-to-noise ratio of signals that wireless devices send and then it uses this data to compute the position within a building. The signal strength measurements employ a signal strength map with approximately one scan every square meter. Specifically, RADAR maintains a database of (x, y, z, SS_i) values, where x, y, and z are the physical coordinates of the training sample and SS_i is the signal strength measurements from the i-th access point (AP) for $i = 1, \ \ldots, n$. Each measured signal strength vector is then compared against the database and the coordinates of the best matches are averaged to give the solution. The RADAR approach offers two advantages: it requires only a few base stations and uses the same infrastructure that provides the buildings general-purpose wireless networking. Bahl and Padmanabhan [5, 14] report localization accuracy of about 3 meters of their actual position with about 50 percent probability. However, significant changes in the environment, such as moving metal file cabinets or large groups of people congregating in rooms or hallways, may necessitate reconstructing the predefined signal

strength database or creating an entirely new database. They discuss the problems of localizing an object in multiple floors and changing environmental conditions, as well as tracking of moving users. The error distance for tracking a mobile user is at most 3.5 meters.

Niculescu and Nath [10] introduced an algorithm on the field of location-sensing that works on simple geometric principles of Euclidian geometry concerning triangles and quadrilaterals. The information of the landmark locations is slowly propagated towards the nodes that are further away, while at the same time closer nodes enrich this information by determining their own location. Three variations of this algorithm are presented: 'DV-hop', 'DV-distance', and 'Euclidian'. In 'DV-hop', which is a completely ad-hoc approach, the nodes that somehow know their position, referred as anchors, flood their location to all nodes in the network. Each unknown node records the position and the number of hops to at least three anchors. Whenever an anchor α_1 infers the position of another anchor α_2, it computes the distance between them, divides that by the number of hops and floods this average hop distance into the network. Each unknown node uses the average hop distance to convert hop counts to distances, and then performs a triangulation to three or more distant anchors to estimate its own position. 'DV-hop' works well in dense and regular topologies, but for sparse or irregular networks the accuracy degrades to the radio range. The 'DV-distance' method is similar with the previous one with the difference that distance between neighboring nodes is measured using radio signal strength and is propagated in meters rather than in hops. Finally, the 'Euclidean' method works by propagating the true Euclidean distance to the landmark, so this method is the closest to the nature of GPS. This method uses simple properties of quadrilaterals. For instance, using 5% landmarks Niculescu's algorithms have an error of approximately 25%, 45% and 55% of the transmission range, for the 'DV-hop', 'DV-distance' and 'Euclidian' algorithms, respectively. However, when the number of landmarks increases to 20%, the algorithms perform at approximately 12%, 25% and 12% of the transmission range.

Saverese *et al.* [9] proposed another approach that is closest to our research. Their system employs a distributed algorithm that determines the position of nodes in an ad hoc network in two phases, namely the 'startup' and 'refinement' phase. In the 'startup' phase, landmarks broadcast their location among all nodes in the network and hosts estimate their position by triangulation. These estimations are rough approximations and are

improved in the second phase of the algorithm. The 'refinement' phase proceeds in iterations. At each iteration, each host broadcasts its position estimate, receives the positions and the corresponding range estimates from its neighbors, and computes a least square triangulation solution to determine its new position. After a number of iterations, when the position update becomes small, the refinement phase stops and reports the final position. The authors report that their algorithm is able to achieve position errors of less that 33% of the radio range, with 5% range error, 5% anchor nodes and an average connectivity of 7.

3.1.4 GPS

(a) (b)

Figure 3.1: (a) The GPS constellation of 24 satellites and (b) a portable GPS receiver.

The most widespread and reliable system of localization is the Global Positioning System (GPS), which is constituted by a constellation of 24 satellites [1] (Fig. 3.1). It uses these satellites as reference points to calculate the position of an object with precision of meters. Substantially it gives in each cell-meter of planet a unique address. It is included in the satellite systems of auto-localization, since each user must have a special appliance GPS which is needed in the localization. Moreover, the optical contact with at least 3 satellites is essential. Its precision is better than 10 meters for military use and 100 meters for commercial one. The last years has been developed the differential GPS (Differential GPS) that improves considerably the precision in 2 with 5 meters. It uses a network of stationary GPS receivers to calculate the difference between their actual known position

and the position as calculated by their received GPS signal. The "difference" is broadcasted as a local FM signal, allowing many civilian GPS receivers to "fix" the signal for greatly improved accuracy. Its disadvantages are that it needs the help of a land station and has low degree of renewal.

One of the basic disadvantages of GPS concerning the other systems of localization is that the appliance that is needed is quite expensive, bulky and consumes lot of energy. Consequently, it is disadvantageous in order to be incorporated in other appliances. To obtain less than 10 meters accuracy, it needs to take precise measurements from at least 3 satellites, which is difficult to be achieved in big cities with the abundance of buildings. Consequently, an important disadvantage is that it cannot work in internal spaces, such as buildings.

3.1.5 Ekahau Positioning Engine

A known software based on real-time location system is the Ekahau Positioning Engine (EPE) [30]. It is based on signal strength calibration. Ekahau is capable of pinpointing Ekahau T201 Wi-Fi Tags, laptops, PDAs and other Wi-Fi enabled devices, with floor-, room-, and door-level accuracy. It uses a proprietary protocol to communicate signal strength readings to the server. The specifications of the system report that it locates a wireless device with up to 1-meter average accuracy (indoors, three or more overlapping access point signals) and it works with any standard 802.11 infrastructure and allows selection of contributing access points. Its server is Java-based, and runs on Windows or any type of Unix with a working Java virtual machine, which is a network independent software architecture. It is compatible with all major 802.11b, 802.11g and 802.11a WLAN adapters.

However, after experiments that have been made by students of our lab in ICS-FORTH, we conclude that the accuracy of Ekahau is much worse than it is reported in their specifications. In particular, if there are less of 4 APs in the environment, a wireless device can be located with up to 4-meter average accuracy. The accuracy of the system is enhanced considerably when there are more than 7 APs in the environment.

Moreover, Scott Gifford [31] has tried Ekahau at the University of Michigan and found that although the tags are available at a moderate cost, the battery lifetimes are very short (approximately 1 day with location updates every second). The network was sufficiently

dense that no additional infrastructure was required. In his experiments, he found that the laptop had a median error of 2 meters, and was able to locate the device in the correct room 86% of the time, and the PDA had a median error of 3.2 meters and was placed in the correct room 72% of the time.

3.1.6 NearMe

NearMe [32] is a 802.11-based location system that performs self-mapping (i.e., a location system which can build the radio map as the system is used). NearMe is a service designed to determine when two devices are in proximity. It allows 802.11 APs to be associated with physical places (e.g., 2nd floor copy room) or resources (e.g., duplex printer). NearMe uses radio traces to build a neighborhood graph of which APs are near each another out to eight hops. Although NearMe does not estimate absolute locations, it does estimate ranges between APs based on the traces and the neighborhood graph.

3.1.7 E911

The US Federal Communications Commission's E911 telecommunication initiatives require that wireless phone providers develop a way to locate any phone that makes a 911 emergency call [33]. E911 is not a specific location-sensing system, but has led many companies that are developing a variety of location systems to determine a cellular phone's location.

Location-sensing systems developed to comply with the E911 initiatives can also support new consumer services. For example, a wireless telephone can use this technology to find the nearest gas station, post office, movie theater, bus, or automated teller machine. Data from many cellular users can be aggregated to identify areas of traffic congestion. To comply with E911, vendors are exploring several RF techniques, including antenna proximity, angulation using phased antenna arrays, lateration via signal attenuation and time of flight, as well as GPS-enabled handsets that transmit their computed location to the cellular system. To meet the FCC requirement, positioning must be accurate to within 50 meters for 95 percent of calls with receiver-based handset solutions such as GPS, or to within 300 meters with network-transmitter-based approaches.

3.2 Robot localization

Two relative publications that have been inspired from the robot localization community, are those of the Rice University [6, 34]. These systems use an infrastructure of IEEE 802.11 access points. Based on the signal strength data, gathered by wireless cards at various predefined points of an indoor environment, they build a signal strength map of the environment. They locate a wireless device by using this map and applying a probabilistic inference of the position. Their first system, is based on a two step process. In the first step a host uses a probabilistic model to compute the conditional probability of its location in a number of different locations based on the received signal strength from nine APs. The second step of their algorithm exploits the limited maximum speed of mobile users to refine the results of the first step and reject solutions with significant change in the location of the mobile host. Ladd *et al.* report that 77% of the time, their stationary hosts can be located with at most 1.5 meters of error using 9 APs, whereas, in case of mobility, 64% of the hosts can be located with at most 1 meter of error using the same number of APs.

Their second system, which is based on their first one, uses a topological map of the environment instead of a grid-based map and also they use probabilistic techniques to succeed more accurate location results. Ladd *et al.* report that in all cases, they determined the correct cell in at least 70% of trials.

3.3 Summary

In this chapter, we presented some representative studies on the localization. We focused on recent studies that deal with the determination of a wireless device or a robot. More specifically, we referred to systems based on distance (Active Badge, Active Bat, Cricket, Spot-On, Easy Living), systems based on direction (APS), systems based on signal strength (RADAR, APS with AoA, Savarese *et al.*) and systems based on the robot localization (Ladd *et al.*). Table 3.1 depicts the classification of these location-sensing systems, in relation to the classification we described in Ch. 2.

Location System	Technique	Physical/ Symbol	Absolute/ Relative	Ad-hoc/ Infrastructure
Active Badge	Diffuse infrared cellular proximity	Symbol	Absolute	Infrastructure
Active Bat	Ultrasound, time-of-flight, lateration	Physical	Absolute	Infrastructure
Cricket	Proximity, lateration	Physical	Absolute/ Relative	Infrastructure
Spot-On	Lateration	Physical	Relative	Ad-hoc
Easy Living	Vision, triangulation	Symbol	Absolute	Infrastructure
RADAR	802.11 RF, scene analysis, triangulation	Physical	Absolute	Infrastructure
APS	Proximity, lateration	Physical	Relative	Ad-hoc
APS with AoA	RF, ultrasound, AoA	Physical	Relative	Ad-hoc
Savarese *et al.*	RF, ultrasound, SS map, triangulation	Physical	Relative	Ad-hoc/ Infrastructure
Ladd *et al.*	RF, SS map	Physical	Relative	Infrastructure

Table 3.1: Location-sensing systems classification.

Chapter 4

Cooperative Location-sensing System

There are several efforts that address the challenges of location-sensing in ad-hoc networks, assuming cooperative devices that operate in a self-organizing manner. In this Chapter, we present a system like that, referred as *Cooperative Location-sensing System (CLS)*. We describe its algorithm, which consists of a communication protocol and a voting process, and we discuss the heuristics that is based on.

4.1 Overview

CLS enables devices to estimate their position in a self-organizing manner and without the need of an extensive infrastructure or training. It can also take advantage of the IEEE 802.11 infrastructure, if available. It consists of a communication protocol and a voting process. As we describe in detail, the communication protocol disseminates positioning information and distance estimates among hosts in the network. A host that seeks its position, uses these position information and distance estimates to compute its position in the voting process. The voting scheme aims to incorporate positioning information from hosts in the network and enables them to estimate their position in a self-organizing and adaptive manner.

All the hosts in the network participate in the communication protocol. Depending on whether or not a host can compute its own location running CLS, it is called active or passive. In this thesis, we consider an architecture, where all hosts are active. That is, an active host can be either a laptop or a PDA (Personal Digital Assistant).

CLS runs iteratively at each host. For each iteration, it considers a snapshot of the

network and assumes that all hosts are stationary during that run. An active host reports its position to the other hosts as soon as it computes it during a run. We refer to the duration of an iteration as a *CLS run* or simply *run*. The exchange of CLS messages and voting process take place concurrently during a run.

We use a grid-based representation of the terrain. Each host maintains a regularly-spaced grid of cells. Fig. 4.1(a) illustrates an example of a real terrain and Fig. 4.1(b) illustrates the grid-based representation of the terrain.

At the beginning of a CLS run, hosts initialize their grid and during the voting process, hosts cast votes on the cells. Each host is configured with a voting weight, a constant, that depends on the confidence of the host about its position estimation. We assume that landmarks have higher voting weight than hosts that compute their position using the CLS, since the former know their positions *a priori*, whereas the latter do not know their position but they try to compute it.

For example, Fig. 4.1(c) illustrates three hosts, A, B, and C, which contribute with positioning information, the wireless range of host C (R_C) and the distance interval from u, which is the host that wants to estimate its position, to A. Fig. 4.1(d) illustrates the accumulation of votes on the grid of host u after A, B, and C have casted their votes.

At the end of the voting process, the value of a cell in the grid is the sum of the voting weights of all those hosts that casted votes for that cell during a CLS run. The higher the value of a cell is, the more hosts agree that this is a likely position of that host. A cell is represented using the local grid coordinate system. We assume that all hosts in the terrain share a common coordinate system, used to represent their position.

4.1.1 Design goals

The initial CLS implementation was designed in order to fulfil the following goals.

1. To be robust enough to tolerate multiple network failures (such as device failures or disconnections) and changes in the environment due to host mobility.
2. To be easily extensible to incorporate application-dependent semantics or external (location-related measurements) information.
3. To be computationally inexpensive, so that devices with limited capabilities (such as PDAs or sensors) can participate in the network.
4. To be suitable for indoor and outdoor environments.

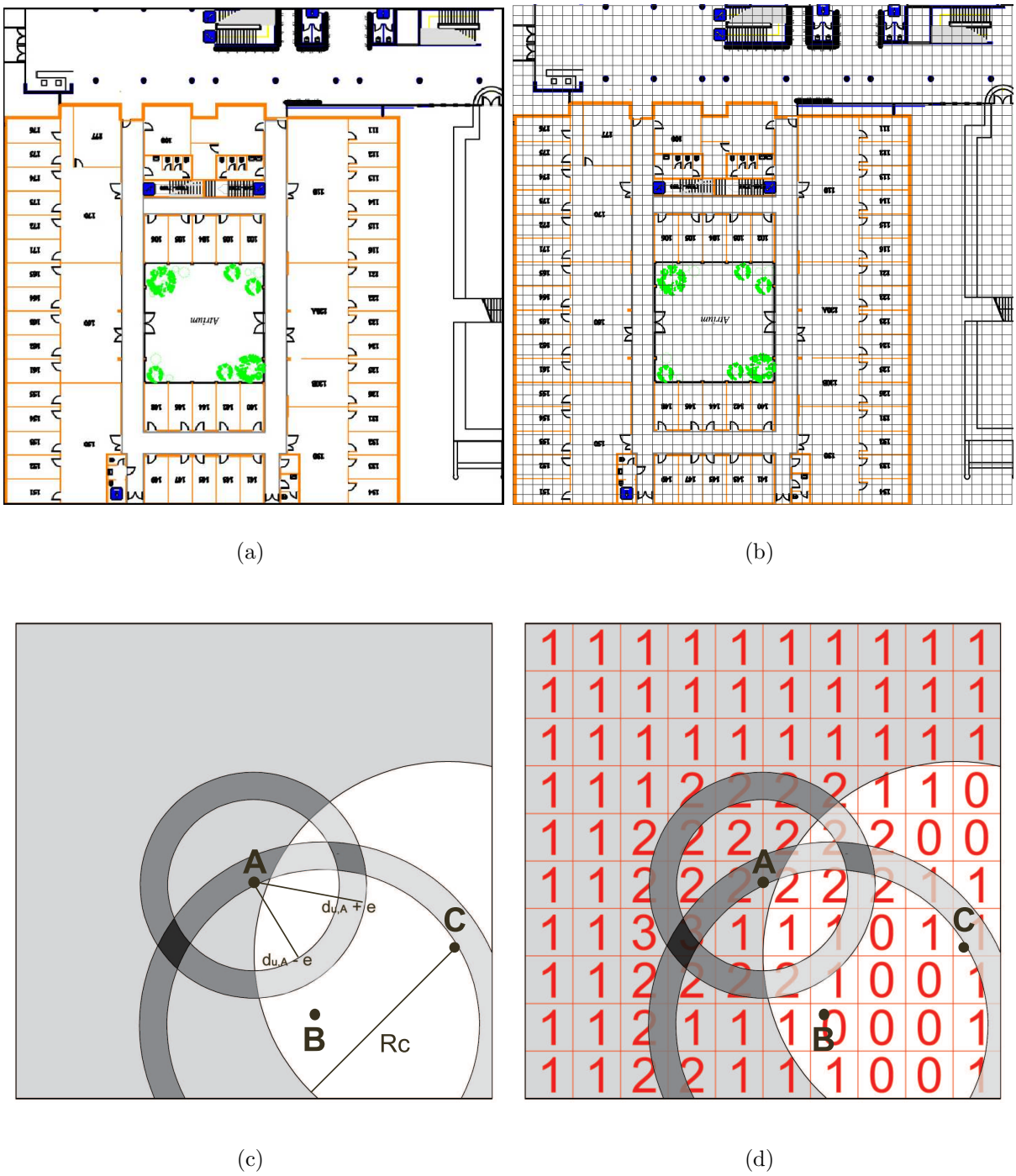


Figure 4.1: Example of (a) a terrain, b) the corresponding grid-based terrain, (c) the positioning information of three hosts A, B, and C, and (d) the accumulation of votes on the grid of host u after A, B, and C have casted their votes.

5. To be scalable, inexpensive, and easily deployable without the need for extensive training and specialized infrastructure.

4.2 Communication protocol

4.2.1 CLS beacon

A CLS run starts with a neighborhood discovery protocol that enables hosts to learn about their one-hop neighbors (i.e., hosts within their transmission range). At the beginning of a run, a host broadcasts a *CLS beacon* with its own positioning information, called *positioning entry*, with a specific signal strength value. A *positioning entry* of a host consists of several fields, namely, its host id (id_n), maximum wireless transmission range (R_n), position coordinates (X_n and Y_n), and voting weight (w_n). The position field may be empty, if the host does not have any estimation yet, or include the position and time it was configured or computed (t_n). We assume that each host is configured with its maximum wireless transmission range. The *CLS beacon* is broadcasted single-hop and thus only hosts within the transmission range of the sender may receive its beacon. Hosts that receive this beacon are required to respond with their own positioning information to announce their presence.

4.2.2 CLS entry

A recipient of such message can estimate its distance from the sender with an available distance estimation method based on the signal strength value that received. For that, CLS assumes a radio propagation model that converts a signal strength value to a distance interval $(d-\epsilon, d+\epsilon)$, where d is the mean distance estimate and ϵ is the range error. A host combines this distance estimate with the position information (*positioning entry*) of the sender. This set of information that a host maintains for a neighboring host is defined as the *CLS entry*. For example, consider host m that receives a CLS beacon from host n that has id id_n, position coordinates X_n and Y_n computed at time t_n, transmission range R_n, and voting weight w_n. Host m computes its distance from n to be the interval $(d^l_{m,n}, d^u_{m,n})$, where $d^l_{m,n}$ is equal to $(d-\epsilon)$ and $d^u_{m,n}$ is equal to $(d-\epsilon)$, and finally generates a *CLS entry* for n equal to $(id_n, X_n, Y_n, t_n, R_n, w_n, (d^l_{m,n}, d^u_{m,n}))$.

4.2.3 CLS update messages

When an active host generates a new CLS entry, it disseminates it in a controlled manner in the network. By controlled dissemination, we mean that active hosts only re-broadcast the updated or new CLS entries to prevent infinite loops and repeated broadcasts of the same or stale information. Stale entries are distinguished based on their time field t_{node_id}. We refer to these messages as *CLS updates*. *CLS updates* enable active hosts to learn about other hosts in the terrain, some of which are more than one-hop away. When a host finds an estimation for its position either for the first time or an update one, it broadcasts a new beacon.

4.2.4 CLS table

A host maintains a table with all the received CLS entries, called *CLS table*. This table is initialized at the beginning of a run and updated when the local host gathers a non-stale CLS update. Note that it contains entries not only for hosts that are within the range of the reference host but also for hosts that the reference host has learned about through updates (from its neighbors). For example, assume that a host, e.g., host m, has not received any beacon or update *directly* from a given host (e.g., host B), but has a positioning entry about it (sent from another host, host C). CLS infers that B is likely not in the wireless range of m. The distance interval between m and B is set to (R_B, ∞), where R_B is B's maximum wireless transmission range. Table 4.1 shows the CLS table of host m with entries for peers A (within range of m) and B (out of range of m).

Node id	Position	Time	Range	Weight	Distance	Vote
A	(X_A, Y_A)	t_A	R_A	w_A	$(d^l_{m,A}, d^u_{m,A})$	positive
B	(X_B, Y_B)	t_B	R_B	w_B	(R_B, ∞)	negative

Table 4.1: CLS table of host m with entries for peers A and B

4.3 Voting process

We represent as G_m the grid that a host m maintains during a run and $v(x, y)$ the value of the cell with coordinates (x, y).The value of the cell reflects the agreement of other hosts

in the network that this cell is a likely position of the local host. The voting process takes place as the local host receives *CLS update* messages and builds its *CLS table*.

We define the *position estimation* of host m from host n ($P_{m,n}$) to be the set of cells in grid G_m that are likely positions of host m given its *CLS entry* about host n (i.e., the position of host n, its wireless transmission range R_n, its voting weight w_n, and their estimated distance interval $(d^l_{m,n}, d^u_{m,n})$).

In case that the two hosts (e.g., hosts m and n) are one-hop neighbours, the set $P_{m,n}$ is equal to:

$$P_{m,n} = \{(x, y) \in G_m,\ d^l_{m,n} \leq \sqrt{(x_n - x)^2 + (y_n - y)^2} \leq d^u_{m,n}\}.$$

If host m only learns about n indirectly, its position from host n is defined as:

$$P_{m,n} = \{(x, y) \in G_m,\ R_n \leq \sqrt{(x_n - x)^2 + (y_n - y)^2}\}.$$

At each run, each host (e.g., host h) gathers updates and attempts to compute its own location performing the following steps:

Step 1: At the beginning of a run, each host initializes the value of the cells in the grid by assigning zero to them:

$$\forall (x, y) \in G_h,\ v(x, y) = 0 \tag{4.1}$$

Step 2: For each CLS entry about a host with known or computed location (e.g., host k), host h performs the following steps:

1. It computes $P_{h,k}$, which is the position estimation of host h from host k.

2. It increases the value of the cells in $P_{h,k}$, by w_k, where w_k is the voting weight of host k:

$$\forall (x, y) \in P_{h,k},\ v(x, y) = v(x, y) + w_k \tag{4.2}$$

Step 3: The set of cells in the grid with maximal values defines a possible region where host h may be located. Landmarks and hosts that computed their own location earlier in a run determine to some extent the location of other hosts and their estimation errors are propagated in the network. To reduce the impact of these errors, we apply the following heuristic. During the voting process in a CLS run, a set of cells has to satisfy the following two conditions in order to be an acceptable solution:

- **ST (Solution Threshold)**: The number of votes in each cell of the potential solution must *be above* a threshold. We refer to this threshold as the voting threshold.

- **LECT (Local Error Control Threshold)**: The size of the grid region (i.e., number of cells with maximal value) that contains the potential solution must *be below* a threshold. We refer to this threshold as the local error control threshold.

In effect, ST controls how many hosts with known location must "agree" with the proposed solution and LECT controls how precise this solution is. Thus, a high ST value reduces the error propagation throughout the network but delays the positioning estimation. The values of ST and LECT also depend on network characteristics such as the density of hosts, ratio of landmarks and hosts with unknown location, and range errors.

Step 4: The local host casts votes from the newly generated CLS entries of its table until the two conditions are satisfied. That is, if the above two conditions fail, the voting process returns to Step 2. The local host may also adaptively relax these thresholds. When the host finds an acceptable set of cells with maximal values, it computes its centroid and terminates the voting process. The centroid's coordinates are the average of the respective coordinates of the cells with maximal values. This centroid becomes the position of host for this CLS run. By selecting the centroid, we minimize the maximum possible error (given that the correct solution lies somewhere in that grid). Nevertheless, by narrowing down an area to one point, the algorithm potentially introduces an error.

If the area of the potential solution is above the LECT, its corresponding centroid will potentially introduce an unacceptable error. In that case, the potential solution will be rejected. The fundamental idea behind both thresholds is that it is likely that other hosts in the network will have higher precision in their position estimates. Therefore, more distance estimations of hosts with known location will be available to increase the voting weight or narrow down the area of the potential solution within acceptable standards.

4.3.1 Example of voting process

We consider hosts A, B and C with known or estimated position and host u that currently tries to position itself. We discuss this example from the perspective of host u, after it received beacons and updates from A and B and updated its CLS table.

Fig. 4.2 shows the grid of host u, after it accumulated the votes from A, B and C. On

the grid, we have superimposed the positions of A, B and C, the distance estimate between A and u $(d_{u,A} - \epsilon, d_{u,A} + \epsilon)$, and the transmission range of C (R_C). Each cell is marked with its value, given that A, B and C's voting weight is one and are the only hosts that have voted so far. The darker an area is, the more voting weight has been accumulated on the corresponding grid cells.

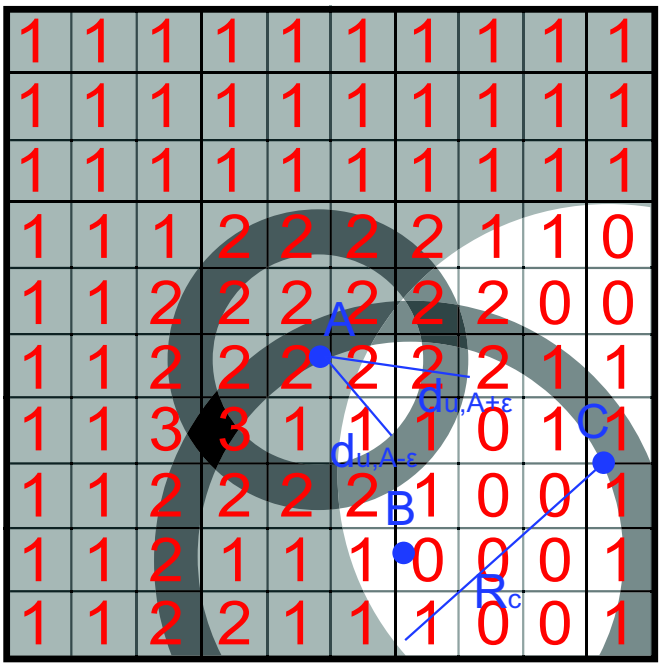


Figure 4.2: Accumulation of votes on the grid of host u (with unknown position) after A, B, and C have casted their votes. A and B contribute with positioning information, whereas u only learns about C from either A or B. The position of hosts A, B, and C, wireless range of C (R_C), and distance interval from u to A are superimposed on the grid.

In this example, we assume that host u has not received any beacon or update from C but learns about C from either A or B's update. It then infers that its position is likely to be outside of the range of C and casts a vote for the cells in the grey area outside of the range of C. The set of cells with the maximal value defines a possible solution. In this example, this set consists of the two cells with value '3'. In order to be accepted, the two conditions of the voting process need to be satisfied. Once a potential solution has been accepted, the algorithm computes the centroid of that set of cells and reports it as the estimated location of the newly solved host. The host does not try to refine further its location estimation during that run, even when it receives additional positioning information from other hosts.

4.4 Discussion of ST and LECT thresholds

As we described in Section 4.3, a host should satisfy the ST and LECT thresholds in order to be an acceptable solution. The algorithm of the system is iterative, therefore in order for the iteration process to be terminated, these thresholds should be satisfied. Specifically, the algorithm is terminated, when an acceptable minimum number of hosts can 'hear' the host and a maximum number of cells with maximal value are found.

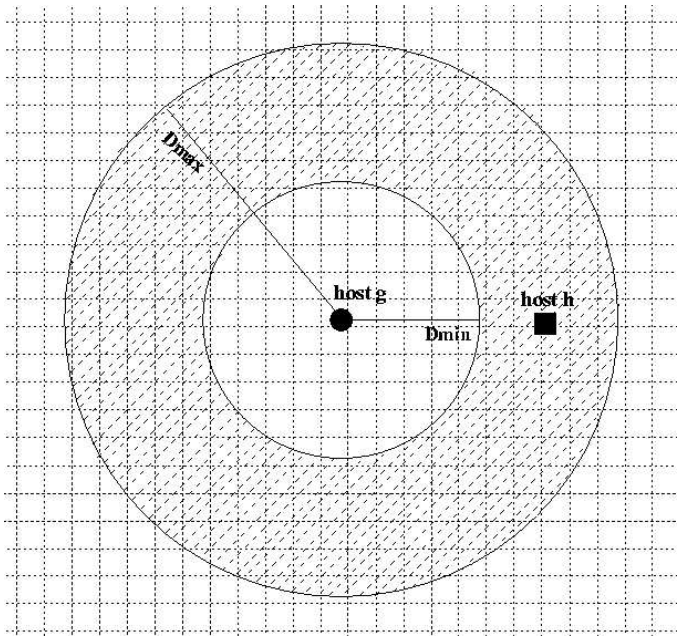
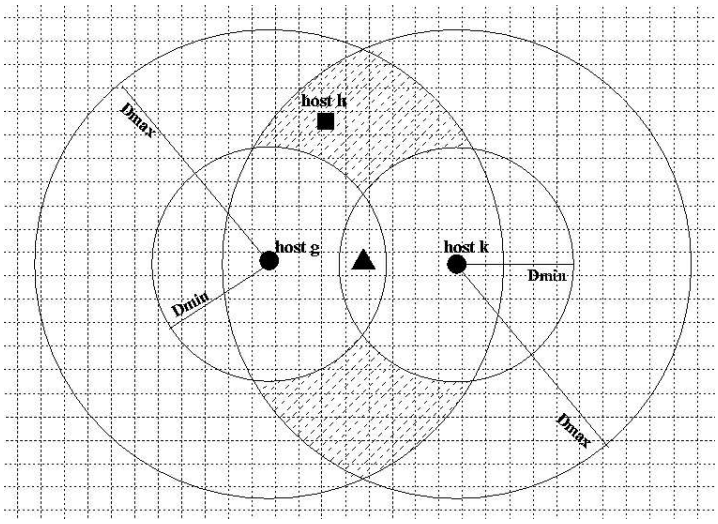


Figure 4.3: Graphical representation of voting process with 1 neighbour host.

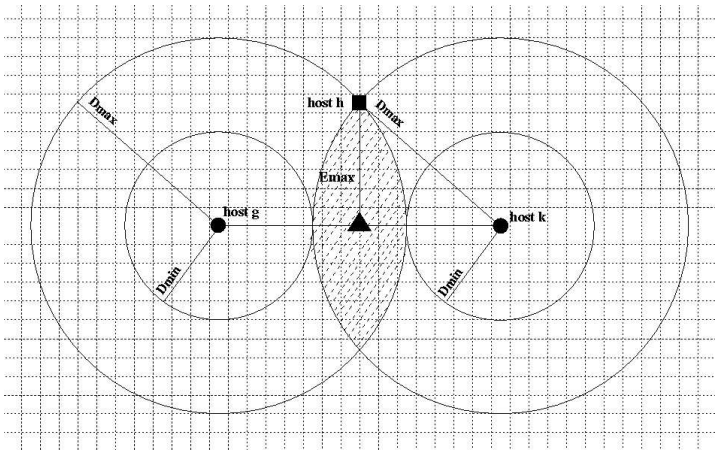
The LECT value depends on the range error ϵ, since ϵ affects the distance estimation of a host and therefore the number of cells that will take a vote from that host. Thus, in order to find an initial LECT value for our experiments we made the following observations:

Firstly, we suppose that the position of host h is defined solely from host g. This means that host h, after receiving beacons from host g, casts votes to the cells of the interval $(d_{h,g} - \epsilon, d_{h,g} + \epsilon)$, as we described in Section 4.2. However, this case is not acceptable, since, as it is illustrated in Fig. 4.3, the true position of host h is located in the solid square, whereas the possible cells after the voting process (those with maximum value) that host h might be located, correspond to the shaded ring. The final position of host h, that is, the centroid of these cells, resides in the center of the ring, where host g, and not h, is located. Thus, the position of a host cannot be found solely from one neighbour.

Secondly, if the position of host h is computed from two other hosts, e.g., host g and host k, we observe two possible cases. In the first case, as it is shown in Fig. 4.4(a), there are



(a)



(b)

Figure 4.4: Graphical representation of voting process with 2 neighbour hosts, in case of (a) two different regions of cells with maximum value and in case of (a) one region of cells with maximum value.

two regions of cells with maximum value (shaded cells), that host h can possibly reside. We suppose that the true position of host h is depicted with the solid square. However, neither this case is acceptable since the centroid of these cells would be outside of the shaded rings, which is depicted with the solid triangle. In the second case, as it is shown in Fig. 4.4(b), there is only one region of cells that host h can possibly reside. The shaded cells define the area, after the voting process, where host h is possibly located. We suppose that, in the worst case, host h is located in the solid square and the final position of host h, that is, the

centroid of these cells, resides in the center of the shaded area, depicted with the solid triangle. Therefore, the maximum location error ($Emax$) would be the distance between the true and the estimated position, that is, equal to $\sqrt{D_{max}^2 - \frac{d_{g,k}^2}{2}} = \sqrt{D_{max}^2 - (D_{min} + \epsilon)^2}$. However, as we will see further down, this distance error is prohibitive, thus neither this case is acceptable.

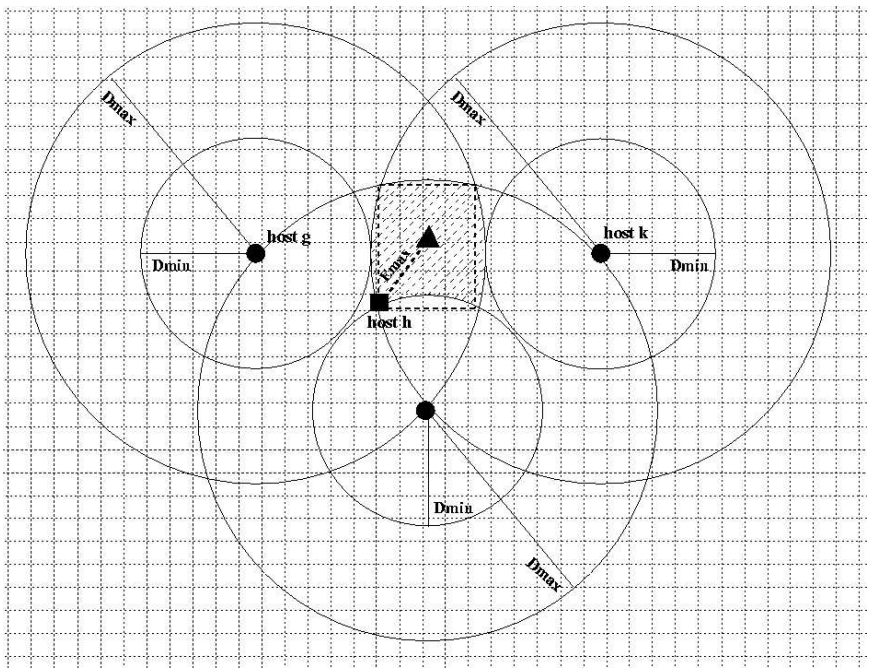


Figure 4.5: Graphical representation of voting process with 3 neighbour hosts.

Finally, in case the position of host h is defined from three hosts instead of two, that is, from host g, host k and host m, the number of cells in the grid, with maximum value, where host h possibly resides, are shown in Fig. 4.5 with the shaded cells. In this case, the three circles that intersect, transform an area of $(2\epsilon + 1)^2$ unit cells. This is proved if we consider that each ring $(D_{max} - D_{min})$ has width of $2\epsilon + 1$ cell units, since $D_{max} - D_{min} = (D_{h,neighbour\_host} + \epsilon) - (D_{h,neighbour\_host} - \epsilon) = (2\epsilon)$ cells. Thus, there are $2\epsilon + 1$ cells (1 is the estimated cell). We suppose that, in the worst case, host h is located in the solid square and the final position of host h, that is, the centroid of these cells, resides in the center of the shaded area, depicted with the solid triangle. Therefore, the maximum location error ($Emax$) would be the distance between the true and the estimated position, that is, equal to $\sqrt{\frac{2\epsilon + 1^2}{2}}$, which is an acceptable case, as we will see further down. Thus, when three rings are intersected, an average value for the LECT threshold is equal to $(2\epsilon + 1)^2$. As the number of neighbours of host h is increased, the number of the intersected cells is getting

lower and the LECT is satisfied.

4.4.1 Example of ST and LECT

We evaluate the impact of ST and LECT thresholds on the accuracy level of our system. Specifically, we consider the case of 100 hosts, 10% of which are landmarks, and set the transmission range equal to 20 meters and the range error equal to 10% of the transmission range, that is, equal to 2 meters. We also set the cell's area equal to 1 square meter. Considering the case depicted in Fig. 4.4(b), the size of the area in which host h can reside is equal to 60 cells. The centroid of these cells place host h at the solid triangle point, whereas in the worst case its true position is in the solid square point. This means that, for the above range error, the location error will be equal to $E_{max} = 6.5$ cells $= 6.5$ meters. Given that we aim for better accuracy, this location error is large.

Considering the case depicted in Fig. 4.5, maintaining the same values for the range error, the possible cells that host h resides are equal to $(2\epsilon + 1)^2 = (2 \cdot 2 + 1)^2 = 25$. The centroid of these cells place host h at the solid triangle point, whereas in the worst case its true position is in the solid square point. This means that, considering the above range error, the location error will be equal to $E_{max} = 3.5$ cells $= 3.5$ meters. We consider this location error acceptable. Thus, the choice of 25 cells comprises to a maximum bound, namely our LECT value.

On the other hand, the total number of landmarks that reside in the simulation area and the average degree of connectivity have a great impact on the ST value. A host receives votes from hosts and landmarks within its range but also from hosts and landmarks outside its range through the *CLS updates*, as described in Section 4.2. Therefore, an acceptable initial ST threshold is equal to the following equation 4.3, since we suppose that eventually each host will receive votes from every landmark and every other host due to the *CLS updates*:

$$w_{all\_landmarks} + w_{all\_other\_hosts} \tag{4.3}$$

At this point, we should note that during a CLS run, ST and LECT can be adjusted dynamically to deal with poor network conditions and low density of landmarks. Specifically, a CLS host assigns initial values on the ST and LECT thresholds based on the density of landmarks in the terrain, as we described above. However, after a number of failed

attempts to satisfy them, it iteratively relaxes them. In this way, we tradeoff precision with accuracy. We use the following heuristic:

1. During a run, each local host keeps a counter of the number of times that either one of the two condition tests fails.

2. When such failure occurs, the host increases the counter and resets the ST and LECT thresholds. Specifically, it updates the ST threshold by dividing it with this counter. Also, it multiplies the initial LECT threshold with this counter.

4.5 Privacy issues

The privacy requirement is a critical issue in location-sensing. To enable cooperation among peers and CLS servers, an authentication mechanism is required. In particular, CLS servers can distribute an encryption key (public key) to every peer and can give a decryption key (private key), to each peer which should be kept secret. The decryption function applied to a plaintext message yields a ciphertext message that any peer can decode using the corresponding encryption function, but can only have been created by the peer with the private key, that is, the message has been digitally signed.

4.6 Summary

In this Chapter, we presented CLS, a location-sensing system, which is based on the cooperation among hosts and operates in a self-organizing manner. We described the main algorithm, which consists of a communication protocol and a voting process. In particular, the communication protocol disseminates position estimates among hosts in the network and the voting process aims to incorporate positioning information from other hosts and enable them to estimate their position in a self-organizing and adaptive manner. Finally, we discussed two main thresholds of CLS and their use in the position estimations.

Chapter 5

Performance Analysis of CLS in Stationary Settings

In this Chapter, we study the performance of CLS when all the hosts are stationary and there is no mobility. The impact of mobility on CLS will be studied in Ch. 7. To evaluate the performance of the system in a stationary environment, we run simulations, using *ns-2*, a popular tool that can incorporate mobility and wireless networking. Specifically, we investigate the impact of the following parameters on the performance of the system: range error (i.e., distance estimation error), ST and LECT values, connectivity degree and percentage of landmarks, and grid size.

Performance metric: Accuracy

The *accuracy level* is the main metric used in the performance analysis of CLS. We define the accuracy level to be (α, ϵ), if the α percentage of hosts can estimate their location with an error of at most ϵ units. This percentage considers all hosts in the terrain excluding landmarks. In the location-sensing area, distance error denotes accuracy and the percentage of hosts, that present this accuracy, denotes precision. Therefore, the metric we use takes account of both accuracy and precision.

Simulation test-bed

We evaluate CLS, using a popular simulation tool, namely the *network simulator 2* (*ns-2*) [35], which is targeted at networking research. We consider a flat grid of 100x100 cell units in size. We place 100 nodes, randomly, using a pseudo-random generator which selects

an integer (in place of X and Y coordinates of the nodes), using a uniform distribution. A node corresponds to a wireless device that runs CLS. We use the terms node, host, and wireless device, interchangeably. The 10% of nodes are landmarks, which are also chosen randomly. That is, 10 nodes out of 100 are landmarks and the rest 90 are hosts with unknown position. The average connectivity degree of the nodes is equal to 10. This means that each node has 10 one-hop neighbours on the average. We achieve this connectivity degree by setting the transmission range equal to 20 units. In particular, the area of the grid is equal to 100x100 = 10.000 square units. We assume that the transmission range covers a circular area equal to $\pi \cdot R^2$, where R is the transmission range. Since there are 100 nodes and each node has 10 one-hop neighbours on the average, the total area should be covered by at least 10 circles, with each one's radius being equal to the transmission range. Thus, each circular area is equal to 10000 / 10 = 1000 square units, and the radius R that corresponds to the transmission range, is equal to $\pi \cdot R^2 = 1000 \Leftrightarrow R \simeq 18$ units. However, since we assumed that the flat grid is a square and not a circle, we set the transmission range equal to 20 units, in order to cover the entire area.

Both landmarks and hosts with unknown position have the same transmission range. Moreover, we set the voting weight of a landmark equal to the total number of all nodes in the terrain, that is, 100 and the voting number of the other hosts equal to one. Both the location error and range error are specified as percentages of the transmission range. For example, 50% location error means an error equal to half of the transmission range of the wireless infrastructure. We assume a range error of 10% of transmission range unless otherwise stated. In every experiment, we run CLS 100 times, unless otherwise stated, in order to achieve more tight results.

5.1 Impact of range error

In real environments, each node estimates its position with the aid of the signal strength values that receives from other nodes. However, in order to simulate a real environment, we assume a radio propagation model that converts a signal strength value to a distance interval. In particular, the distance interval is formed by the bounds (d - ϵ, d + ϵ), where d is the distance estimate and ϵ is the range error. The range error is specified as a percentage of transmission range. Thus, we run the simulations with different range errors

and we observe their impact in the accuracy level. We choose range errors of 10%, 20% and 30% of the transmission range. This choice is based on an extensive empirical study that we have made in ICS-FORTH, as Ch. 8 describes. Particularly, our study shows that 90% of the cells correspond to 10% range error of the transmission range or less and the maximum range error is 23.5% of the transmission range, appearing in some cells due to the topology of the environment. In our simulations, the range error values correspond to a low and upper bound of a typical range error of a deployed IEEE 802.11 network.

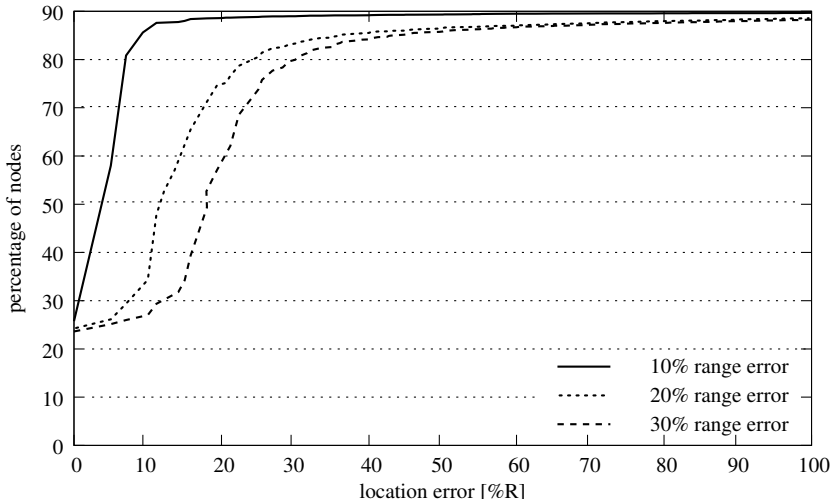


Figure 5.1: Accuracy level as a function of the range error. 10% of nodes are landmarks.

Fig. 5.1 shows the accuracy level for range errors of 10%, 20% and 30% of the transmission range. In this set of simulations the transmission range was set equal to 20 meters and thus we assumed range errors of 2, 4 and 6 meters, respectively. As it is illustrated in this figure, in case of 10% range error, 85% of the nodes have at most 10% location error, i.e., for transmission range equal to 20 meters, the location error is at most 2 meters. Respectively, in case of 20% range error, 80% of the nodes have at most 20% location error, i.e., for transmission range equal to 20 meters, the location error is at most 4 meters, and in case of 30% range error, 80% of the nodes have at most 30% location error, i.e., for transmission range equal to 20 meters, the location error is at most 6 meters.

Fig. 5.2 illustrates the impact of range error on the accuracy of CLS system. The configuration setting in this figure corresponds to a network with 10% landmarks and average connectivity degree of 12. It presents the average location error, for all hosts with unknown position, when the range error varies from 5% to 50% of the maximum transmission range. We observe that when the range error is doubled, the location error

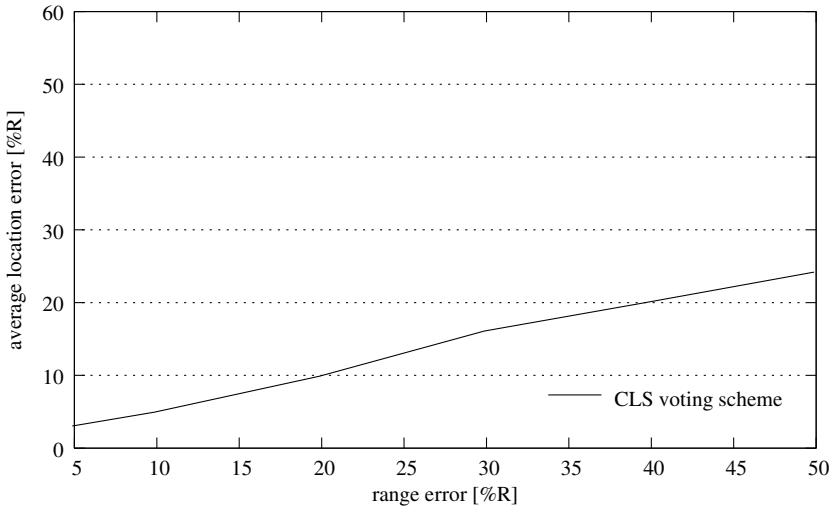


Figure 5.2: Impact of range error on CLS performance. 10% of nodes are landmarks.

is also doubled. In addition, for a high range error, that is, equal to the half of the transmission range (50% of the transmission range), the average position error is 25% of the transmission range. As mentioned earlier, the percentage of hosts for the average position error in a setting includes only the hosts with unknown positions (90 hosts in this example). In addition, for range error equal to 10%, 20% and 30% of the transmission range, the position error is 5%, 10% and 18% of the transmission range, respectively.

5.2 Impact of ST and LECT thresholds

In Section 4.4, we discussed how ST and LECT affect the accuracy of CLS. At this point, we test through simulations the impact of these thresholds. In particular, using the simulation setting of Section 5, we conclude that each host has 10 one-hop neighbours whereof 10% are landmarks, that is, each host has one landmark and other 9 other non-landmark hosts as neighbours. Since, by definition, a landmark knows or has estimated its position accurately, we assign to its vote a higher weight than in the case of simple nodes, to indicate its higher accuracy level and confidence of its location estimates. Thus, the voting weight of a landmark is set to the total number of hosts (including landmarks) in the terrain, that is equal to 100, and the voting weight of a node to one. Therefore, according to the equation 4.3, that we described in Section 4.4, an acceptable initial ST threshold is equal to $1000 + 89 = 1089$.

Thus, we set the LECT equal to 25 and the ST equal to 1089, reducing it gradually.

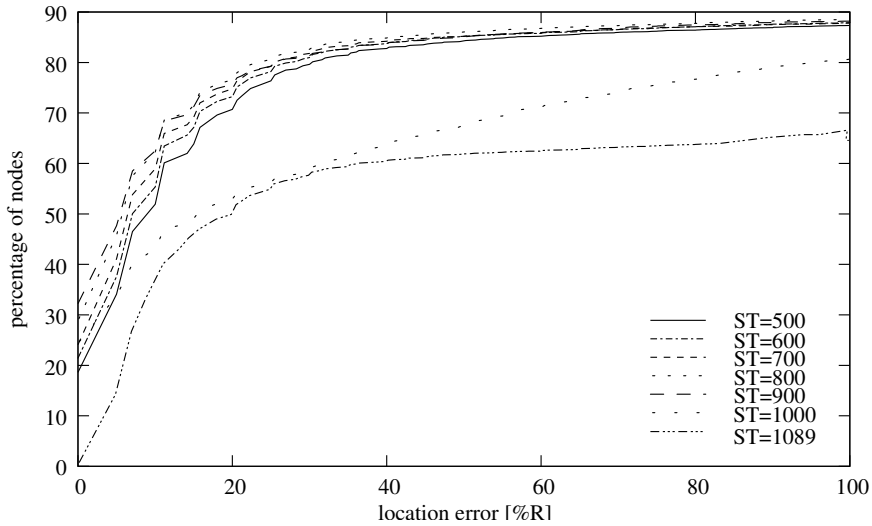


Figure 5.3: Accuracy level for LECT = 25 and ST = 1089, 1000, 900, 800, 700, 600 and 500.

Fig. 5.3 depicts the accuracy level for LECT equal to 25 and ST values of 1089, 1000, 900, 800, 700, 600 and 500. For instance, the ST value equal to 1000 means that in order for the solution to be acceptable, there must be at least 10 landmarks that can cast a vote for host h, considering that the voting weight of a landmark is 100. These landmarks can be either one-hop away from host h or can hear host h, not directly, but through another host. Thus, in Fig. 5.3, we notice that for a fixed LECT, the best ST value is 800. As mentioned above, this value means that there must be at least 8 landmarks that hear directly or indirectly host h, so as the estimated position is considered acceptable. For values close to 1000 the accuracy level declines rapidly since there are nodes that can not hear more than 9 landmarks, even in indirect way. Moreover, for values close to 700 and below it, the accuracy level declines gradually, because if host h can hear at most 7 landmarks, the votes that casts in its grid are not adequate enough in order to find its precise position.

When we found the best value for the ST threshold from the selected ones, so as for CLS to present the best accuracy, we kept this value fixed and varied the LECT values, so as to find the best value for this threshold, as well. Fig. 5.4 shows the accuracy level for ST equal to 800 and LECT values equal to 2, 5, 10, 20, 30, 40, 50 and 60 respectively.

We conclude that the optimal value for LECT threshold is 5. This value means that if we set the number of cells with a maximal value to be 5 cells, we have the best accuracy level, that is, the least location error. In effect, since the centroid is going to be calculated from only these 5 cells, there would be a very little variation between the estimated location and the real one. On the contrary, as the LECT value increases, the accuracy

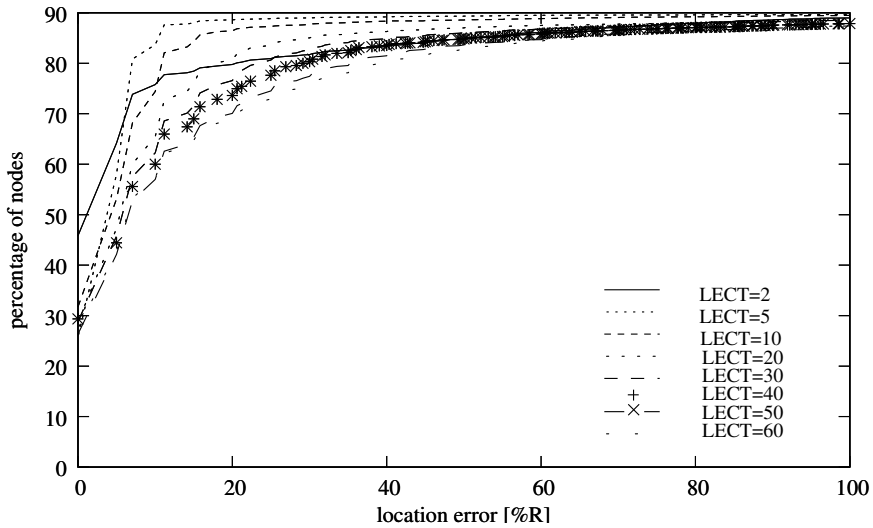


Figure 5.4: Accuracy level for ST = 800 and LECT = 2, 5, 10, 20, 30, 40, 50, 60.

level declines, since the centroid (final position) will be calculated from a greater number of cells, respectively. This means that there is a greater possibility for wrong estimations. On the other hand, when we set the LECT value equal to 2 the accuracy level of the system declines. This means that some nodes could not satisfy this threshold, namely, the number of cells with maximal value in their grid to be at most 2.

As it seems, our initial choice was correct. When we set the LECT value to 25 we take a satisfactory accuracy level. However, since we considered that each host can hear at least 8 landmarks, the shaded area in Fig. 4.5 would be even smaller.

5.3 Impact of connectivity degree and percentage of landmarks

We studied the performance of the CLS system regarding the average connectivity degree, that is, the one-hop neighbours that a host has, and the percentage of landmarks which are placed in the simulation terrain. In order to conduct the simulations we maintain fixed the total number of the nodes equal to 100 and the range error equal to 5% of the transmission range.

We tested CLS for average connectivity degree equal to 3, 10, 12, 16, 20 and 25. In order to achieve the respective connectivity degree we set the transmission range equal to 10, 20, 25, 40, 50 and 60 units, respectively. In addition, for every connectivity degree we varied the percentage of landmarks to 5%, 10% and 20% of the total number of nodes.

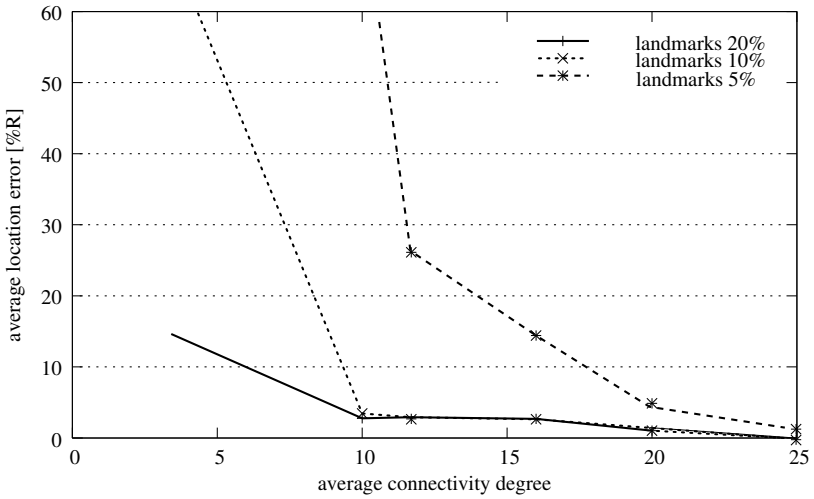


Figure 5.5: Accuracy level of CLS for various connectivity degrees and percentage of landmarks. The range error is 5%.

Fig. 5.5 shows the average location error of CLS for different average degrees of connectivity and percentage of landmarks. Our results show that for low connectivity degree or few landmarks the location error is bad, but in case of 10% or more landmarks and connectivity degree of at least 7, the location error is reduced considerably. In general, it is illustrated that the percentage of landmarks affects strongly the accuracy level of CLS.

5.4 Impact of grid size

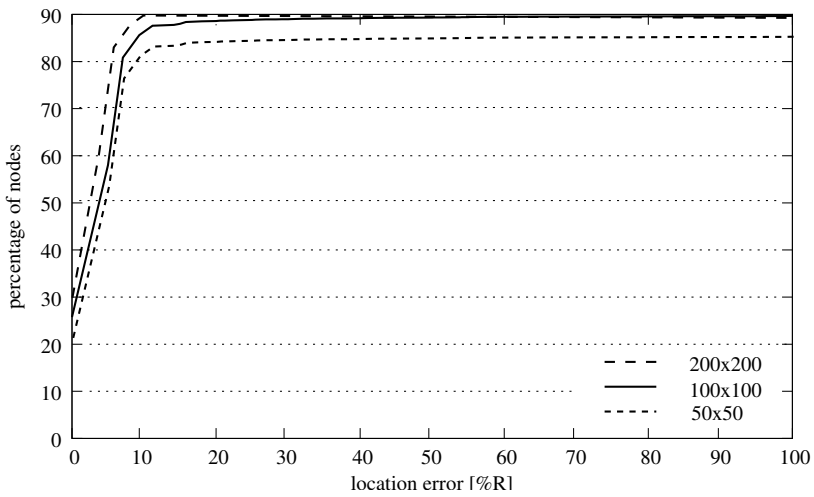


Figure 5.6: Accuracy level of CLS for a grid resolution of 50x50, 100x100 and 200x200 cells.

The grid size of the voting algorithm affects the accuracy level. The finer the grid, the higher the accuracy but also the memory and CPU power requirements. Fig. 5.6

illustrates the accuracy level for a grid resolution of 50x50, 100x100, and 200x200 cells. Compared to 200x200 and 50x50, a grid size of 100x100 is a reasonable choice to balance the computational complexity and accuracy requirements.

5.5 Evaluation on ns-2 vs. the initial simulation platform

As we briefly described in Ch. 1, H. Fretzagias and M. Papadopouli designed CLS. They evaluated its performance via simulations using the platform that Fretzagias had developed at UNC. Since *ns-2* is a popular tool that can incorporate mobility and wireless networking, we decided to re-implement CLS in *ns-2*.

5.5.1 Impact of range error

First, we compared the accuracy level of the two implementations of CLS for range errors of 10%, 20% and 30%. The configuration setting in both figures (Fig. 5.7(a) - Fig. 5.7(b)), corresponds to a network with 10% landmarks and average connectivity degree equal to 10 (including landmarks).

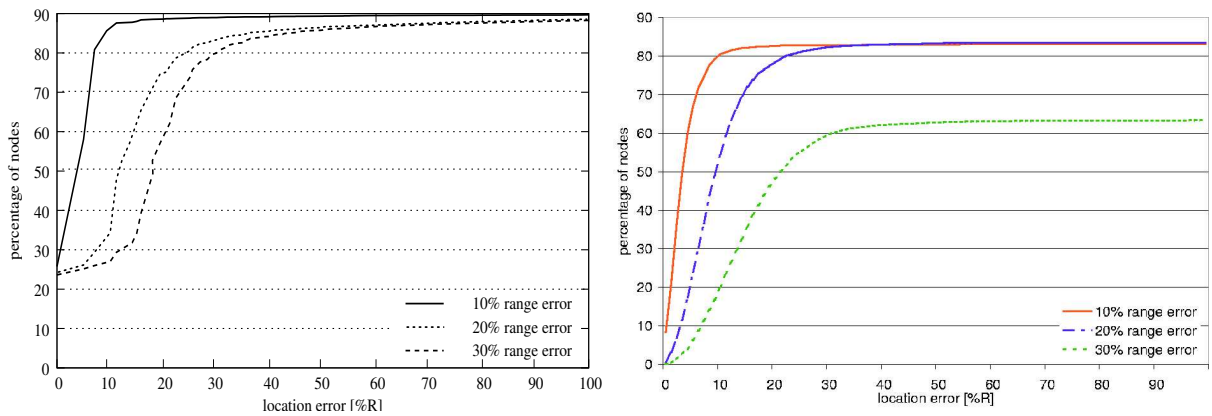


Figure 5.7: (a) CLS implementation in *ns-2* vs. (b) the initial implementation of CLS: accuracy level as a function of the range error. 10% of hosts are landmarks and average connectivity degree is equal to 10.

As Fig. 5.7(a) and Fig. 5.7(b) depict, in our implementation CLS performs better

compared to the initial implementation. For range error of 10% of transmission range, 80% of nodes have at most 8% location error in both simulation studies. Similarly, for range error of 20%, 80% of nodes have at most 22% location error in both simulation studies. Although for range errors of 10% and 20% the accuracy level is a little better in case of our implementation, for range error 30%, the accuracy level of our implementation outperforms by far the initial one. In particular, for range error equal to 10%, 80% of nodes have at most 30% location error in our simulation study, whereas in the initial simulation study only 60% of nodes have at most 30% location error. The configuration settings were kept the same varying only the ST and LECT. In all the subsequent experiments, we keep the ST and LECT values fixed and equal to 800 and 5, respectively (Section 5.2). In the earlier implementation [15], we did not have any reference on the exact choice of LECT and ST.

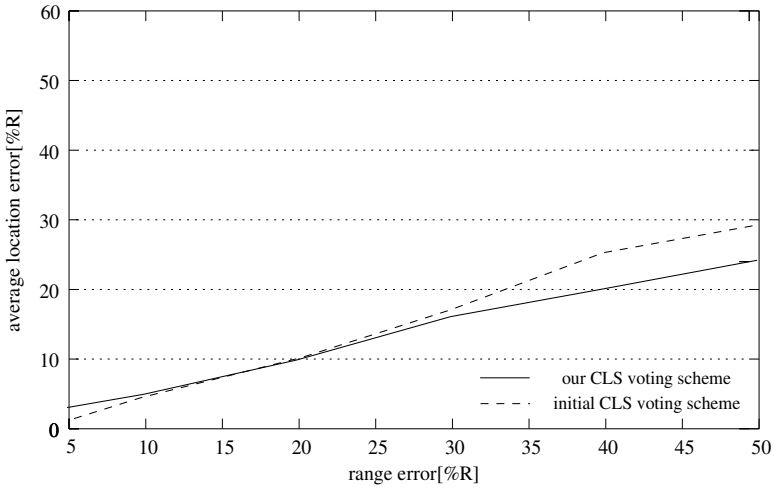
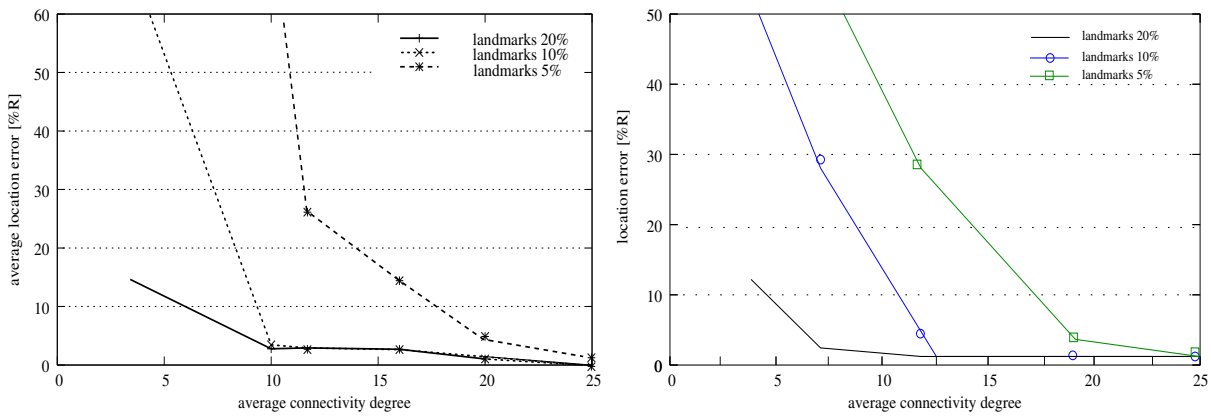


Figure 5.8: CLS implementation in *ns-2* vs. the initial implementation of CLS: impact of range error. 10% of hosts are landmarks and average connectivity degree is equal to 12.

Fig. 5.8 illustrates the impact of range error on the accuracy of both implementations of CLS system. In both cases, the configuration setting corresponds to a network with 10% landmarks and average connectivity degree equal to 12 (including the landmarks). It presents the average location error for all hosts with unknown position, when the range error varies from 5% to 50% of the maximum transmission range. For range errors less or equal to 30% of the transmission range, the location error presents the same impact for both implementations. For instance, for range error equal to 10%, 20% and 30% of the transmission range, the average location error is 5%, 10% and 18% of the transmission

range, respectively, in both simulation studies. Nevertheless, for a greater range error, our implementation outperforms the initial implementation, for the same reasons we describe above. For instance, for range error equal to 40% and 50% of the transmission range, the average location error in our simulation study is 20% and 25% of the transmission range, respectively, whereas for the same percentages of range errors, the average location error in the initial simulation study is 26% and 29% of the transmission range, respectively.

5.5.2 Impact of connectivity degree and percentage of landmarks



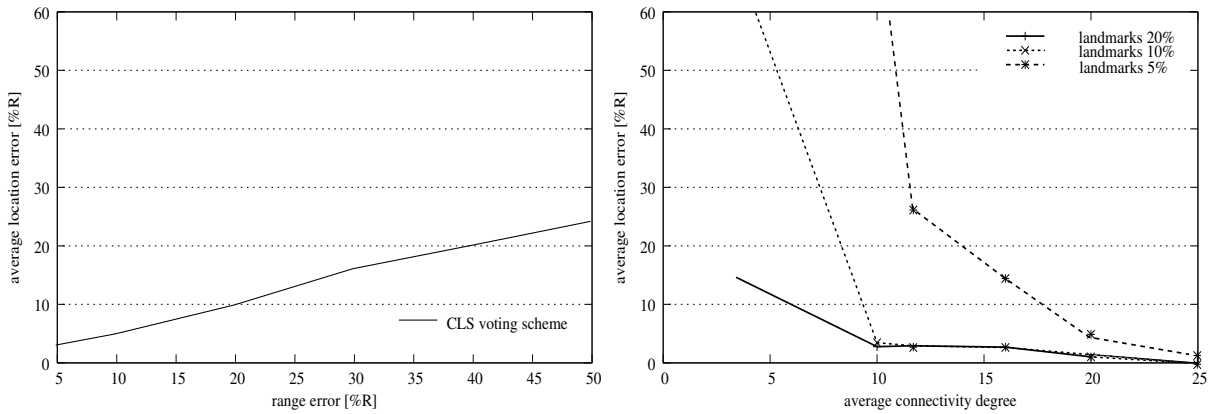
(a) (b)

Figure 5.9: (a) CLS implementation in *ns-2* vs. (b) the initial implementation of CLS: impact of various connectivity degrees and landmark %. The range error is 5%.

We compared the average location error of the two implementations of CLS system for different degrees of connectivity and percentage of landmarks. In both cases the configuration setting corresponds to 5% range error. In particular, we tested both implementations for average connectivity degree from 5 to 25. Fig. 5.9(a) and Fig. 5.9(b) depict that the location error is affected with the same way regarding different degrees of connectivity and percentage of landmarks in both cases. For instance, in both simulation studies, the figures illustrate that for low connectivity degree or few landmarks the location error is bad, but in case of 10% or more landmarks and connectivity degree of at least 7, the location error is reduced considerably.

5.6 Sensitivity study

In this Section, we compare, quantitatively and qualitatively, the three parameters that affect the average location error of CLS, namely, the range error, the connectivity degree and the percentage of landmarks.



(a) (b)

Figure 5.10: Sensitivity study of the impact of (a) range error and (b) various connectivity degrees and landmark % on the CLS performance.

After observing Fig. 5.10(a) and Fig. 5.10(b), we concluded that the number of landmarks has a high impact on the average location error. More particularly, when the number of landmarks is higher than 10% the location error decreases dramatically. In addition, the average connectivity degree has a higher impact on the average location error when the percentage of landmarks is less than a threshold of 10%. For instance, a 50% increase in the average connectivity degree assuming 10% landmarks, corresponds to a decrease of a location error higher than 50%, whereas in case of 5% landmarks the decrease is more dramatic.

Furthermore, another borderline on the accuracy of CLS is when the number of landmarks is equal or more than 10% and the average connectivity degree is more than 10. Assuming the above setting, we observed that the range error has a greater impact on the average location error than the average connectivity degree. For instance, a 50% decrease in the range error corresponds to a 50% decrease of the location error, whereas in case of 10% landmarks and average connectivity degree more than 10%, the decrease is less than 50%.

5.7 Summary

In this Chapter, we studied the performance of CLS when all the hosts are stationary. Specifically, we investigated the impact of range error, the connectivity degree and percentage of landmarks, the grid size, and of ST and LECT values on the performance of the system. Considering 10% of hosts to be landmarks, average connectivity degree equal to 10, and range error equal to 10% of the transmission range, 80% of the hosts can position themselves with a location error of at most 1.5 meters. We observed that when the range error is doubled, the location error is also doubled. Finally, for low connectivity degree or few landmarks the location error is bad, but in case of 10% or more landmarks and connectivity degree of at least 7, the location error is reduced considerably. We concluded that the percentage of landmarks affects strongly the accuracy level of CLS.

Chapter 6

Extended CLS

CLS system can incorporate application-dependent or external information in the algorithm to further refine the accuracy of the measurements. This can take place in the form of virtual landmarks casting additional votes. In particular, we use real-life signal strength measurements and we incorporate them in the simulation testbed. The voting process that we described in Section 4.3 can be extended to consider a more sophisticated signal propagation model than the ideal disk model that we assume. Thus, each host estimates its position depending on the signal strength measurements and then disseminates its position information to the other hosts. In this Chapter, we discuss and evaluate such extension, namely, the use of a signal strength map of the environment, and we refer to this extension as *extended CLS*.

We introduce a training phase (Section 6.2) during which a signal strength map is built using signal strength information received from the IEEE 802.11 infrastructure. During the voting process, a host considers the signal strength map information in addition to its distance measurements. Hosts could acquire these maps via a simple resource discovery protocol. We evaluate the extended CLS system in the estimation phase (Section 6.3), using two algorithms. One based on a median signal strength threshold and one based on the confidence intervals of the measured signal strength values. To avoid the time-consuming procedure in the training phase, we use the method of cubic interpolation, among other interpolation methods (Section 6.3.4).

6.1 Simulation testbed

Our experimental testbed is the Institute of Computer Science, which resides in the first floor of Foundation for Research and Technology (FORTH), in Heraklion of Crete, Hellas. The layout of our testbed is shown in Fig. 6.1. The testbed has dimension of 50 meters by 50 meters, an area of 2500 square meters. We divided it in 50x50 cells and each cell has 1 x 1 square meters. Thus, our testbed is divided in 2500 cells. In our testbed, there are 5 APs of the IEEE 802.11 infrastructure, at the locations indicated with the star marks in Fig. 6.1.

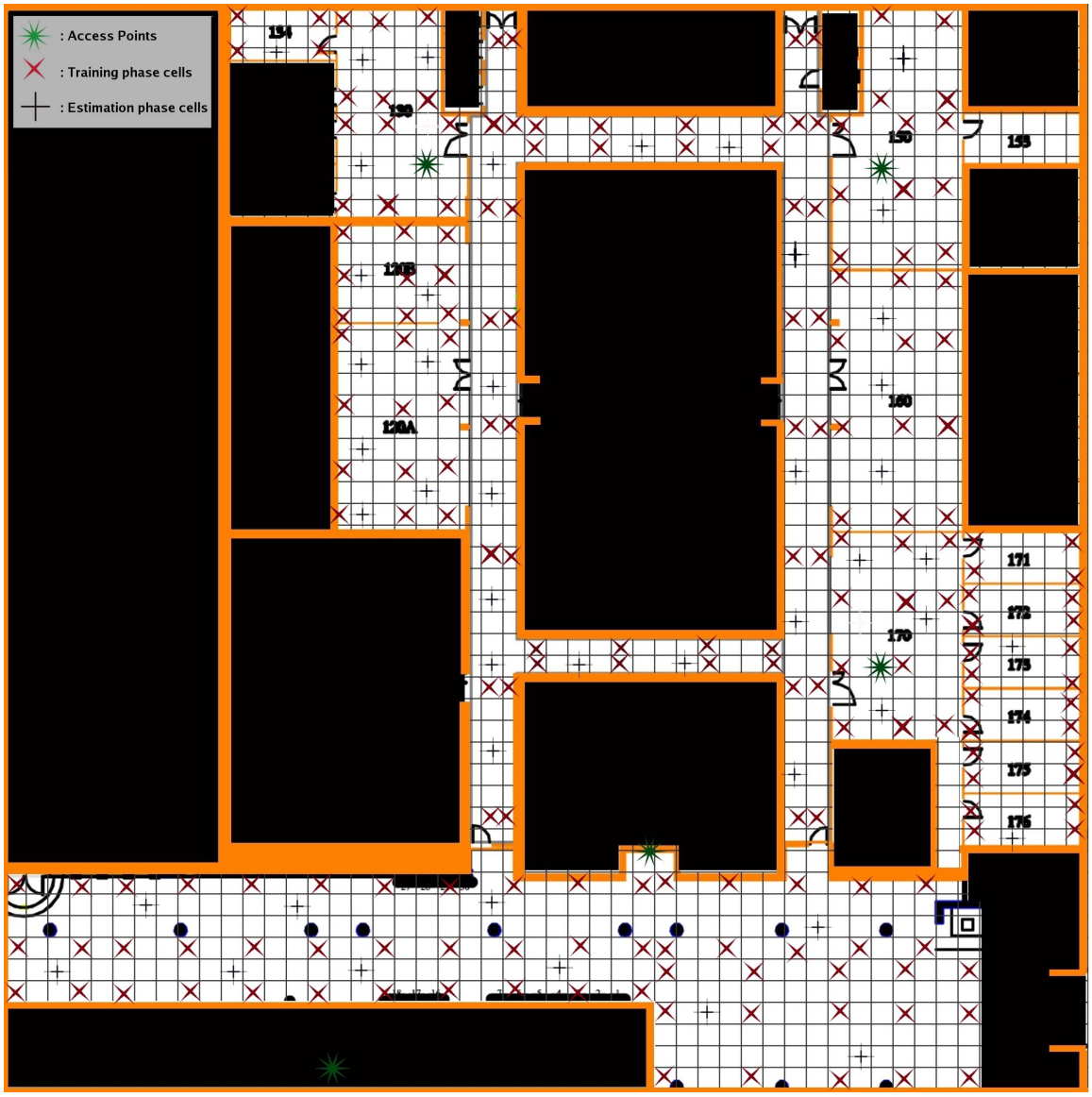


Figure 6.1: Floorplan of the first floor of ICS-FORTH. It shows the 5 APs (stars) and the positions at which signal strength was measured for the training and estimation phase ('X' marks and cross '+' marks, respectively).

Each base station is a Cisco Aironet 1231 b/g Access Point. The APs of the network operate in the 2.4 GHz licensefree ISM (Industrial, Scientific and Medical) band. They have a raw data rate of 2 Mbps and a one-way delay of 1-2 ms. The transmission power in the IEEE 802.11b and the IEEE 802.11g is 50mW and 30 mW, respectively. The data rates supported are between 1 - 54 Mbps. Our mobile host, carried by the user being tracked, was a Pentium-based laptop computer running Microsoft Windows Professional 2000. The mobile host was equipped with a WiFi Orinoco Gold 11 Mbps network interface card (NIC), based on Lucent technology.

The specifications of the APs refer that the range of the network, is 200 m, 50 m, and 25 m, respectively, for open, semi-open, and closed office environments. This classification is based on the type and density of obstructions between the transmitter and the receiver. Thus, our testbed environment would be classified as being open along the hallways where the base stations are located and closed elsewhere. The base stations provide overlapping coverage in portions of the floor, and together cover the entire floor.

The white cells show the total area where the measurements and the simulations were performed. In particular, our measurements were conducted in two phases, which are described further down. We used a wireless network auditing software, called NetStumbler [36], in order to measure and record the signal strength values from the APs.

6.1.1 Signal strength measurements via NetStumbler

NetStumbler is a freeware tool for Microsoft Windows that allows you to detect Wireless Local Area Networks (WLANs) using 802.11b, 802.11a and 802.11g standards.

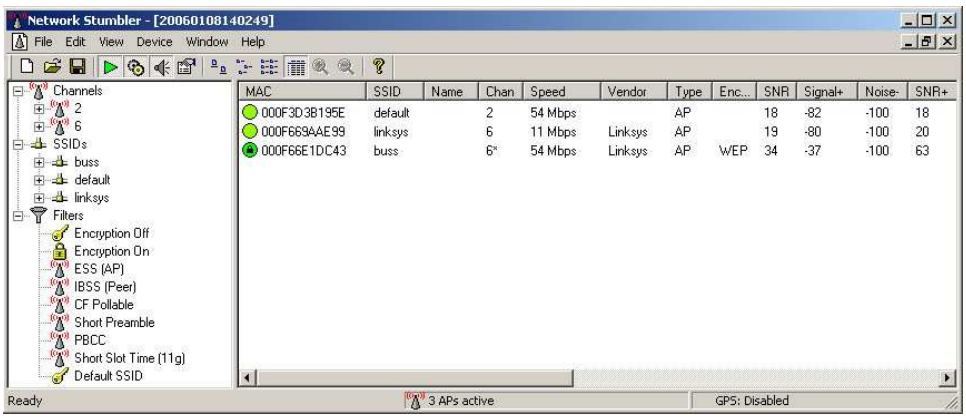


Figure 6.2: A screenshot of Netstumbler detecting three networks: default, linksys, and buss.

Its main use is network monitoring. Some of its common uses concern wardriving, verifying network configurations, finding locations with poor coverage in one's WLAN, detecting causes of wireless interference, detecting unauthorized access points, and aiming directional antennas for long-haul WLAN links. Fig. 6.2 depicts a screenshot of Netstumbler detecting three networks: default, linksys, and buss.

6.2 Training phase

In the first phase, i.e., *training phase*, we collected signal strength measurements in our floorplan from the APs and we built the signal strength map. A *signal strength map* is a grid-based map in which each cell contains the signal strength information for each AP in the corresponding position.

6.2.1 Data collection

A key step in our research methodology is the data collection phase. We record information about the radio signal as a function of the user's location. We use the signal information to infer the location of a user in real time. Among other information, the NIC makes available the signal strength values. SS is reported in units of dBm. A signal strength of s Watts is equivalent to $10 * log_{10}(s/0.001)$ dBm. For example, a signal strength of 1 Watt is equivalent to 30 dBm.

The *NetStumbler* extracts the SS and the SNR information from the AP, each time a broadcast packet is received. More specifically, we took signal strength measurements from all the five APs at 200 different locations (cells) of our floorplan, depicted with the 'X' marks in Fig. 6.1. That is, for each cell, we recorded the signal strength measurement of every AP. The sampling at each cell lasted 1 minute and we collected 1 signal strength value at each second. Thus, we had 60 measured signal strength values from every AP for each of these cells. In some cells, where we encountered numerous disconnections from some APs, we had less than 60 measured signal strength values for these APs.

6.2.2 Data processing

After we collected the signal strength values, we wrote a simple application using Perl [37] to control the entire data collection process from the mobile host. Thus, for each

cell c, we created a tuple of the form (c, AP_i, ss_{set}), where $i \in \{1, 2, 3, 4, 5\}$ corresponding to the five APs, and ss_{set} is the set of signal strength values for the corresponding AP during the data collection. For each of these tuples, we computed the *minimum*, *maximum*, *median*, *mean* and *standard deviation* of the corresponding signal strength values for each of the access points.

To avoid the time-consuming procedure of taking measurements for the whole environment, we decided to build our signal strength map using only 200 positions. In this way, we trade off location accuracy for a dramatic reduction in training time. In order to fill the whole map with the suitable signal strength values we used the method of cubic interpolation which is described further below (6.3.4)

6.3 Estimation phase

In the second phase, i.e., *estimation phase*, we evaluated the accuracy level using only the signal strength map. We took signal strength measurements in 45 positions, (depicted in Fig. 6.1 with the cross '+' marks), different from the ones of the training phase (depicted with 'X' marks). At each of these 45 positions, we collected signal strength measurements from the IEEE 802.11 beacons from each AP and computed the median and the mean signal strength value, and the standard deviation of all the collected signal strength values for this cell for each AP.

To evaluate the impact of the signal strength map we performed the following simulations. We considered the simulation's scenario floorplan that corresponds to the floorplan of Fig. 6.1. Each of the 45 cross marks represents the position of a "virtual host" that runs the CLS system in the simulations. As mentioned in the main description of our system (Section 4.2), each host has access to a copy of the signal strength map computed during the training phase, and the measured signal strength values. We assume that APs act as landmarks and can position themselves on the grid. In order to estimate our position we used two methods and we evaluated the corresponding results.

6.3.1 Estimation using a threshold based on median signal strength

In the first method, we tried to find a potential position using a threshold based on the median signal strength. First, we tested this method without running CLS but using only

the signal strength information and named this algorithm 'no CLS with median threshold'. Then we incorporated the signal strength information in CLS and called this 'extended CLS with median threshold'.

Baseline case: no CLS with median threshold

Firstly, we used only the signal strength map information and we tried to estimate a position of a device. In this case, as we aforementioned, each host maintains a grid that is initialized with zeros at the beginning of the run. Moreover, each of the 45 hosts uses the collected signal strength information from the IEEE802.11 beacon packets of each AP and computes a median signal strength $\overline{s_i}$ for each AP i. The host compares this value against its signal strength map information (for each AP), which was formed in the training phase, and accumulates votes on the cells of its grid as follows:

For each cell c, it compares the computed median signal strength value $\overline{s_i}$ (from AP_i) with the $SM_i[c].min$, $SM_i[c].max$ and $SM_i[c].median$. SM_i is the part of the signal map that corresponds to AP i, and $SM_i[c].max$, $SM_i[c].min$ and $SM_i[c].median$ are the maximum, minimum and median signal strength values of cell c of the map for AP i, respectively. If the condition (6.1) is satisfied, the cell c accumulates a vote from AP_i.

$$\frac{|\overline{s_i} - SM_i[c].median|}{SM_i[c].max - SM_i[c].min} \leq 0.1 \tag{6.1}$$

The final position is found by computing the centroid of all the cells with maximal values. The condition (6.1) means that if a cell's corresponding area in the signal map has a value different from the measured value by less than 10% of the difference of the corresponding maximum and minimum measured values in the signal map, the cell accumulates a vote on behalf of the given AP.

Extended CLS with median threshold

Then, we incorporated the signal strength information in CLS and called this algorithm 'extended CLS with median threshold'. At this time, the grid of each host was initialized at the beginning of the run, but instead of zeros, we set the value 50, which is the weight of an AP (equal to the total number of all hosts in the terrain), in all the cells that had the maximum values from the 'no CLS with median threshold' algorithm for this host. Next, we ran the CLS algorithm as was described in Section 4.3.

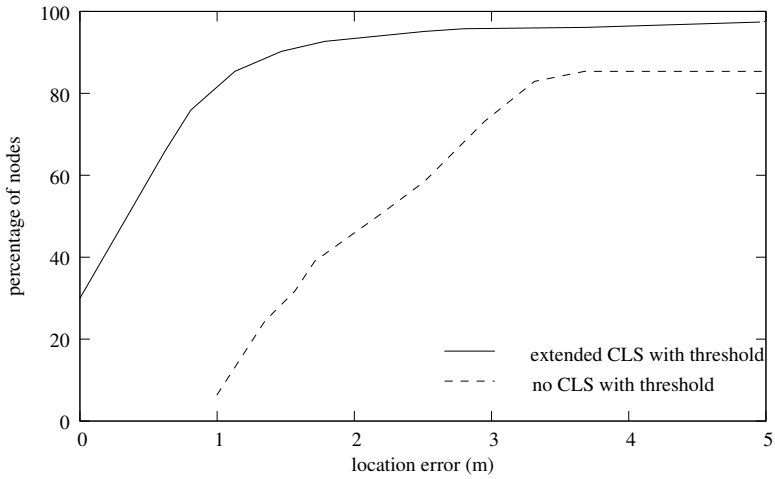


Figure 6.3: Accuracy level using the 'no CLS with median threshold' and 'extended CLS with threshold' algorithms.

Fig. 6.3 shows the accuracy level in the aforementioned setting where there are 5 APs and 45 hosts. More specifically, the 'no CLS with median threshold' line shows the accuracy level when the position is computed based only on the information from the AP signal strength map and the 'extended CLS with median threshold' line shows the accuracy level when the position is computed based on the information from both the AP signal strength map and the distance estimation between the hosts. The landmarks are not included in the percentage of hosts that estimate their location.

More particularly, in the 'no CLS with median threshold' algorithm, 80% of the hosts have at most 3 meters of location error, whereas in the 'extended CLS with median threshold', the same percentage of hosts has at most 1 meter of location error. In addition, in the 'no CLS with median threshold' algorithm, 50% of the hosts have at most 2.2 meters of location error, whereas in the 'extended CLS with median threshold', the same percentage of hosts has at most 0.3 meter of location error. In case of 90% of hosts, in the 'extended CLS with median threshold' algorithm, the location error is at most 1.8 meters, whereas in the 'no CLS with median threshold' algorithm, only 40% of hosts have at most 1.8 meters of location error.

6.3.2 Estimation using confidence intervals of signal strength

In the first method, we tried to find a potential position using confidence intervals based of signal strength. First, we tested this method without running CLS but using only the signal strength information and named this algorithm 'no CLS with confidence intervals'.

Then we incorporated the signal strength information in CLS and called this 'extended CLS with confidence intervals'.

Baseline case: no CLS with confidence intervals

In our second method, we tried to find a potential position with the aid of confidence intervals. Again, each of the 45 hosts uses the collected signal strength information from the IEEE802.11 beacon packets of each AP and computes the mean signal strength $\overline{s_i}$ for each AP i. At this time, the host compares this value against its signal strength map information (for each AP), which was formed in the training phase, and accumulates votes on the cells of its grid as follows: For each cell c, it compares the computed mean signal strength value $\overline{s_i}$ (from AP_i) with the LB_i and UB_i, where LB_i is the low bound and UB_i is the upper bound of the corresponding confidence interval of the signal strength of the cell c. If the Eq. (6.2) is satisfied, that is, if the measured signal strength value $\overline{s_i}$ belongs to the corresponding confidence interval then the cell c accumulates a vote from AP_i. The final position is found by computing the centroid of all the cells with maximum values.

$$LB_i[c] \leq \overline{s_i} \leq UB_i[c] \tag{6.2}$$

Extended CLS with confidence intervals

Then, we incorporated the signal strength information, in our CLS system and we called this algorithm 'extended CLS with confidence intervals'. Similarly with 'extended CLS with median threshold', the grid of each host was initialized at the beginning of the run, but instead of zeros, we set the value 50 (weight of an AP) in all the cells that had the maximal values from the 'no CLS with confidence intervals' algorithm for this host. Next, we ran the CLS algorithm as was described in Section 4.3. Fig. 6.4 shows the accuracy level in the aforementioned setting where there are 5 APs and 45 hosts. Specifically, the 'no CLS with confidence intervals' line shows the accuracy level when the position is computed based only on the information from the AP signal strength map and the 'extended CLS with confidence intervals' line shows the accuracy level when the position is computed based on the information from both the AP signal strength map and the distance estimation between the hosts. Again, the landmarks are not included in the percentage of hosts that estimate their location.

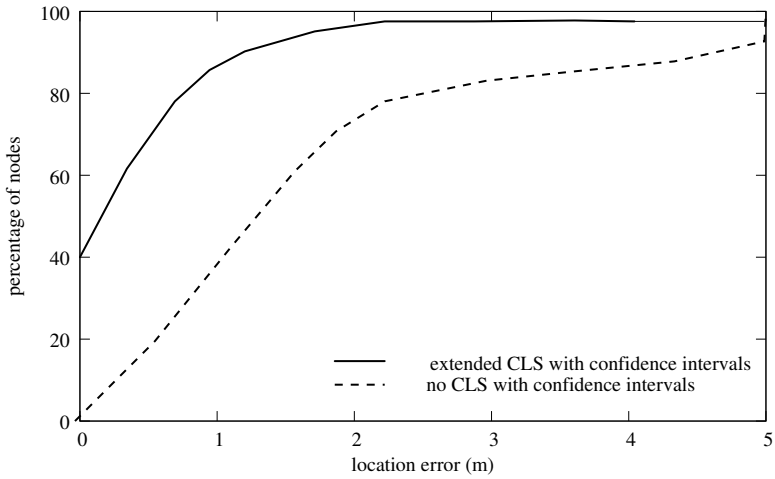


Figure 6.4: Accuracy level using the 'no CLS with confidence intervals' and 'extended CLS with confidence intervals' algorithms.

More particularly, in the 'no CLS with confidence intervals' algorithm, 80% of the hosts have at most 2 meters of location error, whereas in the 'extended CLS with confidence intervals', the same percentage of hosts has location error less than 1 meter. In addition, in the 'no CLS with confidence intervals' algorithm, 50% of the hosts have at most 1.2 meters of location error, whereas in the 'extended CLS with confidence intervals', the same percentage of hosts has at most 0.2 meter of location error. In case of 90% of hosts, in the 'extended CLS with confidence intervals' algorithm, the location error is at most 1.2 meters, whereas in the 'no CLS with confidence intervals' algorithm, only 40% of hosts have at most 1.2 meters of location error.

6.3.3 Comparison of estimation algorithms

Fig. 6.5 illustrates the comparison of the two aforementioned estimation methods. In particular, Fig. 6.5(a) depicts the location error of CLS based only on the signal strength map information, using the 'no CLS with median threshold' and 'no CLS with confidence intervals' algorithms. As we can see, in the 'no CLS with median threshold' algorithm, 80% of the hosts have at most 3 meters of location error, whereas in the 'no CLS with confidence intervals' algorithm, the same percentage of hosts has at most 2 meters of location error. Fig. 6.5(b) depicts the location error of CLS based both on the signal strength information and the distance estimations between the hosts, using the 'extended CLS with median threshold' and 'extended CLS with confidence intervals' algorithms. Specifically, the 'extended CLS with median threshold' algorithm, 80% of the hosts have

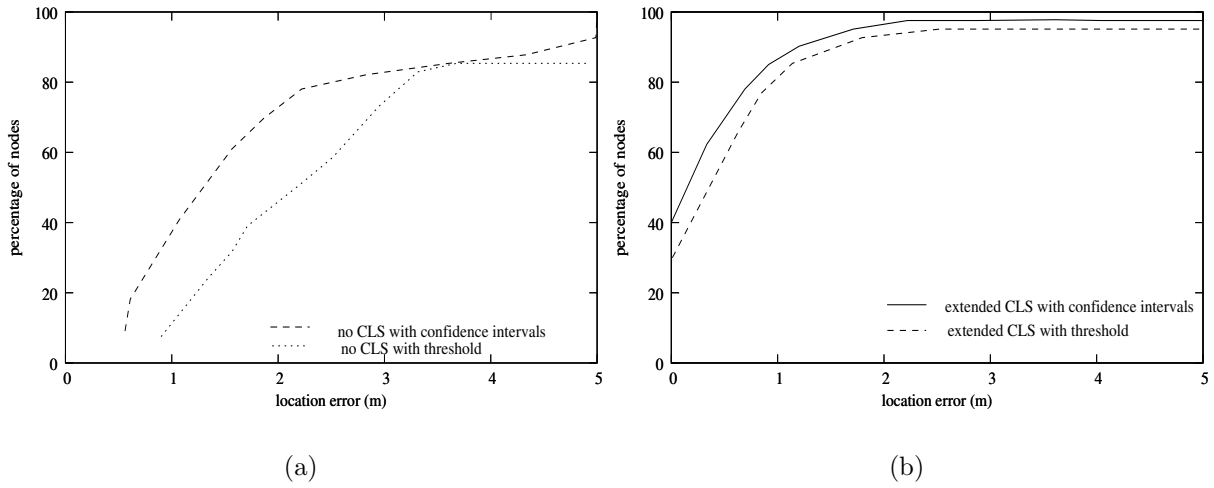


Figure 6.5: Comparison of the accuracy level using (a) the 'no CLS' algorithms and (b) the extended CLS algorithms.

at most 1 meter of location error, whereas in the 'extended CLS with confidence intervals' algorithm, the same percentage of hosts has at most 0.8 meters of location error (less than 1 meter). Therefore, we conclude that using the 'confidence intervals' algorithm, we obtain better accuracy, especially when we use only the signal strength map information. If we incorporate the signal strength map information in CLS, the location error presents similar results, since the sharing of distance information between the hosts enhances the location estimations, by far, regardless of the algorithm (with 'threshold' or 'confidence intervals') that is used.

6.3.4 Methods for building the signal strength map

As it mentioned in Section 6.2, we built our signal strength map using only 200 out of 2500 cells in order to avoid the time-consuming procedure of taking measurements for the whole environment. In this way, we trade off location accuracy for a dramatic reduction in training time. We filled the whole map with the suitable signal strength values using the method of cubic interpolation. We ended up using this method after we have tested several methods for building the signal strength map, since it showed the best results. More particularly, we present briefly these techniques and the results that came up after their comparison.

Least squares method

Least squares is a mathematical optimization technique that attempts to find a "best fit" to a set of data by attempting to minimize the sum of the squares of the ordinate differences between the fitted function and the data [38].

Linear interpolation

Interpolation is a method of constructing new data points from a discrete set of known data points. Linear interpolation takes two data points, say $(x_a, y_a) and (x_b, y_b)$, and the interpolant is given by equation (6.3)

$$f(x) = \frac{x - x_b}{x_a - x_b} y_a - \frac{x - x_a}{x_a - x_b} y_b \tag{6.3}$$

Although, linear interpolation is quick and easy, it is not very precise. Another disadvantage is that the interpolant is not differentiable at the point x_k [38].

Cubic interpolation

Since linear interpolation uses a linear function for each of the intervals $[x_k, x_{k+1}]$, cubic (or spline) interpolation uses low-degree polynomials in each of the intervals, and chooses the polynomial pieces such that they fit smoothly together. The resulting function is called *spline* [38]. Cubic interpolation incurs a smaller error than linear interpolation and the interpolant is smoother.

To compare the above three interpolation methods, we experimented in the Networks and Telecommunications Laboratory (NetLab) in ICS-FORTH, that is depicted in the floorplan in Fig. 6.6 with the number 170. This laboratory is divided in 60 cells and each cell has area equal to 1x1 square meters. We took real measurements in 12 cells of the room, while the signal strength values of the remaining 48 cells were obtained using each one of the aforementioned interpolation techniques. In particular, we individually interpolated the median, maximum and minimum values of the signal strength, that is, our interpolated data were pairs of the form: $(D_i, median(SS_i))$, $(D_i, max(SS_i))$ and $(D_i, min(SS_i))$, respectively. SS_i is the signal strength of the AP i that corresponds to all the cells with distance D_i from AP i. In this room the cells can receive beacons from 4 APs.

Fig. 6.7 depicts three curves, displaying the corresponding location error of each one

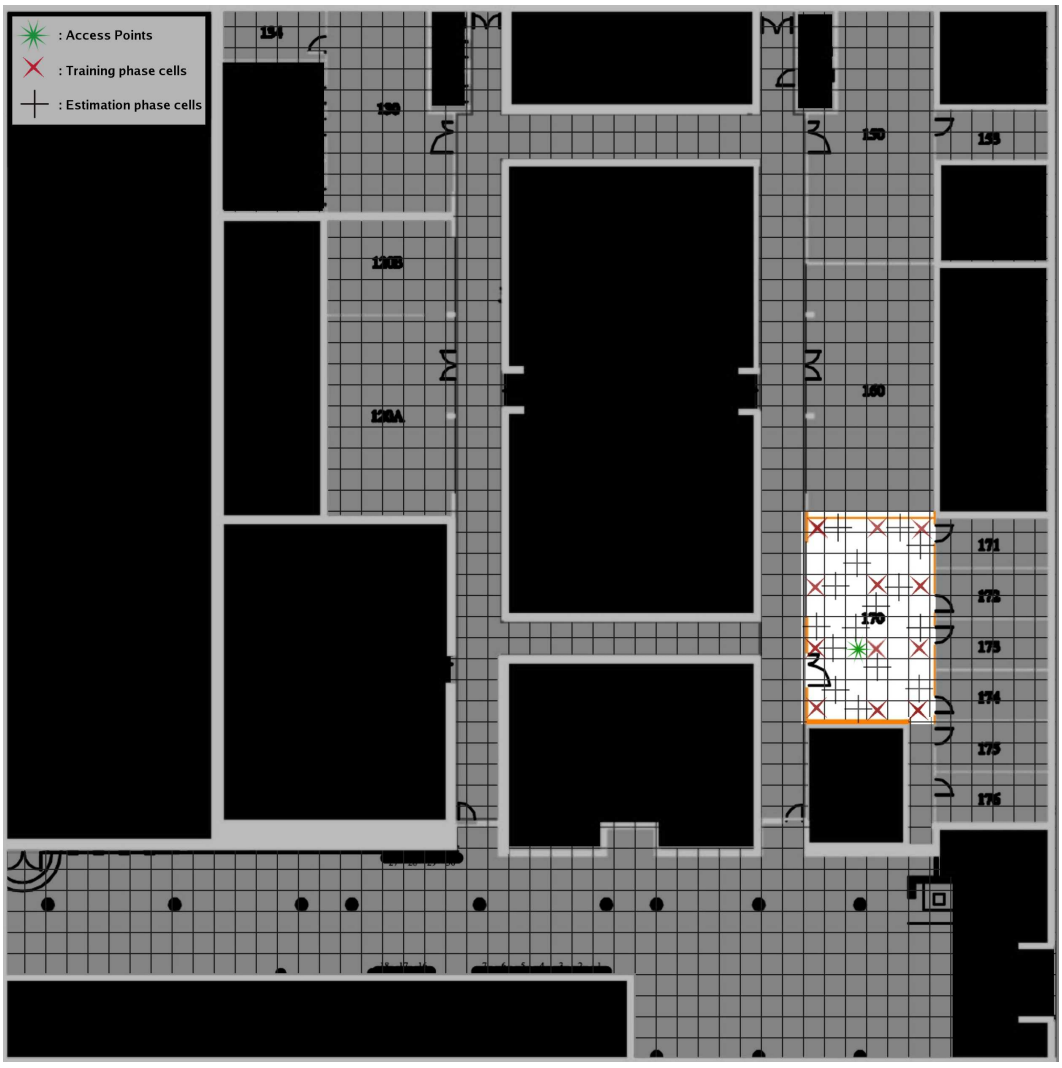


Figure 6.6: The testbed of Networks and Telecommunications Laboratory (NetLab) in ICS-FORTH. It shows the positions at which signal strength was measured for the training and estimation phase ('X' marks and cross '+' marks, respectively).

of the interpolation methods. The location error is estimated by inserting the true signal strength values measured in 15 positions, along with the corresponding interpolated values, in the 'no CLS with median threshold' algorithm. It illustrates that cubic interpolation results to a smaller location error compared to the other two methods.

We also tested the accuracy level of our system using the 'no CLS with median threshold" algorithm, where at this time our interpolated data were triads of the form: (x, y, $median(SS_i)$), (x, y, $max(SS_i)$) and (x, y, $min(SS_i)$). SS_i is the signal strength of AP i that corresponds to the cell with coordinates (x, y). Fig. 6.8 displays the location error for the cubic interpolation technique, based on the distance (D_i) and on the cell's coordinates (x, y). We observe that the latter method results in better accuracy, compared to the

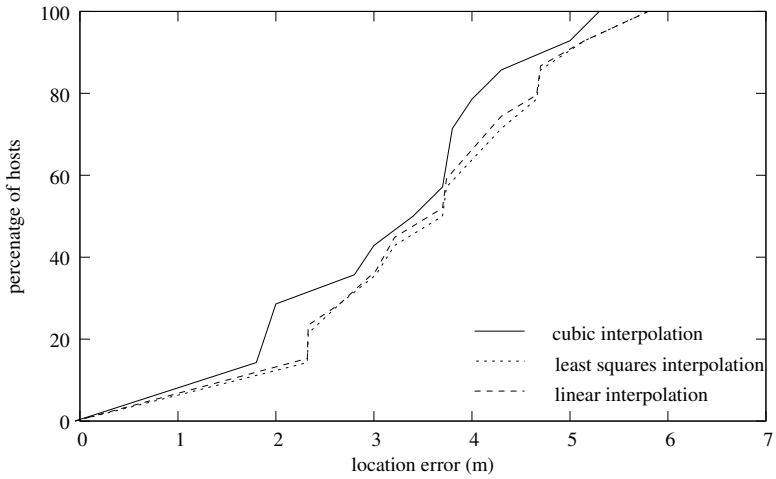


Figure 6.7: Comparison of linear interpolation, cubic interpolation and least squares technique. The interpolated data have the form: (D_i, SS_i).

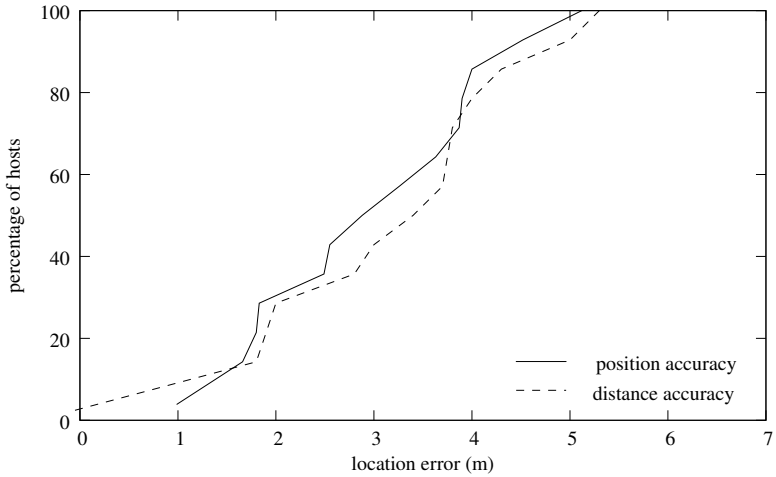


Figure 6.8: Comparison of cubic interpolation using distance accuracy and position accuracy.

former one. This is justified, considering the fact that in the latter approach we associate the signal strength value, measured at a specific position (with distance equal to D_i from the AP_i), with all the spots lying in the circumference, centered at the coordinates of the AP i and with radius $r = D_i$. On the other hand, the former approach associates the signal strength value, measured at a specific position, with a specific cell which has coordinates (x, y).

Cubic interpolation performs better if we train the system with measurements taken uniformly in the testing area, including points on its boundary cells, rather than concentrated in the interior of the area. By definition, in the simple case of interpolating a scalar function $f(x)$ in the interval [a,b], it is necessary to know the values $f(a)$ and $f(b)$. In order to estimate its values outside this interval, we should perform extrapolation methods.

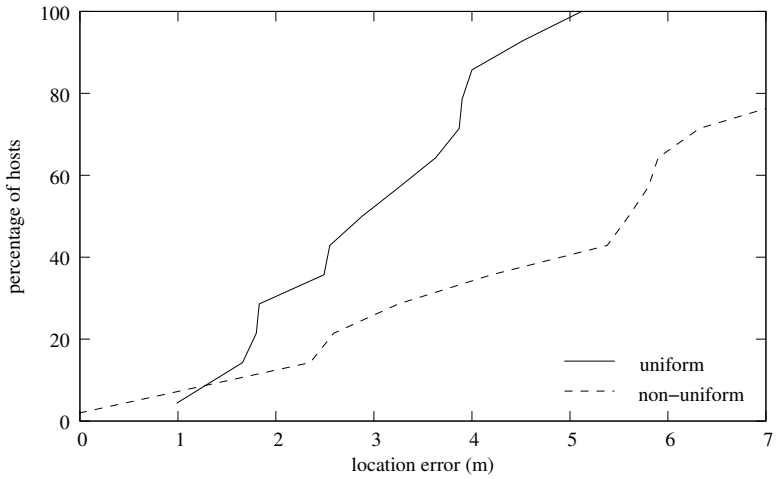


Figure 6.9: Comparison of cubic interpolation using uniform interpolated data and non-uniform interpolated data.

Thus, among other cells, we also took our initial signal strength measurements in cells residing in the border lines of the area. Moreover, we selected these cells uniformly so as to simplify the arithmetic calculations. Fig. 6.9 depicts the difference of the accuracy level when we generated the interpolated data from uniformly spaced data in the boundary cells and from randomly and non uniformly spaced data.

6.4 Impact of connectivity degree

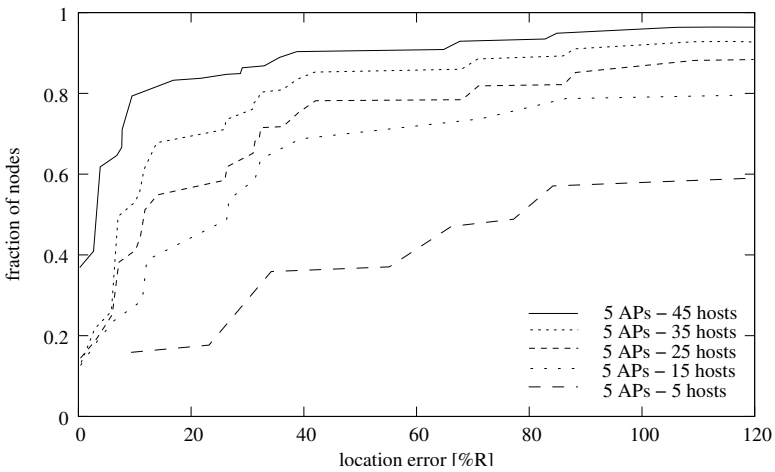


Figure 6.10: Location error for various degrees of connectivity.

We studied the performance of the extended CLS system regarding the average connectivity degree. In order to conduct the simulations we maintain fixed the total number of the landmarks equal to 5 and the range error equal to 10% of the transmission range

and we varied the number of the nodes, that is, 5, 15, 25, 35, and 45 nodes. The average degree of connectivity was 1.8, 9.6, 10.1, 10.23, and 11.18, respectively.

We observed that for low degrees of connectivity, that is, below 10, the location error is bad, whereas for connectivity degree above 10, the location error decreases.

6.5 Evaluation on ns-2 vs. the initial simulation platform

The method of estimating a position using thresholds (6.3.1) was evaluated in the initial CLS implementation [15]. We compared our implementation of CLS system using thresholds with the initial implementation of the system using thresholds, as it was mentioned in Ch. 1, and present our results in the following figures.

First, we compared our 'no CLS' algorithm (Fig. 6.11(a)), including the 'no CLS with median threshold' (Section 6.3), with the initial 'no CLS' implementation (Fig. 6.11(b)) which used the same method with threshold as we did. In both figures the grid size of the 'no CLS' schemes was 50x50. These figures show that our 'no CLS with median threshold' algorithm has worse performance in relation to the initial one, that is, the 80% of the nodes have maximum location error of 3 meters, whereas in the initial implementation, the 80% of the nodes have maximum location error of 2 meters .

Moreover, we compared our 'extended CLS' algorithm (Fig. 6.11(c)), including the 'extended CLS with median threshold' (Section 6.3), with the initial extended CLS implementation which is depicted with the line '10%' in Fig. 6.11(d). Again, the initial extended CLS implementation used the same method with threshold as we did. In these figures the grid size of the 'extended CLS' schemes was 100x100, the range error was equal to 10% of the transmission range and there were 5 APs. From these figures we conclude that both our extended CLS algorithm and the initial extended CLS implementation present similar accuracy level, that is, the 90% of the nodes, in each algorithm, have maximum location error of 1.2 meters.

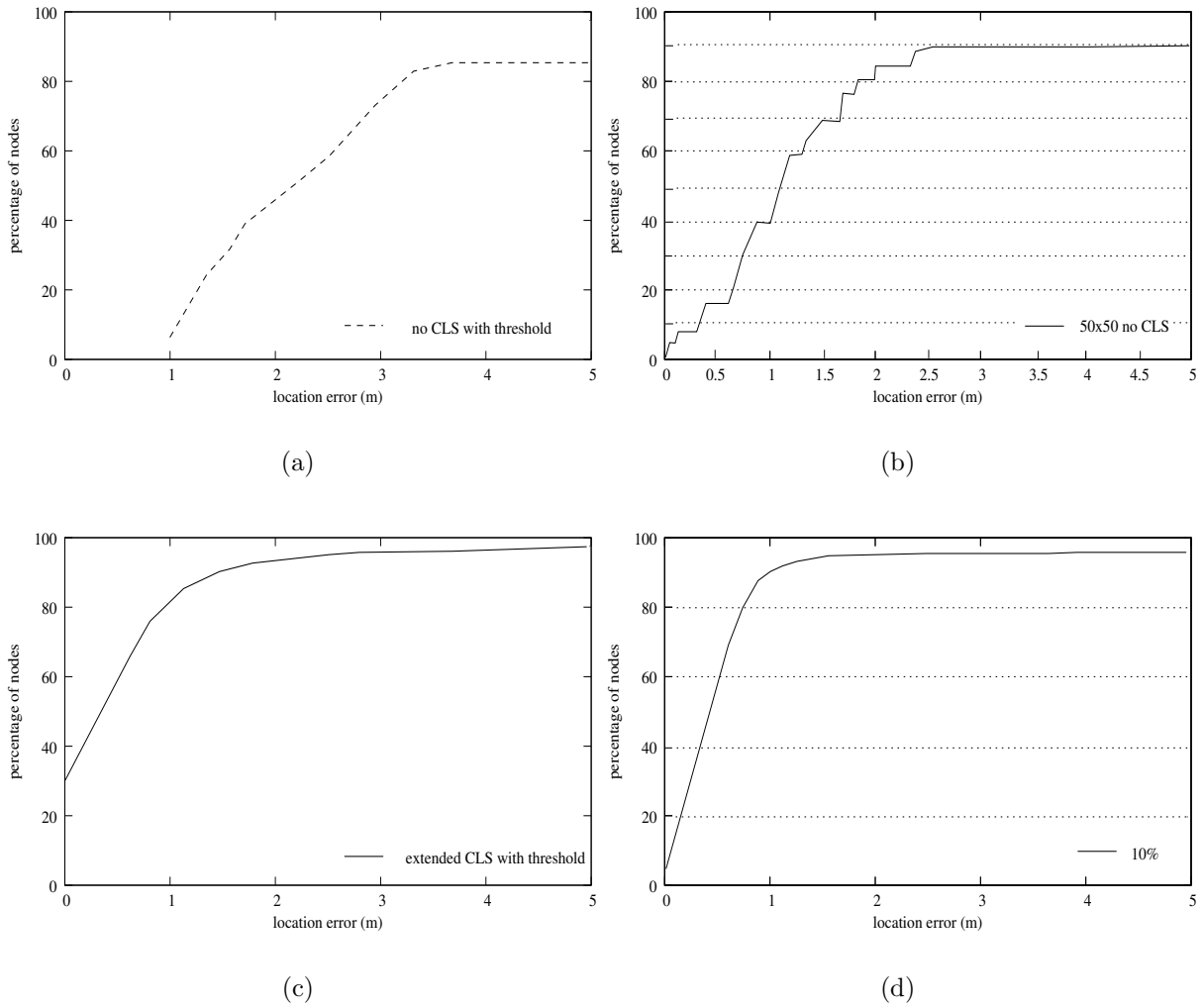


Figure 6.11: (a) Our 'no CLS with median threshold' algorithm vs. (b) the initial '50x50 no CLS' algorithm, and (c) our 'extended CLS with median threshold' algorithm vs. (d) the initial extended CLS algorithm with '10%' range error.

6.6 Summary

In this Chapter, we extended CLS by incorporating additional information (signal strength map) and designed heuristics that use this information to identify the areas in which it is more likely the device to be located. In particular, we used real-life signal strength measurements. We evaluated the extended CLS system using two algorithms. One based on a median signal strength threshold and one based on the confidence intervals of the measured signal strength values. More particularly, using only the first algorithm, 80% of the hosts have at most 3 meters of location error, whereas using also the CLS position estimates, the same percentage of hosts has at most 1 meter of location error.

Using only the second algorithm, 80% of the hosts have at most 2 meters of location error, whereas using also the CLS position estimates, the same percentage of hosts has location error less than 1 meter.

Chapter 7

Performance of CLS under Mobility

The movement of the peers in a wireless network is the most challenging problem in the location-sensing area. The CLS algorithm, that we described analytically in Ch. 4, can be extended to handle mobility. In this Chapter, we examine the impact of the movement of the peers, the impact of the speed of the mobile peers and the impact of the frequency that a mobile node receives position information from CLS, on the accuracy level of CLS system. Depending on its speed, a host can tune the frequency it runs CLS. In addition to the positioning information it collects from other hosts, it can also 'vote' for a region based on its mobility pattern and speed. For example, it can compute an upper bound of the travelled distance and filter out all of the cells in the grid located outside of that range. By observing these results, we propose an extension for the CLS under mobility, aiming to present, at least, similar accuracy results with CLS where the nodes were stationary. In the first part, we examine the influence of the mobility in CLS, using *ns-2* for the simulations' scenarios. In the second part, we propose some extensions for CLS, where, more than half of the total number of nodes are mobile, and we use real signal strength measurements which, in turn, are incorporated in CLS.

7.1 Description of the mobility pattern

We examined the impact of mobility in the CLS performance by running scenarios with mobile nodes in *ns-2*. In all scenarios, described further down, the mobility model that is used is the Random Waypoint Model [39]. In Sections 7.1.1 and 7.1.2 we describe the Random waypoint model and the movement scenario generation for wireless simulations,

respectively.

7.1.1 Random waypoint model

Random waypoint model (RWP) is one of the most widely used mobility models in performance analysis of wireless ad-hoc networks. In the traditional RWP model in $\Re^2$, the path of the node is defined by a sequence of random waypoints, $P1, P2, \ldots,$ placed randomly using a uniform distribution in some convex domain $D \subset \Re$. At time $t = 0$, the node is placed at some point $P_0 \in D$, either randomly using, e.g., uniform distribution or at some fixed starting point. Then the node moves at constant speed u along a line towards the next waypoint P_1. Once the node reaches waypoint P_1, it takes a new heading towards the next waypoint P_2 etc. Each line segment between two waypoints is referred to as a leg and its length is denoted by l. Optionally, the nodes may have a '*pause time*', when they reach each waypoint before continuing on the next leg, where durations are independent and identically distributed random variables.

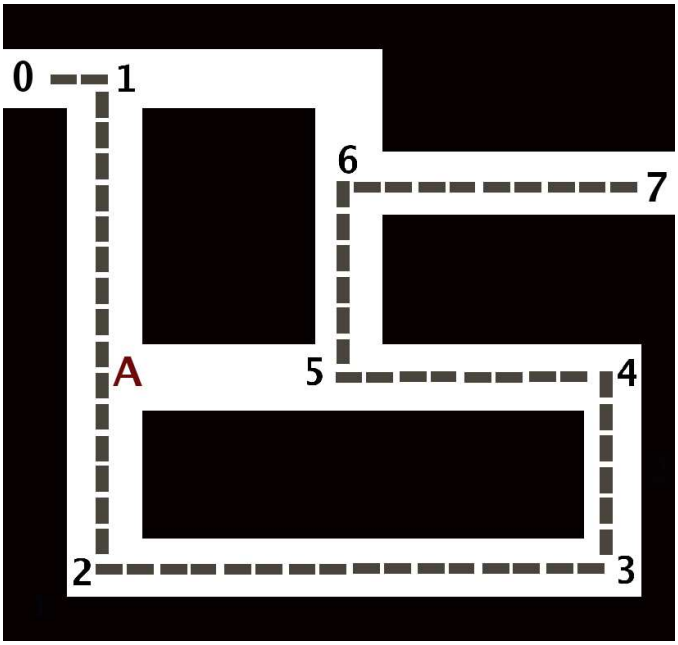


Figure 7.1: Example of a movement path in a floorplan. The mobile node starts from the position '1' and finally stops in position '7'. The intermediate numbers define the successive destinations that it passes by. 'A' depicts a position that it has a pause time.

A mobile node can move in every direction within the border lines of the simulation floorplan. Fig. 7.1 depicts a simple example of a mobile node's movement path in a floorplan. Its total path comprises from successive straight lines. For instance, the mobile

node starts from the position with number '1' and its first destination is the position '2'. Once the node reaches '2', it takes a new heading towards the next position, '3' and so on. Its final destination is the position with the number '7'. During its movement, the mobile node may have a pause time, depicted with the symbol 'A'.

7.1.2 Creating node movements

The mobile node has X, Y, Z(=0) co-ordinates that is continually adjusted as the node moves. The starting position and future destinations for a mobile node may be set in *ns-2* by using the following script code:

```
$node set X_ <x1>
$node set Y_ <y1>
$node set Z_ <z1>

$ns at $time $node setdest <x2> <y2> <speed>
```

At $time sec, the node starts moving from its initial position of $(x1, y1)$ towards a destination $(x2, y2)$ at the defined speed. An example of a movement scenario file is shown below:

```
$node_(0) set X_ 24
$node_(0) set Y_ 11
$node_(0) set Z_ 0
$node_(1) set X_ 21
$node_(1) set Y_ 12
$node_(1) set Z_ 0

$ns_ at 1.0 "$node_(0) setdest 24 7 1.3"
$ns_ at 2.0 "$node_(1) setdest 13 16 1.6"
$ns_ at 10.0 "$node_(0) setdest 18 12 0"
$ns_ at 12.0 "$node_(0) setdest 12 12 1.2"
```

Here two mobile nodes ($node\_(0)$ and $node\_(1)$) have starting positions with coordinates $(24, 11)$ and $(21, 12)$, respectively. For instance, at the first sec, the $node\_(0)$ starts

moving from its initial position towards the destination $(24, 7)$ with speed equal to 1.3 m/s. $Node\_(0)$ has a pause time between the 10th and the 12th sec.

7.2 Impact of different scenarios in the movement of the mobile nodes

First, we examined the movement behavior of the mobile nodes, by running 10 different scenarios for 100 nodes, 10 of which were mobile, 10 were landmarks and the other 80 were stationary nodes. The scenarios are conducted in a square area of 100x100 square units. The speed takes values, randomly, using a uniform distribution, in the interval (0-2] m/s. To achieve more realistic simulations, the mobile nodes pause for a time interval. The pause time is chosen randomly, using a uniform distribution, from the interval [0-2] sec, and can be occurred a random, uniformly distributed, number of times during the movement of the node. Moreover, the range error was 10% of the transmission range R ($R = 20$ meters) and the simulation time was set to 100 sec, meaning that each mobile node receives information about its position every 100 sec. In each scenario, the initial and the destination position of each mobile node was different from the other scenarios and thus the movement paths were different. Specifically, every time that CLS runs, it considers a snapshot of the network and assumes that all hosts are stationary during that run, even though the mobile nodes move during that CLS run.

Fig. 7.2 depicts the mean location error of the 10 different scenarios of each mobile node as simulation time elapses, during a CLS run. Each curve of the figure represents each node's mean location error of the 10 different scenarios. The location error of all mobile nodes follows similar curves, even if their movement paths are different. In particularly, in the time interval 0 - 15 sec, the location error of the nodes decreases. This is explained, since the system approximates the position, taken from the snapshot of the network, faster than the node moves away from that position. After that time, the node keeps on moving but the estimated position which was found remains the same until the end of the simulation time, thus the location error starts to increase. If there is no mobility, the location error remains the same, whereas, if the mobile node moves towards the estimated position, the location error starts to decrease.

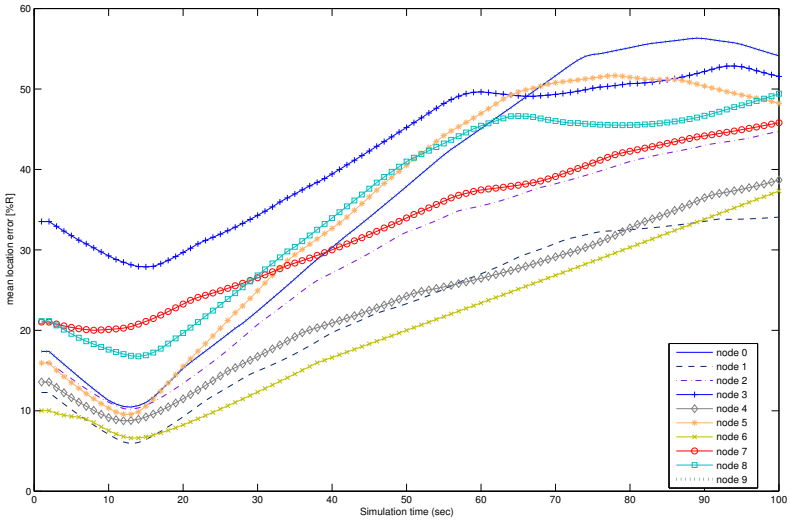


Figure 7.2: Impact of different movement paths of 10 mobile nodes (node0 - node9) on the CLS accuracy. Each curve represents the mean location error of the 10 different scenarios of each mobile node, as simulation time elapses.

7.3 Impact of the speed of the mobile nodes

We examined the impact of the mobile nodes' speed in the location error. The scenario was conducted in a square area of 100x100 square units. The scenario was consisted of 10 landmarks, 80 stationary nodes and 10 mobile nodes which were moving with different speed at each time, taking the values from 1 to 3.5 m/s with a step of 0.5. That is, we ran 6 times the same scenario setting the initial and destination positions of each node and keeping them fixed for each run. The pause time was chosen randomly, using a uniform distribution, from the interval [0-2] sec, and was occurred a random, uniformly distributed, number of times during the movement of the node. The simulation time was set to 100 sec and the range error was 10% of the transmission range R ($R = 20$ meters).

Fig. 7.3 shows the estimation error of four mobile nodes out of ten, in various speeds (1, 1.5, 2, 2.5, 3 and 3.5 m/s) as time passes, during a CLS run. Each of the four subfigures refers to a different mobile node, and illustrates how the location error increases as the node moves with higher speed.

More precisely, the fluctuations of each subfigure is explained as follows. At first, the estimation error of each mobile node starts to decrease as it receives position estimations from CLS. For instance, in Fig. 7.3(a), the location error of *node*0 decreases in the time interval 0-6 sec. After the moment that CLS stops the estimation process (an estimated

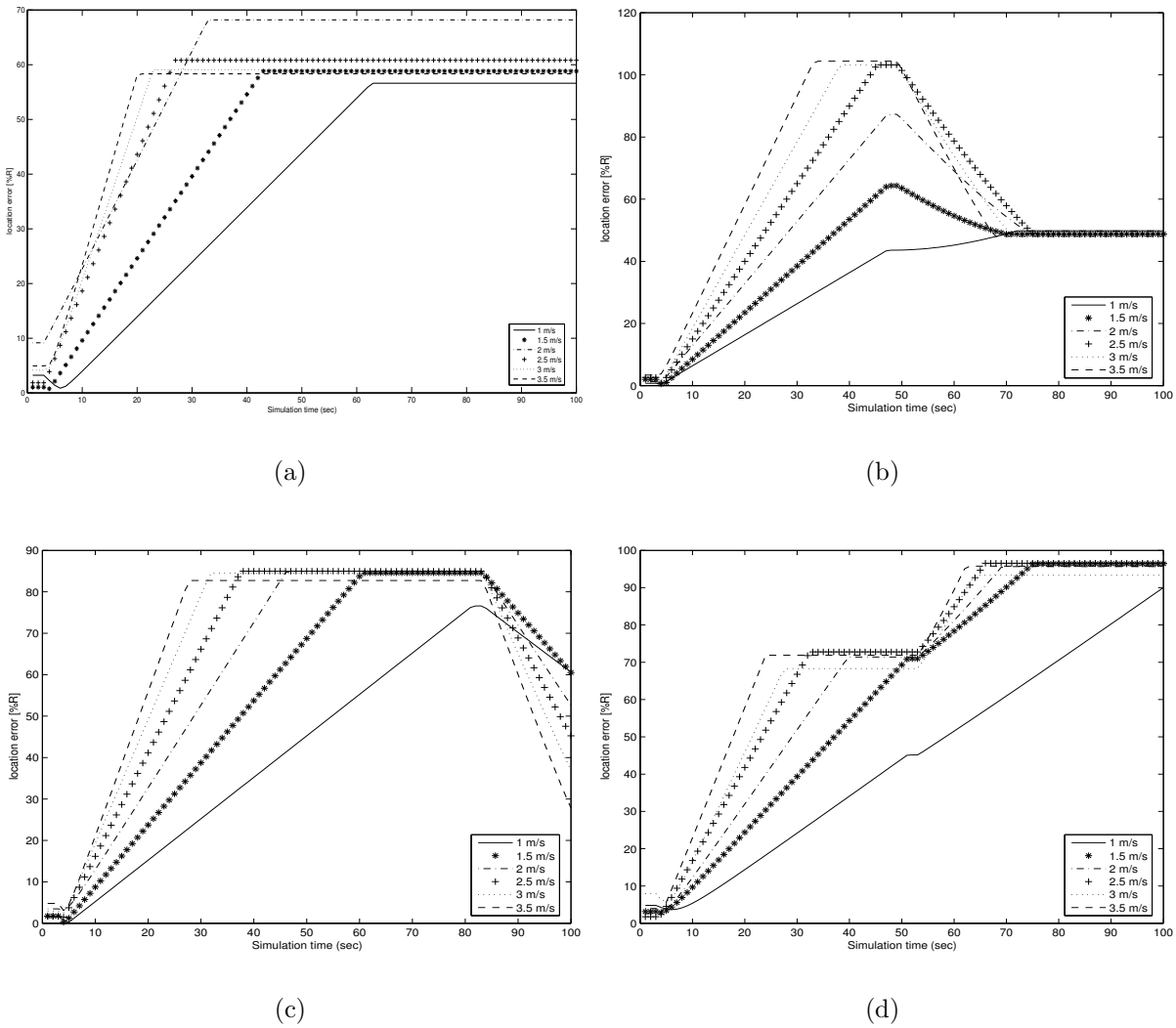


Figure 7.3: Location error of (a) node 0, (b) node 5, (c) node 6 and (d) node 9 in various speeds, as time passes.

position was found), the location error starts to increase rapidly, since the mobile node continues to move. Finally, the location error either remains the same, in case the mobile node does not move anymore, as in Fig. 7.3(a), or starts to decrease, in case the mobile node moves towards the estimated position, as in Fig. 7.3(c). The larger the speed the sharper the fluctuations. For instance, if the mobile node walks with the pedestrian speed of 1 m/s, it does not manage to get far from its estimated position. Therefore, the location error is much less than in any other speed and the fluctuations in its location error are more smooth.

7.4 Impact of the frequency of CLS runs

We examined the impact of the frequency of CLS runs on the location error. In particular, we tested how the frequency that a node receives position information from CLS affects the location error.

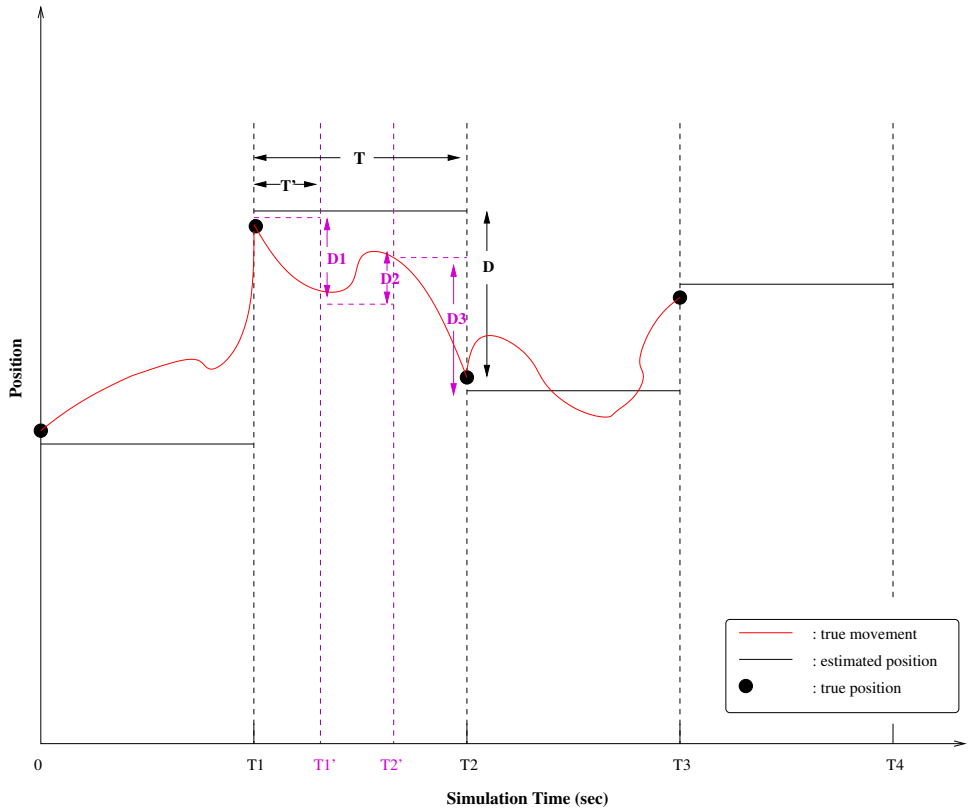


Figure 7.4: Impact of the frequency of CLS runs on the location error.

Fig. 7.4 illustrates how the frequency of the received position information affects the location error of a mobile node. For instance, we assume that a mobile node receives information from CLS system every T sec. Thus, if we consider the time interval $T1 - T2$, the difference between the estimated position and the position that the mobile node actually is, is equal to D. On the other hand, if the mobile node receives information for its position from CLS more regularly, for instance every T' sec, then in the time interval $T1 - T1'$, the difference between the estimated position and the position that the mobile node actually is, will be equal to $D1 < D$. In the time intervals $T1' - T2'$ and $T2' - T2$, these differences would be equal to $D2 < D$ and $D3 < D$, respectively.

The simulation setting was consisted of 10 mobile nodes. We ran 5 times the same scenario, setting the initial and destination positions of each node and keeping them fixed at each run. The speed was set equal to 2m/s and the pause time was set equal to 0 sec.

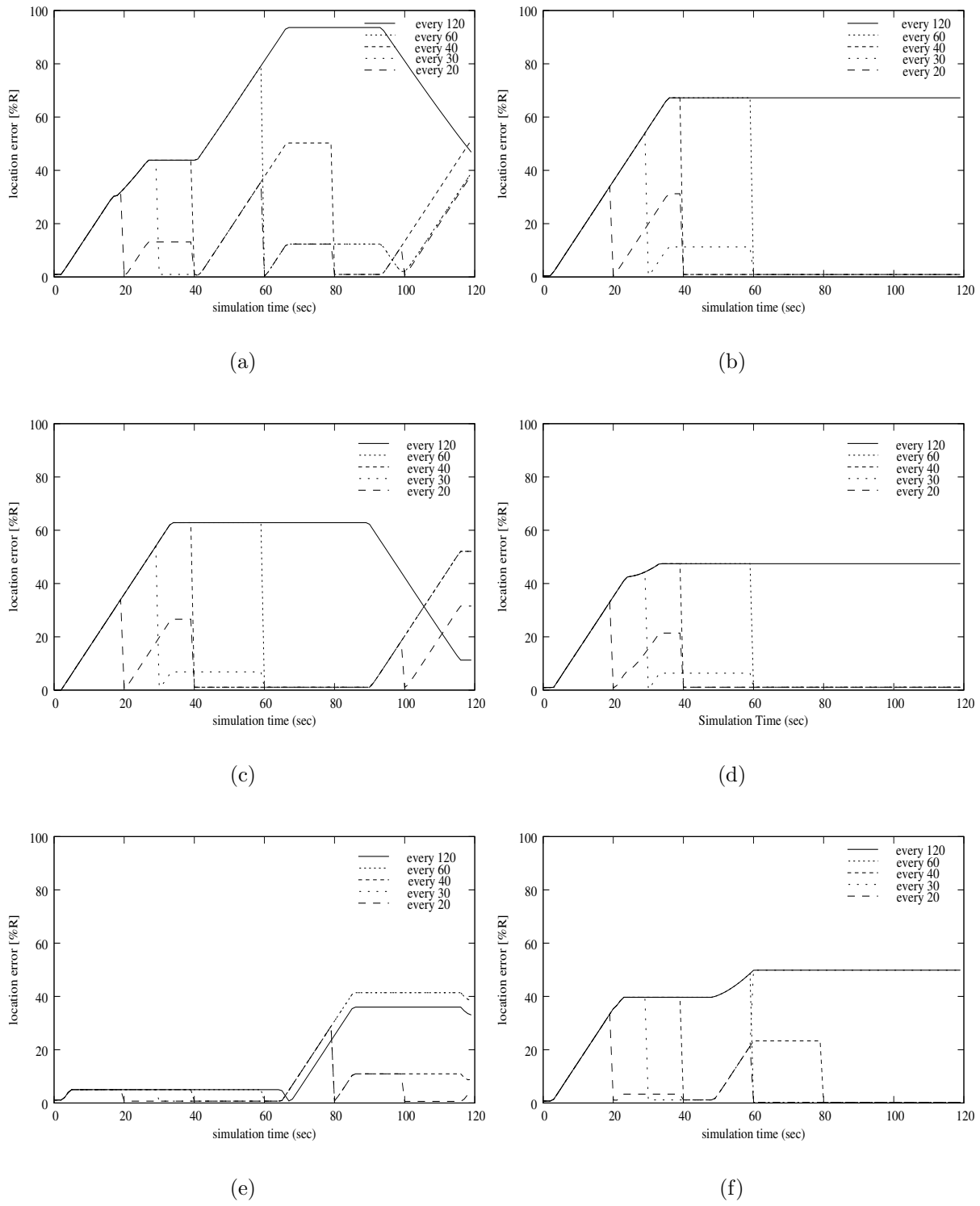


Figure 7.5: Location error of (a) node 0, (b) node 1, (c) node 2, (d) node 3, (e) node 4 and (f) node 5 in various time intervals.

The nodes were moving in a time interval of 120 sec. Particularly, we run the same scenario every 120, 60, 40, 30 and 20 sec. Thus, CLS ran 1, 2, 3, 4 and 6 times, respectively, at

each node. At each time interval, we took a snapshot of the current positions of all the nodes, including the mobile ones, supposing that for the duration of this time interval, the position of the mobile nodes did not changed.

Consequently, we measured the difference between the estimated and the actual position, in various time intervals of the simulation. Fig. 7.5 shows the location error of 6 out of 10 nodes in these time intervals. The name of the curves in the figures, 'every$T$', denotes the difference between the estimated and the true position of a mobile node, if it receives position information from CLS every T sec. The fluctuations of each subfigure are due to the different movement paths of each mobile node. For instance, in Fig. 7.5(b), the mobile node seems to move during the time interval 0 - 37 sec. After that time instance, its location error remains the same.

As it is shown, when the time rate of the position information is more frequent, the location error is less than in any other case. For instance, in Fig. 7.5(b), if the mobile node receives CLS position information every 120 sec, its maximum location error, during its movement, is equal to 65% of the transmission range. However, if it receives CLS position information every 30 sec or every 20 sec, its maximum location error, during its movement, is equal to 55% and 35% of the transmission range, respectively.

7.4.1 Impact of connectivity degree on message exchanges

However, each CLS run adds an overhead, due to the position information exchanges among the hosts, and to the computations of the algorithm. So it is wise to find an acceptable update time, such that there will be little overhead and small difference between the true and the estimated position of a mobile node.

The message exchanges vary regarding the average connectivity degree. Fig. 7.6 depicts the impact of connectivity degree on the CLS message exchanges. In particular, for connectivity degree equal to 2.6, 4, 4.5, 6, and 10, the message exchanges between the nodes are equal to 4.7, 9, 30, 55, and 65, respectively.

7.5 Extended CLS under mobility

As we described in the previous sections, mobile nodes introduce some extra location error, since CLS runs periodically, whereas mobile nodes move continuously. Occupancy,

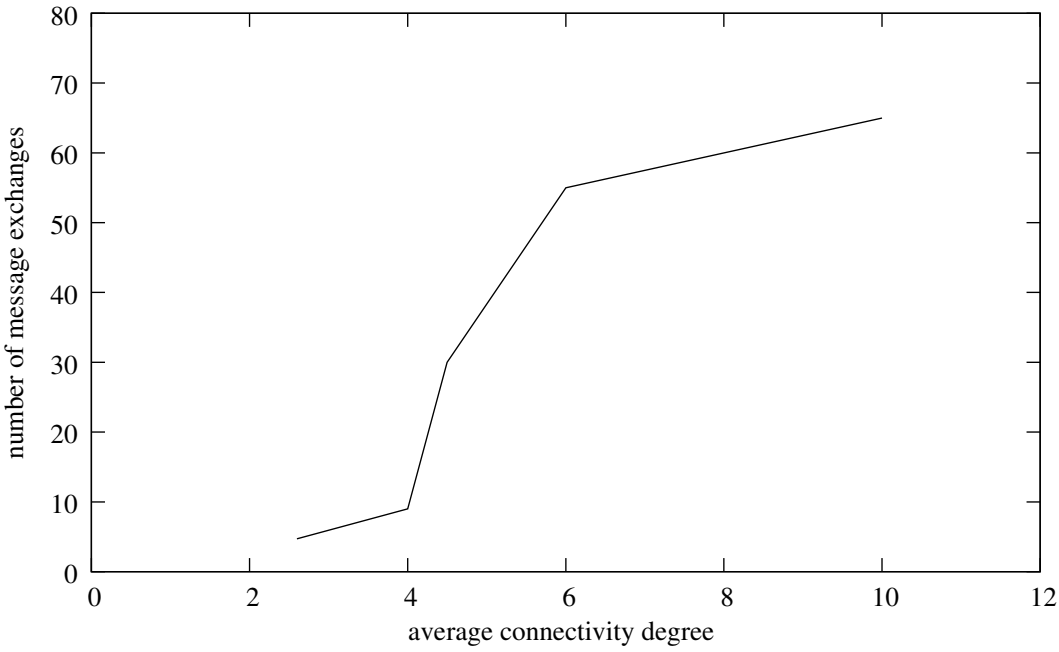


Figure 7.6: Impact of connectivity degree on the CLS message exchanges.

presence, trajectory, or floorplan information can be incorporated in the voting process, that we described in Section 4.3, to filter out the regions in which a device is very unlikely to be located. For example, areas within walls or obstacles are not likely positions for nodes. To enhance the accuracy of CLS, we propose the following extensions: the use of topological information, the use of a signal strength map and the use of the pedestrian speed.

Simulation setting

The simulation setting in the following subsections is the same. In order to test our observations, we used the floorplan of ICS-FORTH (Fig. 7.7). The scenario consists of 50 nodes, 30 of which were mobile, 5 were landmarks and the other 15 were stationary nodes. The mobile nodes had constant speed 1m/s and they followed different movement paths from each other. All of them had a pause time, in some cells, with duration 1 to 10 seconds. Moreover, the range error was 10% of the transmission range R ($R = 20$ meters) and the total simulation time was set to 120 sec. Since the position of mobile nodes changes, as time elapses, the connectivity degree of the nodes also changes. However, the average connectivity degree is equal to 10. Fig. 7.7 depicts, indicatively, the movement track of two mobile nodes. The CLS system runs at every 10 sec and each mobile node receives position estimates from CLS every 10 sec.

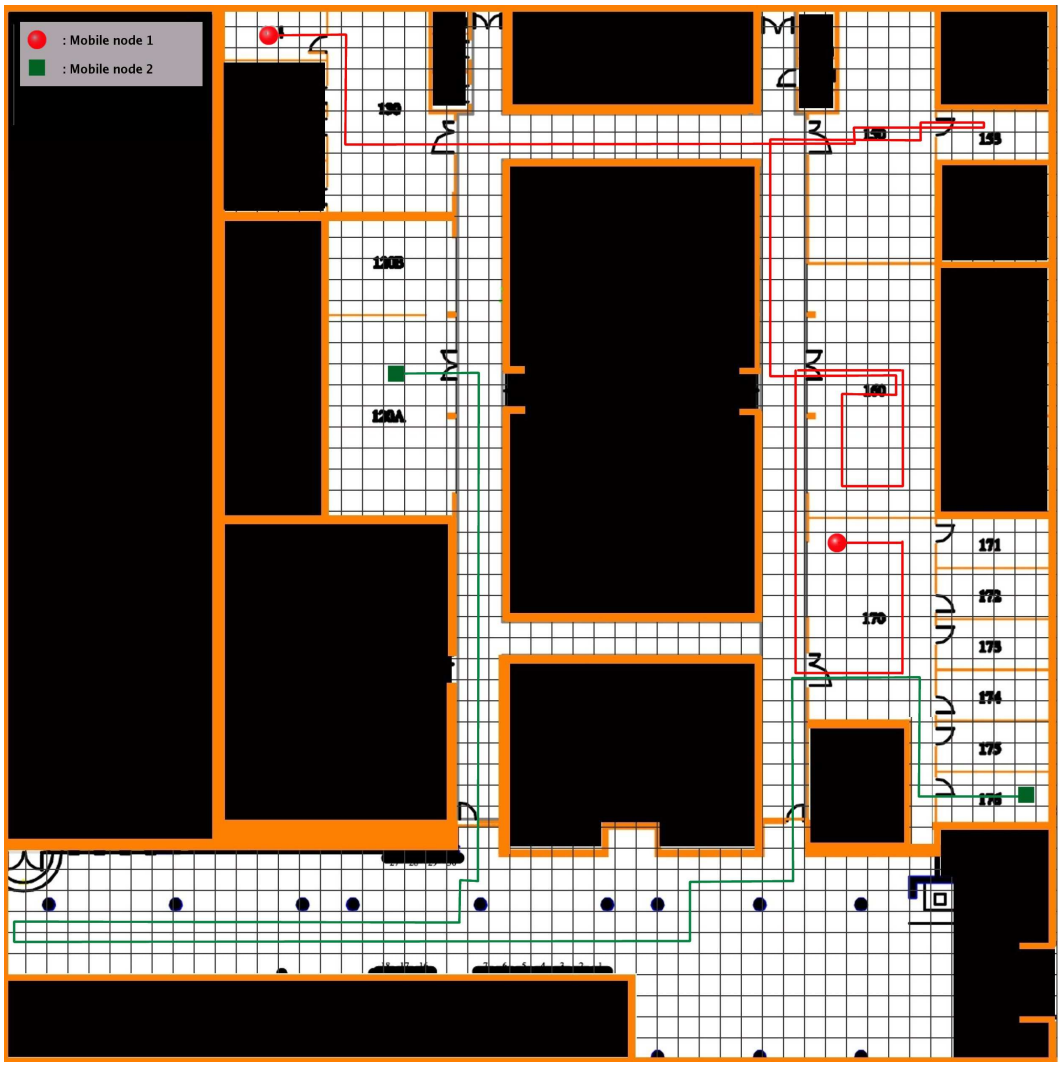


Figure 7.7: Floorplan of the first floor of ICS-FORTH and two movement tracks of two different mobile nodes.

7.5.1 Use of topological information

The first extension in CLS, takes advantage of the topology of the environment. More particularly, a mobile user cannot walk through walls and is unlikely to enter in some forbidden areas. Thus, when the topology of the environment is known, as in our grid-based floorplan, the grid of each node is initialized by assigning negative weights in all the cells that correspond to areas in which the mobile host cannot possibly be. Moreover, apart from these cells, we observed that a mobile node follows some paths in order to go from one place to another. That means that a node does not walk near walls, but prefers to walk in the middle of a room or of a hallway. These cells that a node prefers to walk in, are set with a positive weight in the initialization of each node's grid. This initialization takes place at the beginning of each CLS run.

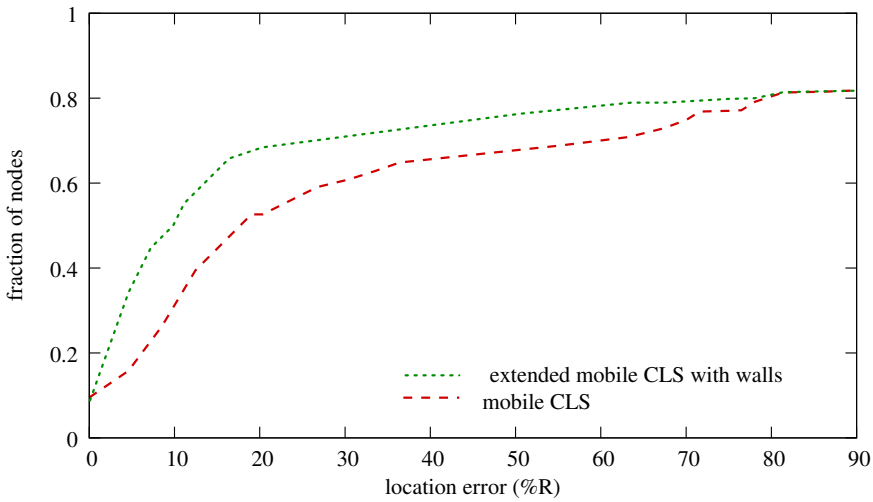


Figure 7.8: Comparison of the accuracy level of CLS with mobile nodes vs. CLS using the environment's topology and vs. CLS using both the signal strength map and the environment's topology.

Fig. 7.8 shows the accuracy level of the nodes when it used the CLS system without any extension, ('mobileCLS') and the accuracy level of the nodes when we used the 'negative' initialization of the cells, where a node is unlikely to be, such as walls, forbidden rooms and columns ('extended mobile CLS with walls'). As it is shown, when we used CLS without any extension, 50% of the nodes had at most 18% location error (%R), while in the extended version, the same % of nodes had at most 10% location error (%R). Additionally, when we used CLS without any extension, 80% of the nodes had at most 90% location error (%R), while in the extended version, the same % of nodes had at most 60% location error (%R).

7.5.2 Use of a signal strength map

Another extension of the CLS, was to consider the infrastructure of the IEEE 802.11 APs and the signal strength map of the environment (as in Section 6). Specifically, we used the signal strength map of the ICS-FORTH floorplan with the confidence intervals of the signal strength (as described in Section 6.3.2). The mobile nodes walk in the floorplan, and every 10 sec they receive position information from the other nodes. However, apart from the CLS position information, at the same time, they also receive the signal strength beacons from each AP of the floorplan. These signal strength values are used to find some extra position information using the estimation method with confidence intervals. Again, the extra position information which corresponds to votes in the grid-based floorplan is

accumulated in the votes of the CLS algorithm.

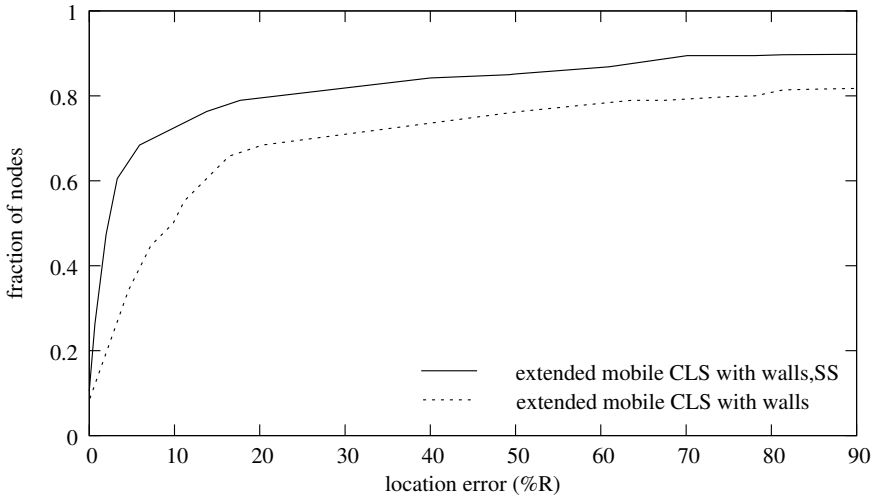


Figure 7.9: Comparison of the accuracy level of CLS with mobile nodes vs. CLS using the environment's topology and vs. CLS using both the signal strength map and the environment's topology.

Fig. 7.9 shows the accuracy level of the nodes, when we used both the extra information of the signal strength map and the information of the environment's topology ('extended mobile CLS with walls & SS'). The scenario settings were kept the same. As it is shown, in 'extended mobile CLS with walls & SS', 50% of the nodes have at most 2% location error ($\%R$) and 80% of the nodes have at most 30% location error ($\%R$).

7.5.3 Use of the pedestrian speed

After observing the pedestrian speed inside buildings, we concluded that a mobile person walks with maximum speed of 1 m/s. Considering a mobility speed of at most 1 m/s, we repeated the scenarios, described in Sections 7.5.1 and 7.5.2, setting all mobile node's speed equal to 1 m/s.

We observed that, if the mobile node receives position information every t sec, the maximum distance that can be found is t meters away from its previous position. More precisely, we consider that at the time instance t_1, the mobile node is located at the point X, and after t sec the node can be located in any point of a disc, centered at X, with radius equal to t meters. We assign to each cell of the grid that resides in the corresponding disc an extra weight.

Fig. 7.10 shows the accuracy level of the mobile nodes, if we also take into account the constrains of the pedestrian speed. As it is shown, there is a small enhancement compared

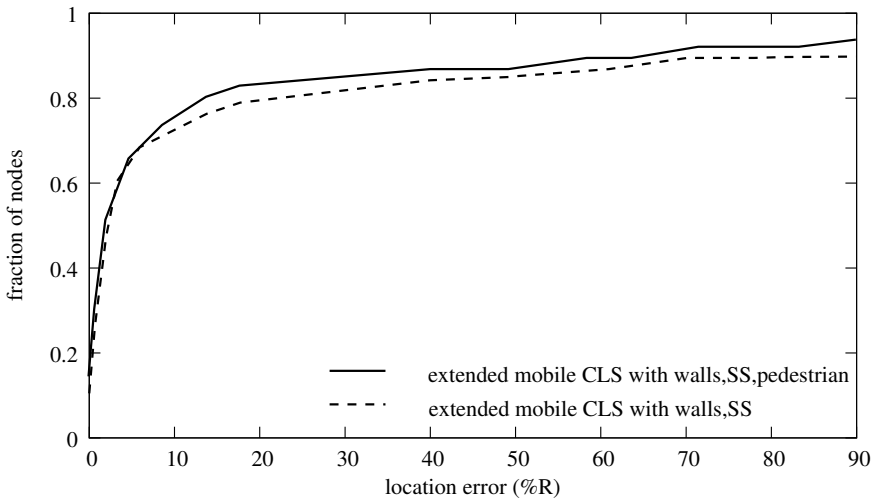


Figure 7.10: Comparison of the accuracy level of CLS using both the signal strength map and the environment's topology vs. CLS using the signal strength map, the environment's topology and the pedestrian speed constrains.

to the previous extensions, since with this way the number of the possible cells, that a mobile can be, is reduced. Specifically, 80% of the nodes have at most 13% location error ($\%R$).

7.6 Sensitivity study

In this Section, we compare the impact of the connectivity degree of the nodes and the frequency of CLS runs on the average location error of CLS.

Fig. 7.11(a) depicts the accuracy level of 30 mobile nodes. The scenario includes also 15 stationary nodes and 5 landmarks. The average connectivity degree of the nodes is equal to 10. On the other hand, Fig. 7.11(b) depicts the accuracy level of CLS in case of 5 mobile nodes, 5 stationary nodes and 5 landmarks, where the average connectivity degree is much smaller. In both cases, the speed is equal to 1 m/s and the pause time ranges from 1-10 sec. The nodes receive position information from CLS every 10 sec and the simulation time lasts for 120 sec. Thus, every node takes 12 estimations from CLS, during its movement. We measured the maximum, minimum, mean and median location error % R for each mobile node during its movement and we present these results in Fig. 7.11(a) and Fig. 7.11(b).

In the former case, we observe that the minimum location error of 80% of the mobile nodes is zero and 95% of them have minimum location error at most 2.5% of R. The

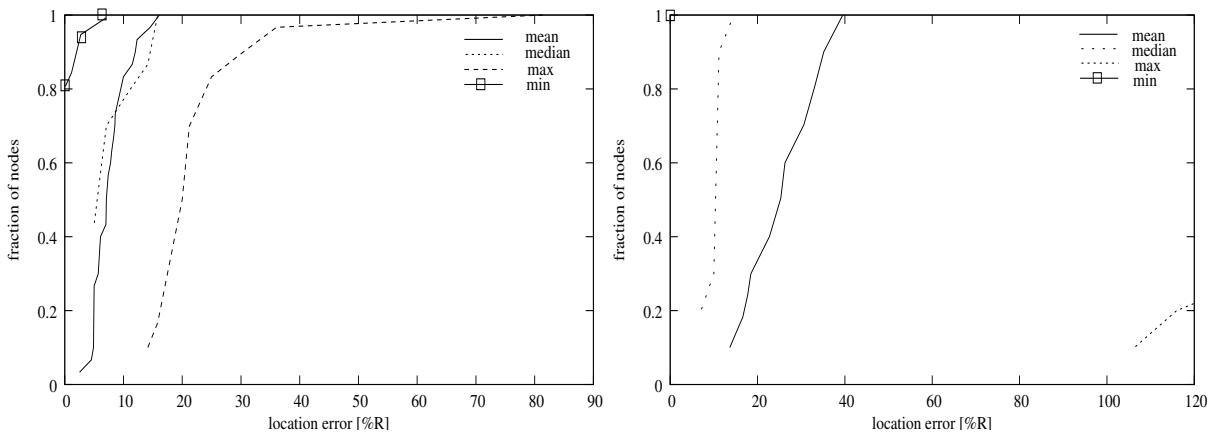


Figure 7.11: Location error %R (max, min, mean and median) of (a) 30 mobile nodes and (b) 5 mobile nodes, moving in different paths and using CLS with the signal strength, topological and pedestrian speed constrains.

maximum location error of 80% of the mobile nodes is at most 24% R. The mean and median location error of 80% of the mobile nodes is at most 9% and 11% of R, respectively. In the latter case, we observe that the maximum location error of the nodes is much greater, more than 100% of R since the average connectivity degree is much smaller. That is, in some cells the nodes cannot 'hear' any other node, therefore they cannot receive estimations from CLS. However, the median location error is similar with the former case.

It is worthy of note that, the aforementioned performance of CLS under mobility, was tested when the frequency of CLS runs was equal to 10 sec. If a mobile node receives position information from CLS more frequent, the CLS performance becomes better. Ideally, if CLS runs continuously, the location error of a mobile node is equal to the location error of a stationary node.

7.7 Summary

In this Chapter, we examined the impact of the movement of the peers, the impact of their speed and the impact of the frequency that a mobile node receives position information from CLS on the accuracy level of CLS system. We proposed extensions for the CLS under mobility to enhance its accuracy, that is, the use of topological information, the use of a signal strength map and the use of the pedestrian speed. We observed that without

any extension, 80% of the nodes had at most 18 meters of location error, whereas using these extensions, the same percentage of the nodes have at most 2.6 meters of location error.

Chapter 8

Experimental Analysis of Range Error

In IEEE 802.11 networks, each wireless node estimates its position with the aid of the signal strength values that receives from other nodes. However, the distance estimations are often inaccurate. For instance, due to multipath interference, line-of-sight obstruction and time-varying phenomena, including the presence or absence of people in a building across the day, it is difficult to obtain a consistent model of the signal attenuation, as a function of distance. In addition, signal strength values differ for different moments during a day. Therefore, we consider a *range error* as a percentage of the transmission range, in order to take more realistic results. In this Chapter, we make an experimental analysis of range error conducting real-life measurements.

8.1 Overview

When we refer to the range error ϵ, we mean an approximation error translated in distance, equal to ϵ meters. Specifically, a measured signal strength value from an AP, even it can be converted with a radio propagation model to a distance d, does not give us the exact position of an object due to the unpredicted phenomena that we aforementioned. Instead of this, a measured signal strength value from an AP can give us an area (set of positions) that an object might be located. This area is formed considering the range error ϵ. Specifically, the possible area of the node is defined by a ring with center the AP and radius $D_{min} = d - \epsilon$ and $D_{max} = d + \epsilon$, as it is depicted in Fig. 8.1.

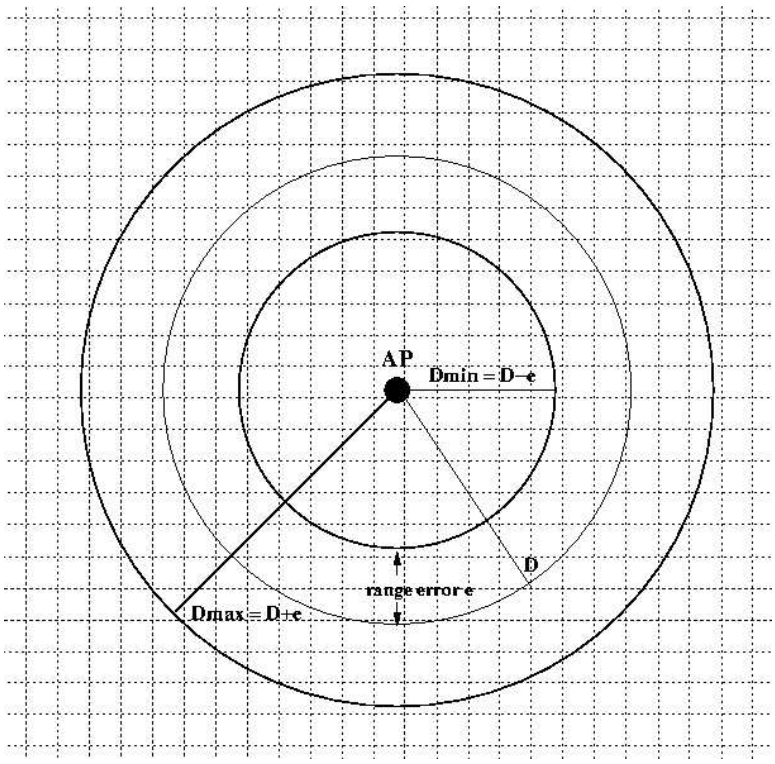


Figure 8.1: Graphical representation of range error.

Moreover, the fact that there is range error in an in-building environment, means that the measured signal strength values cannot give accurate position information by themselves. Thus, we should consider more ways in order to acquire the precise position of an object, as we have shown in Ch. 4.

8.2 Testbed

We managed to find the range error in our environment. Our environment is the Institute of Computer Science (ICS) which resides in the first floor of the Foundation for Research and Technology - HELLAS (FORTH). We divide this environment in 50x50 cells, in order to refer to each position more easily. Each cell has 1x1 square meters. The floorplan of our environment is depicted in Fig. 8.2.

In this floorplan, 5 access points (APs) provide overlapping coverage in the portion of the floor where our experiments were carried out. The 4 APs are equipped with a network interface card from Cisco Aironet, which includes the AP1200 and the fifth AP is equipped with the Cisco AP350 series. In order to find the range error of our environment, we first measured the signal strength in some indicant cells c from every access point i

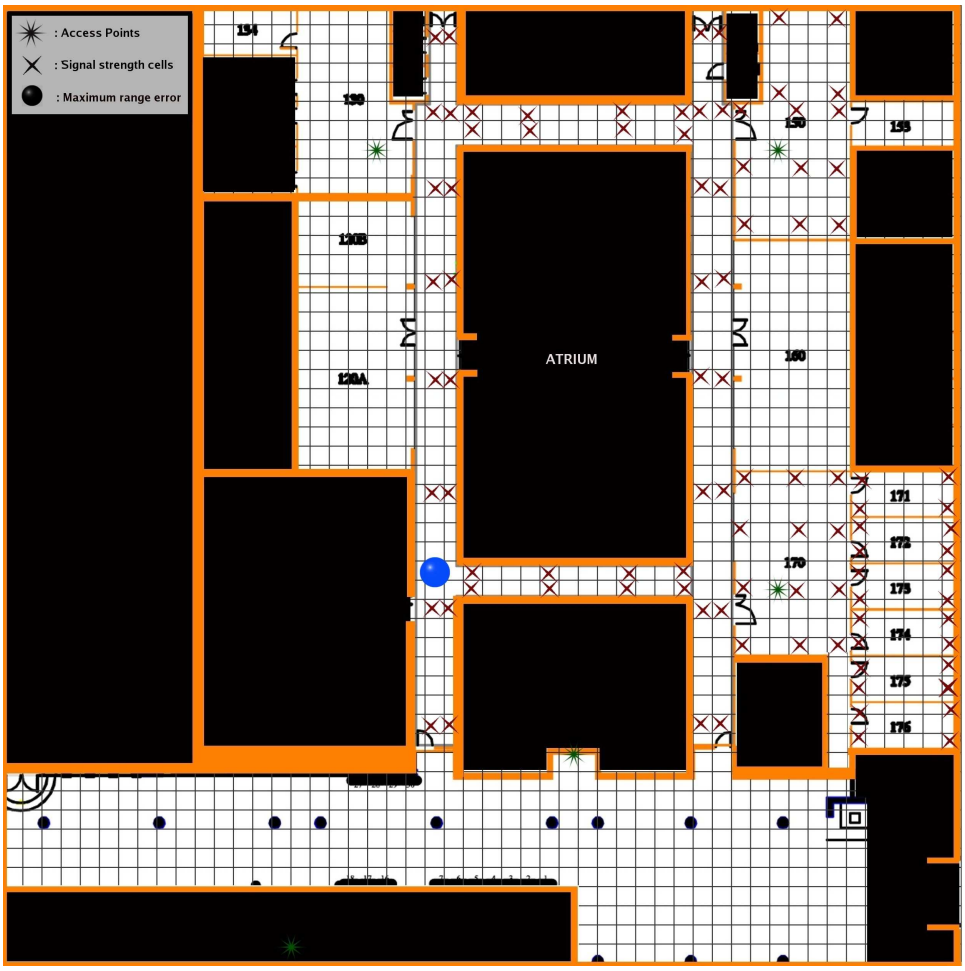


Figure 8.2: Floorplan of the first floor of ICS-FORTH. It shows the five APs, the positions at which the signal strength was measured for the range error calculation and the cells with maximum range error (stars, 'X' marks and solid circle, respectively).

that there is in our floorplan. The topology of the APs and the cells where we acquired the signal strength measurements are illustrated in Fig. 8.2, with a star and with an 'X' mark, respectively.

8.3 Range error calculation

For each cell of the floorplan, we took measurements every second for 1 minute and thus we gathered a set of signal strength values. From these sets, we took the mean μ and the standard deviation σ of the signal strength values and we formed the 95% confidence intervals $CI_i[c]$ for each cell c and the respective APs i.

More particularly, for the independent sample $X_1 + \cdots + X_n$ from a normally distributed population, the mean value μ is equal to $\overline{X} = (X_1 + \cdots + X_n)/n$, and the standard deviation

σ is equal to $\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}$. In order for μ to have 95% confidence interval the following relation should hold:

$$[\overline{X} - a \cdot \frac{\sigma}{\sqrt{N}} < \mu < \overline{X} + a \cdot \frac{\sigma}{\sqrt{N}}] \tag{8.1}$$

where a is equal to 1.96.

In addition, we measured the euclidean distance $d(i, c)$ of each AP i from each cell c. Thus, for every cell c and AP i the $Range\,error_i[c]$ was computed by using the following condition:

$$Range\,error_i[c] = max|d(i, c) - d(i, c')|,\ for\,all\ c'such\,that \tag{8.2}$$

$$CI_i[c] \bigcap CI_i[c'] \neq \oslash \tag{8.3}$$

Cell c' is every cell of our floorplan different from cell c. Eq. (8.3) means that cell c has overlapping signal strength values with cell c' in regard to the same AP i. Thus, when Eq. (8.3) is true, range error of cell c and AP i ($Range\ error_i[c]$) is equal to Eq. (8.2), that is the maximum absolute difference of the euclidean distance $d(i, c)$ of AP i from cell c minus the euclidean distance $d(i, c')$ of AP i from cell c'. The calculation of the range error is represented graphically in Fig. 8.3.
In this figure we suppose that $CI_i[c] \bigcap CI_i[c'_1] \neq \oslash$ and $CI_i[c] \bigcap CI_i[c'_2] \neq \oslash$, where c'_1 and c'_2 are two cells different from cell c, that is, only these two cells verify Eq. 8.3. Since these relations hold, the $Range\,error_i[c]$ is equal to the maximum absolute difference of the euclidean distance $max|d(i, c) - d(i, c')|$. From Fig. 8.3, it is shown that $|d(i, c) - d(i, c'_1)| > |d(i, c) - d(i, c'_2)|$, thus $Range\,error_i[c] = |d(i, c) - d(i, c'_1)|$.

8.4 Results

Fig. 8.4 shows the range error as a percentage of the transmission range of the environment where we conducted the measurements, that is, in the ICS-FORTH. The real transmission range of the APs was different from the one that the APs would have in an open space, since our environment resides inside a building. In order to calculate the true transmission range, we test for each AP, the more distant cell that can 'hear' every AP and we set this distance equal to the transmission range. Thus, in ICS-FORTH the transmis-

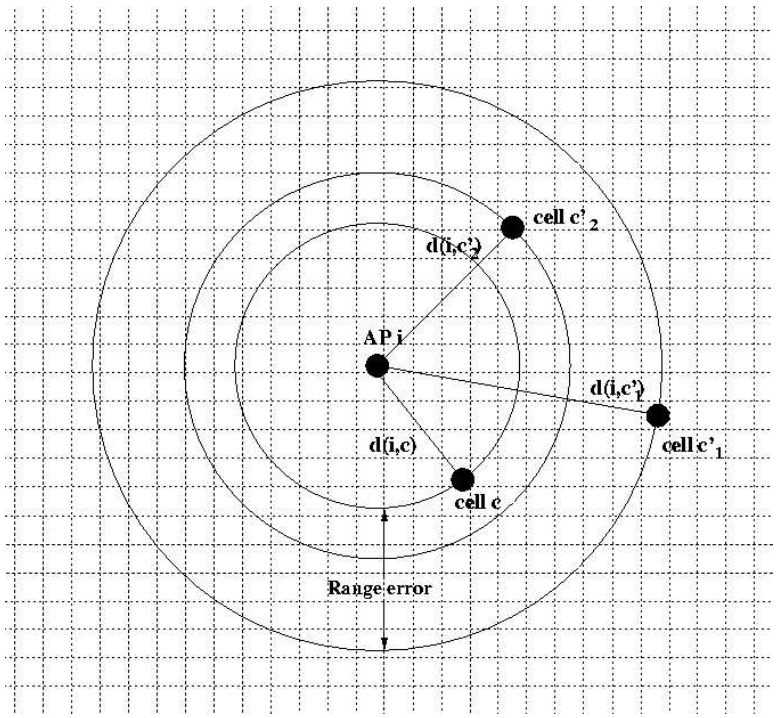


Figure 8.3: Graphical representation of the range error's calculation.

sion range of the access points, considering the environment's topology, is approximately 40 m.

As it is illustrated in Fig. 8.4, 90% of the cells correspond to 4 meters range error, that is 10% of the transmission range or less. The maximum range error is 9.4 meters, that is 23.5% of the transmission range and it is appeared in some cells of our environment.

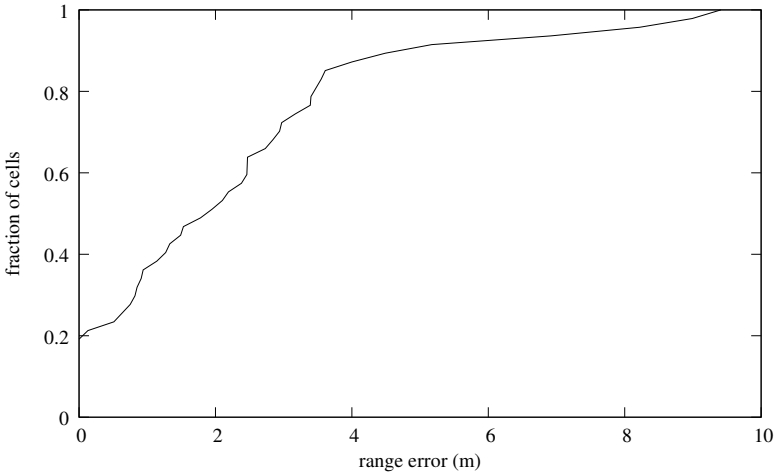


Figure 8.4: Range error in ICS-FORTH in meters.

It is worthy of note that the maximum range error is due to the topology of our environment. For instance, considering only the AP in the Networks and Telecommunications

Laboratory (NetLab) in ICS-FORTH, that is depicted in the floorplan in Fig. 8.2 with the number 170 ('AP NetLab'), the maximum range error appears only in a cell, which is depicted in Fig. 8.2 with the solid circle. As the figure illustrates, this cell has the maximum range error, due to the open space (horizontal hallway) that resides between this cell and the 'AP NetLab'. Specifically, the open space affects the measured signal strength. This means that the cell with the solid circle presents higher signal strength values than equidistant from the 'AP NetLab' cells which are not connected with line-of-sight with this AP. Moreover, the measured signal strength in this cell is similar with the signal strength of cells that reside inside the NetLab. Thus, considering the Eq. (8.2) and Eq. (8.3), c is the cell with the solid circle and c' is the cell inside the Netlab. As we aforementioned, the measured signal strength values in both these cells are similar, that is Eq. (8.3) is verified. Thus, according to Eq. (8.2), the range error is equal to the maximum absolute difference of the euclidean distance $d('APNetLab', c)$ of 'AP NetLab' from cell c minus the euclidean distance $d('APNetLab', c')$ of 'AP NetLab' from cell c'. Nevertheless, since $d(i, c) - d(i, c')$ is high enough due to the topology of the environment, as we mentioned above, the range error in this position is greater. Similar results are presented in respective cells of the other APs.

After we measured the range error of ICS-FORTH, we realized that setting in our system the range error equal to 10% of the transmission range, is a realistic approach. However, in the worst case of setting the range error equal to the maximum value that we found, that is equal to 23.5% of the transmission range, the performance of our system does not deteriorate much, as we have shown in Section 5.

8.5 Summary

In this Chapter, we analyzed experimentally the range error (i.e., distance estimation error) conducting real-life measurements in ICS-FORTH. We observed that 90% of the total area corresponds to a range error of at most 10% of the transmission range. Therefore, the initial choice of setting the range error equal to 10% in all the simulations, that we conducted, was realistic.

Chapter 9

Comparison with Related Work

Significant work has been published in recent years in the area of location-sensing. In this Chapter, we study some of these publications which use radio frequency (RF) signals in order to acquire position information.

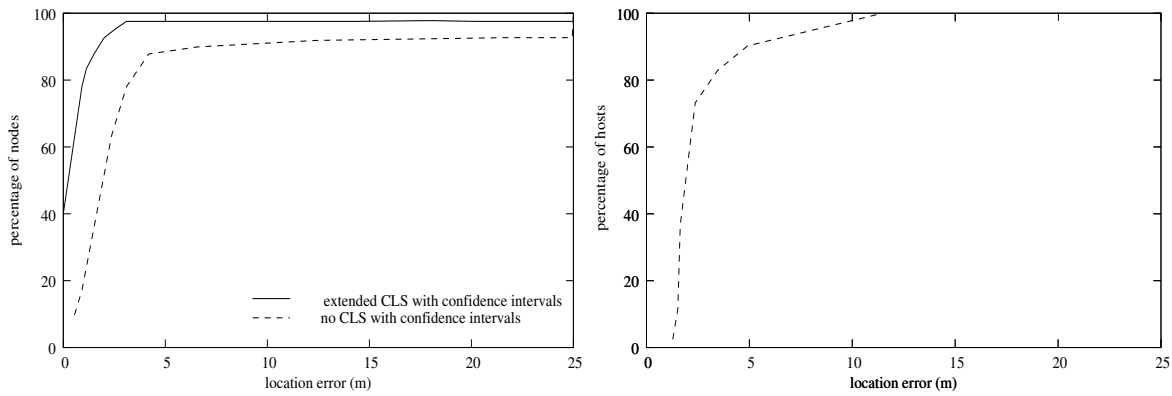
9.1 CLS vs. related work

We compared the performance of CLS with some of the publications that were described in Ch. 3. In particular, we compared CLS with RADAR, Ladd *et al.* system, Cricket, APS and APS with AoA and Savarese *et al.* system.

9.1.1 RADAR

One of these publications, which at the same time comprises a fundamental contribution in the location-sensing community, is the RADAR [5, 14]. RADAR uses maps of signal strength similar to the ones we exploit in Section 6. The signal strength measurements employed a signal strength map with approximately one scan every square meter. More particularly, RADAR uses only a wireless networking signal, employing nearest neighbor heuristics and other pattern recognition techniques for localization. Bahl *et al.*, report that 90% of the hosts can be located with at most 3 meters of error with a sampling density of one sample every 19.1 square meters of their testbed areas using the signal from 5 APs. The error distance for tracking a mobile user is 19% worse than that for a stationary one, that is 3.5 meters. Respectively, if they use 3 APs, 90% of the time their hosts can be located with at most 6 meters of error with a sampling density of one sample every 13.9

square meters of their testbed areas (Fig. 9.1(b)).



(a) (b)

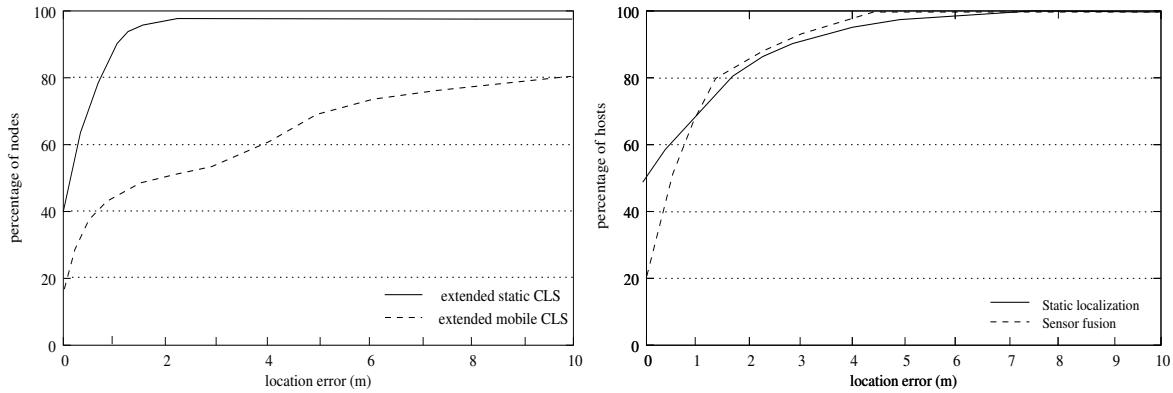
Figure 9.1: CDF of the error distance of (a) CLS and (b) RADAR.

Although our work can take advantage of signal map information, as demonstrated in Section 6, the fundamental algorithm is significantly different and emphasizes on cooperation between the hosts, rather than individual host effort. Even though, if we use only the signal map information with the aid of the confidence intervals, the location error of 90% of the hosts has maximum value of 2.4 meters, using a sampling density of one sample every 12.5 square meters (200 points in an area of 2500 square meters) and using 5 APs. Nevertheless, though we have 5 APs in our floorplan, in 85% of the cells only 4 APs are 'heard' at the same time. Moreover, in our extended implementation, where apart of the signal strength map we use the distance information from the cooperation of the hosts, we can predict the location of a host 90% of the time with at most 1.2 meters of error (Fig. 9.1(a)). The error distance for tracking a mobile user is at most 2.6 meters.

9.1.2 Ladd *et al.*

Another two publications of the Rice University [6, 34] use an infrastructure of IEEE 802.11 access points. Based on the signal strength data gathered by wireless cards at various predefined points of an indoor environment, they build a signal strength map of the environment. They locate a wireless device by using this map and applying a probabilistic inference of the position. Their first system is based on a two step process. In the first step, a host uses a probabilistic model to compute the conditional probability of

its location, in a number of different locations based on the received signal strength from 9 APs. The second step of their algorithm exploits the limited maximum speed of mobile users to refine the results of the first step and reject solutions with significant change in the location of the mobile host. We compare the results of their algorithm with the our CLS implementation (Ch. 4).



(a) (b)

Figure 9.2: CDF of the error distance of (a) the extended static and extended mobile CLS using 5 APs and (b) the static and mobile localization of first publication of Ladd et al. using 9 APs.

The map, where they conducted the tests, consists of four hallways of 139 square meters. They take samples for their experiments in two different orientations at every 1.52 meters. The results for static localization in a hallway show that 77% of the time, their hosts can be located with at most 1.5 meters of error using 9 APs (Fig. 9.2(b)), whereas in our extended implementation, using at most 5 APs, we can predict the location of a host, 77% of the time with at most 0.6 meter of error. Moreover, their training phase is quite extensive and time-consuming since it takes place at each point of their map, whereas in our system the training data is collected in 200 square meters of a total area of 2500 square meters, populating the values of the other cells using cubic interpolation of the actual samples. In this way, we tradeoff location accuracy for a dramatic reduction of time. Their results for mobile localization in a hallway, show that 64% of the hosts can be located with at most 1 meter of error using 9 APs, whereas in our mobile implementation, we can predict the location of a host 45% of the time with at most 1 meter of error, using at most only 5 APs and with less training.

Their second publication uses a topological map of the environment instead of a grid-based map, where the building is divided into cells. The size of these cells is approximately 13.23 square meters. In the training phase, they collect scans in all the cells lasting 28 man-hours, overall. Each cell has signal intensity readings from 14.86 to 33 base stations. In the estimation phase, they choose 5 scans at random for each of the 510 cells and remove them from the training set. The remaining scans are used to train their localizer. Then, for each cell, they use the scans they had removed from the training set as input to the localizer, and attempt to locate themselves. They performed this experiment 100 times, removing different scans each time. In all cases, they determined the correct cell in at least 70% of trials.

In order to compare our work with their second publication, we divided our map in similar size cells of 12 square meters and we tried to localize ourselves. A main difference between our estimation phase and the one above is that for our estimation phase we use new scans. Particularly, we collected these scans a different day from the one we have collected our data for the training phase. That means that we have signal variations due to time-varying phenomena, even for the same cell. However, we succeed correct occupancy of at least 70% of trials, taking readings from only 2 to 5 base stations. In all the other cases the estimation was in an adjacent cell.

9.1.3 Cricket

Cricket [7] is another location sensing system that works using quite a different methodology and primarily targets a different kind of applications than our system. Cricket uses specialized ultrasound and radio frequency hardware to create the infrastructure and embeds receivers in the object being located. Our work is significantly different from theirs, in that Cricket requires investing in extensive infrastructure of specialized hardware, used only for this purpose. On the other hand, our system can operate with limited or no infrastructure. Furthermore, the infrastructure of 802.11 APs are already widely used and our system does not require additional localization hardware.

Nevertheless, Cricket provides a location sensing granularity of 4x4 square feet (1.2x1.2 square meters). Though our system can only provide worse location-sensing accuracy, it is very easy and inexpensive to deploy and maintain, contrary to the deployment of the Cricket sensors.

9.1.4 APS and APS with AoA

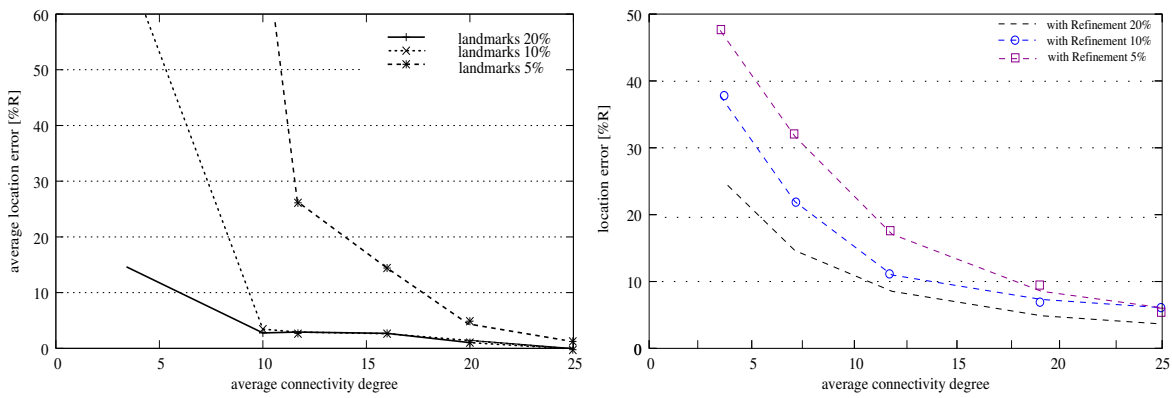
Niculescu's first work [10] shares the same cooperative spirit among hosts with our work. They introduce an algorithm on location-sensing that works on simple geometric principles of Euclidian geometry, concerning triangles and quadrilaterals. Like in our algorithm, the information of the landmark locations is slowly propagated towards the nodes that are further away, while at the same time closer nodes enrich this information by determining their own location. Three variations of this algorithm are presented: 'DV-hop', 'DV-distance' and 'Euclidian'. Though our system is closer to 'DV-distance' in that it uses signal strength information to estimate the distance between hosts, its performance more closely resembles the performance of the 'Euclidian' scheme, but with always 100% solved hosts. More specifically, we outperform all variations of Niculescu's algorithm for 20% or more landmarks, average degree of connectivity 7.5 and distance approximation error of 5%. However, Niculescu's 'DV-hop' and 'DV-distance' variations outperform our algorithm when the number of landmarks becomes 5% and the 'DV-distance' produces equally good results at such low percentage of landmarks. For 10% landmarks, our results are almost equally matched. For example, using 5% landmarks Niculescu's algorithms have an error of approximately 25%, 45% and 55% of the transmission range (for the 'DV-hop', 'DV-distance' and 'Euclidian' algorithms) compared to our error of 85%. However, when the number of landmarks increases to 20%, our error falls to 1.8%, while Niculescu's algorithms perform at approximately 12%, 25% and 12%, respectively.

In their following work [29], they designed and evaluated a cooperative location-sensing system that uses primarily angle estimations and assumes specialized hardware that allows a host to calculate the angle between two hosts. Like in our system, hosts gather data, compute their solutions, and propagate them throughout the network. It is not easy to compare their results with ours due to the different metric used, namely distances vs. angles.

9.1.5 Savarese *et al.*

As it was mentioned in Section 3, the work done by Savarese *et al.* [9] is the closest to our work. They describe a distributed algorithm that determines the position of nodes in an ad-hoc network in two phases, namely the startup ('Hop-Terrain') and refinement

('with Refinement') phase, similarly with our communication and voting phase. Though this work presents many similarities to ours, there are also several very distinct differences. First, we use a grid to solve the location problem instead of a numerical method. Second, our communication overhead is smaller, since we avoid the flooding done in the first phase of Savarese's system by relaying landmark position through the regular exchange of information regarding host information (i.e. the corresponding messages that Savarese's system exchanges during the second phase). Our algorithm can tolerate initial distance measurement errors significantly better than Savarese's and performs better in networks with 10% or more landmarks and a connectivity degree of at least 8. However, Savarese's method can cope better with networks of very small connectivity (such as 4) or fewer landmarks (less than 5%), as it is depicted in Fig. 9.3(a) and Fig. 9.3(b).



(a) (b)

Figure 9.3: (a) CLS 'landmarks' vs. (b) 'with Refinement' phase for various connectivity degrees and landmark %. The range error is 5%.

9.2 Summary

In this Chapter, we compared the performance of CLS with some of the publications that were described in Ch. 3. In particular, in RADAR 90% of the hosts can be located with at most 3 meters of error using 5 APs. The error distance for tracking a mobile user is 19% worse than that for a stationary one, that is 3.5 meters. In CLS, the same percentage of hosts can be located with at most 2.4 meters of location error. In addition, in the extended CLS, 90% of the hosts can be located with at most 3 meters. The error

distance for tracking a mobile user is at most 2.6 meters.

Ladd *et al.* show that, for static localization, 77% of the time, their hosts can be located with at most 1.5 meters of error using 9 APs, whereas in our extended implementation, using at most 5 APs, we can predict the location of a host, 77% of the time with at most 0.6 meter of error. Their results for mobile localization, show that 64% of the hosts can be located with at most 1 meter of error using 9 APs, whereas in our mobile implementation, we can predict the location of a host 45% of the time with at most 1 meter of error, using at most only 5 APs, less training and with frequency of CLS runs every 10 sec.

Savarese *et al.* show that their method can cope better with networks of very small connectivity (such as 4) or fewer landmarks (less than 5%), whereas CLS performs better in networks with 10% or more landmarks and a connectivity degree of at least 8.

It is not easy to compare CLS with the other systems due to the different metric used.

Chapter 10

Conclusion and Future Work

We evaluated a robust location-sensing system, called *Cooperative Location-sensing System* (*CLS*), with extensive simulations using *ns-2*. In particular, we analyzed the impact of range error, density of landmarks, density of wireless devices and mobility on the accuracy of the position estimations. We enhanced the accuracy of CLS, by incorporating additional information, such as a signal strength map, information about the environment (e.g., floorplan) and mobility information.

We observed that when the range error is doubled, the location error is also doubled. Secondly, the location estimations are affected from the density of landmarks more than from the density of hosts. In addition, using only the position estimations from the peers gives better accuracy than using only the signal strength measurements. However, the combination of these two techniques performs even better. Considering the impact of mobility, we observed that the pedestrian speed and the frequency of CLS runs can affect greatly the location estimations.

In particular, using only the CLS position information, we found that 80% of the hosts can position themselves with a location error of at most 1.5 meters, when all hosts are stationary, and 50% of all hosts can position themselves with a location error of at most 3.6 meters, when 60% of the them are mobile, with 10% of hosts to be landmarks, average connectivity degree equal to 10 and range error equal to 10% of the transmission range. Furthermore, considering the aforementioned simulation scenario, 80% of the hosts can position themselves with a location error of at most 0.8 meters, compared to 1.5 meters in the case all hosts are stationary, and the same percentage of nodes can locate themselves with a location error of at most 2.6 meters, when 60% of them are mobile.

Finally, we evaluated the range error in ICS-FORTH by taking real-life signal strength measurements in this environment. We found that 90% of the total area corresponds to a maximum of 10% of the transmission range.

We plan to model CLS and to extend it, so that data collected during the system operation (potentially from different peers) can dynamically refine the initial training. We would like to test our system in case we have mobile landmarks. We are also interested in incorporating in the system heterogeneous devices (e.g, RF tags, sensors) and investigate their performance. We plan to find a way to set the confidence votes (weights) of landmarks and hosts. Finally, we will maintain mobility history from each user to refine the position estimations. Another extension is the use of a more theoretical framework (e.g., particle filters) to support the grid-based voting algorithm. Advances in the mobile robotics area may provide useful insight towards this goal, since particle filters have solved several hard perceptual problems in this field.

Bibliography

[1] P. Enge, P. Misra, "Special Issue on Global Positioning System," in *Proceedings of the IEEE*, January 1999.

[2] J. Hightower and G. Borriello, "A Survey and Taxonomy of Location Sensing Systems for Ubiquitous Computing," 2001. UW CSE.

[3] "NYC wireless." http://www.nycwireless.net.

[4] "The Wireless Island of Patmos, Greece." http://www.12net.gr/802.11.html.

[5] P. V. Bahl and Padmanabhan, "Radar: An In-Building RF-based User Location and Tracking System," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (Tel Aviv, Israel), March 2000.

[6] A. M. Ladd, K. E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D. S. Wallach, "Robotics-based location sensing using wireless ethernet," in *Proceedings of the Eight ACM International Conference on Mobile Computing and Netwrking (MOBICOM 2002)*, (Atlanta, GE), September 2002. ACM Press.

[7] N. B. Priyantha, A. K. L. Miu, H. Balakrishnan, and S. Teller, "The cricket compass for context-aware mobile applications," in *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, (Rome, Italy), pp. 1–14, July 2001.

[8] Nissanka Bodhi Priyantha and Anit Chakraborty and Hari Balakrishnan, "The Cricket Location-Support System," in *6th ACM MOBICOM*, (Boston, MA), August 2000.

[9] C. Savarese, J. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *Proceedings of Usenix Annual Technical Conference*, June 2002.

[10] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)," in *Proceedings of the IEEE Conference on Global Communications (GLOBECOM)*, (San Antonio), November 2001.

[11] J. Hightower, R.Want, and G. Borriello, "SpotON: An indoor 3D location sensing technology based on RF signal strength," February 2000. UW CSE.

[12] Srdan Capkun and Maher Hamdi and Jean-Pierre Hubaux, "GPS-Free Positioning in Mobile ad-hoc Networks," in *HICSS*, 2001.

[13] T. He and C. Huang and B. Blum and J. Stankovic and T. Abdelzaher, "Range-Free Localization Schemes in Large Scale Sensor Networks." In ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), San. Diego, 2003.

[14] P. V. Bahl and Padmanabhan, "Enhancements to the RADAR user location and tracking system," February 2000. MSR TR, Microsoft Research.

[15] Charalampos Fretzagias and Maria Papadopouli, "Cooperative location-sensing for wireless networks," in *Second IEEE International conference on Pervasive Computing and Communications 2004*, (Orlando, Florida), March 14-17 2004.

[16] James J. Caffery, Jr. and Gordon L. Stuber, "Overview of Radiolocation in CDMA Cellular Systems," April 1998.

[17] S. Thrun and D. Fox and W. Burgard and F. Dellaert, "Robust Monte Carlo Localisation for Mobile Robots," *Artificial Intelligence*, pp. 99–142, May 2001.

[18] Dieter Fox and Wolfram Burgard and Sebastian Thrun, "Markov Localization for Mobile Robots in Dynamic Environments," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.

[19] J. Borenstein and H. Everett and L. Feng and D. Wehe, "Mobile robot positioning: Sensors and techniques," 1997.

[20] J. Cox, "Blanche - an experiment in guidance and navigation of an autonomous mobile robot," 1991. IEEE Transactions on Robotics and Automation.

[21] Greg Welch and Gary Bishop, "An Introduction to the Kalman Filter," 1995. University of North Carolina at Chapel Hill.

[22] Maybeck, Peter S., *Stochastic Models, Estimation, and Control, Volume 1*. New York, NY: Academic Press, 1979.

[23] Sebastian Thrun, "Probabilistic Algorithms in Robotics," April 2000. Computer Science Department, Carnegie Mellon University.

[24] Haris Baltzakis and Panos Trahanias, "A Hybrid Framework for Mobile Robot Localization: Formulation Using Switching State-Space Models," *Auton. Robots*, vol. 15, no. 2, pp. 169–191, 2003.

[25] W. Griswold and R. Boyer and S. Brown and T. Truong and E. Bhasker and G. Jay and R. Shapiro, "Activecampus - sustaining educational communities through mobile technology." Technical Report CS2002-0714, UC San Diego, Department of CSE, July 2002.

[26] Roy Want and Andy Hopper and Veronica Falcao and Jon Gibbons, "The Active Badge Location System," 1992. ACM Transactions on Information Systems.

[27] R. Harle, A. Ward, and A. Hopper, "Single reflection spatial voting," in *Proceedings of the First International Conference on Mobile Systems, Applications, and Services*, (San Francisco), May 2003.

[28] John Krumm and Steve Harris and Brian Meyers and Barry Brumitt and Michael Hale and Steve Shafer, "Multi-Camera Multi-Person Tracking for EasyLiving," in *VS '00: Proceedings of the Third IEEE International Workshop on Visual Surveillance (VS'2000)*, (Washington, DC, USA), IEEE Computer Society, 2000.

[29] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) using AoA," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, (San Francisco, CA), April 2003.

[30] "Ekahau." http://www.ekahau.com.

[31] Scott Gifford, "Experiences with Location Sensing Systems at the University of Michigan." University of Michigan.

[32] John Krumm and Ken Hinckley, "The NearMe Wireless Proximity Server." The Sixth International Conference on Ubiquitous Computing, UbiComp 2004, Microsoft Research.

[33] "E911." http://www.fcc.gov/911/enhanced/.

[34] Andreas Haeberlen and Eliot Flannery and Andrew M. Ladd and Algis Rudys and Dan S. Wallach and Lydia E. Kavraki, "Practical Robust Localization over Large-Scale 802.11 Wireless Networks," in *Proceedings of the Tenth ACM International Conference on Mobile Computing and Networking (MOBICOM)*, (Philadelphia, PA), September 2002.

[35] "The Network Simulator (ns-2)." http://www.isi.edu/nsnam/ns/.

[36] "NetStumbler." http://www.netstumbler.com/.

[37] L. Wall, T. Christiansen, R. L. Schwartz, "Programming Perl," 1996. O Reilly and Associates, Inc.

[38] James Singer, "Elements of numerical analysis," 1964. New York, Academic Press.

[39] David B. Johnson and David A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," 1996. Mobile Computing.