DISTRIBUTED SENSING AND COMPUTING IN WIRELESS NETWORKS

by

Mojgan Khaledi

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

School of Computing

The University of Utah

December 2017

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Mojgan Khaledi**
has been approved by the following supervisory committee members:

Sneha Kumar Kasera ,	Chair(s)	**13 June 2017** Date Approved
Neal Patwari ,	Member	**13 June 2017** Date Approved
Jeffrey M Phillips ,	Member	**13 June 2017** Date Approved
Jacobus Van der Merwe ,	Member	**13 June 2017** Date Approved
Hari Sundar ,	Member	**13 June 2017** Date Approved

by **Ross Whitaker** , Chair/Dean of
the Department/College/School of **Computing** and by
David B. Kieda , Dean of The Graduate School.

ABSTRACT

We are seeing an extensive proliferation of wireless devices including various types and forms of sensor nodes that are increasingly becoming ingrained in our daily lives. There has been a significant growth in wireless devices capabilities as well. This proliferation and rapid growth of wireless devices and their capabilities has led to the development of many distributed sensing and computing applications. In this dissertation, we propose and evaluate novel, efficient approaches for localization and computation offloading that harness distributed sensing and computing in wireless networks. In a significant part of this dissertation, we exploit distributed sensing to create efficient localization applications. First, using the sensing power of a set of Radio frequency (RF) sensors, we propose energy efficient approaches for target tracking application. Second, leveraging the sensing power of a distributed set of existing wireless devices, e.g., smartphones, internet-of-things devices, laptops, and modems, etc., we propose a novel approach to locate spectrum offenders. Third, we build efficient sampling approaches to select mobile sensing devices required for spectrum offenders localization. We also enhance our sampling approaches to take into account selfish behaviors of mobile devices. Finally, we investigate an attack on location privacy where the location of people moving inside a private area can be inferred using the radio characteristics of wireless links that are leaked by legitimate transmitters deployed inside the private area, and develop the first solution to mitigate this attack.

While we focus on harnessing distributed sensing for localization in a big part of this dissertation, in the remaining part of this dissertation, we harness the computing power of nearby wireless devices for a computation offloading application. Specially, we propose a multidimensional auction for allocating the tasks of a job among nearby mobile devices based on their computational capabilities and also the cost of computation at these devices with the goal of reducing the overall job completion time and being beneficial to all the parties involved.

To my parents, my husband, and my sweet little one Mehrad.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First, I would like to thank my advisor Professor Sneha Kasera for his guidance throughout my PhD journey. Second, I would like to express my special thanks to Professor Neal Patwari for his insightful scientific advice, and for all of his help in all of my papers. Next, I have to thank my committee members, Professor Jeff Phillips, Professor Jacobus Van der Merwe, and Professor Hari Sundar for their time and comments. I am also want to thank my labmates Sriram, Arijit, Hemma, Naveen, Phil, Shamik, Aarushi, Shobhi, and Harsimran for all of their help and interesting discussions that we had in the last five years.

There are no words to express my gratitude to my parents for their continuous support and their endless sacrifices. I want also to express my special thanks to my brother Mehrdad for all of his insightful technical advice and for all of his support and help. Finally, I would like to express my special appreciation and thanks to my beloved husband Gholamreza and my sweet little one Mehrad for standing beside me throughout my PhD journey.

CHAPTER 1

INTRODUCTION

We are seeing an extensive proliferation of wireless devices including various types and forms of sensor nodes that are increasingly becoming ingrained in our daily lives. Wireless devices keep growing in sales and this growth in sales is expected to continue in the future. Furthermore, various advances in technology are making these devices powerful tools capable of performing complex tasks.

The ever-increasing popularity of wireless devices has created a great opportunity for distributed sensing. Distributed sensing tries to harness pervasive wireless devices for collecting sensing data and utilizing the collected data to provide information about people and their surroundings. It has many applications including environment monitoring, and health-care, among others. In this dissertation, we exploit distributed sensing to create novel efficient localization applications. First, we use the sensing power of a set of Radio frequency (RF) sensors and propose energy efficient localization approaches to track people and objects without requiring them to carry any wireless devices. Second, leveraging the sensing power of a distributed set of existing wireless devices (e.g., smartphones, internet-of-things devices, laptops, and modems, etc.) we propose a novel approach to locate spectrum offenders. Third, we build efficient sampling approaches to select mobile sensing devices required for spectrum offenders localization. We also enhance our sampling approaches to take into account selfish behaviors of mobile devices. Our goal is to select a set of wireless devices that provides maximum coverage for the monitored area considering mobility of both the sensing and the offending devices as well as selfish behaviors in a timely manner. Apart from the above benefits, distributed sensing in wireless networks can lead to invasion of privacy. In this dissertation, we also investigate an attack on location privacy where the location of people moving inside a private area can be inferred using the radio characteristics of wireless links that are leaked

by legitimate transmitters deployed inside the private area, and develop the first solution to mitigate this attack.

The proliferation and rapid growth of wireless devices and their capabilities also have created a huge opportunity to harness the computing power of nearby wireless devices for computation offloading. In this dissertation, we propose a multidimensional auction for allocating the tasks of a job among nearby mobile devices based on their computational capabilities and also the cost of computation at these devices with the goal of reducing the overall job completion time to the benefit of all the parties involved.

1.1 Energy Efficient Target Tracking Application

Radio tomographic imaging (RTI) is an emerging technology that harnesses the sensing capability of a distributed set of RF sensors to track people and objects without requiring them to carry any wireless transmit or receive devices [1]. Moving people and objects can be located based on the changes they cause in the received signal strength (RSS) of the radio links they perturb between transmit and receive sensor nodes. This "device-free" localization of physical objects has several applications including surveillance, rescue operations, and residential monitoring [2]. One critical issue which has been neglected in existing RTI approaches is energy efficiency. RTI has primarily focused on location accuracy and assumed that sensors are connected to wall power sockets or their batteries can be recharged often. When deploying RTI in outdoor settings where wall power is not an option, saving energy of sensor nodes becomes a key requirement. It might not be easy to recharge sensor nodes after deploying them. Even in indoor environments, wall power might not always be available. Therefore, it is desirable to reduce the energy consumption of RTI sensor nodes as much as possible to prolong the lifetime of the RF sensor network. In this dissertation, our goal is to develop approaches to reduce the energy consumption in RTI methods without giving up accuracy.

We develop energy efficient target tracking methods that essentially limit the number of radio links that we must measure at any given time thereby allowing us to deactivate a large number of transceivers and hence save energy on these nodes. Our key idea is to only measure those links that are near the current location of the moving object. In order to find the effective links near to the current location of the moving object, we propose two

approaches. In our first approach, we only consider links that are in an ellipse around the velocity vector from the current location of the moving object. In our second approach, we only consider links that cross through a circle with radius r from the current position of the moving object. Thus, rather than creating an attenuation image of the whole area in RTI, we only create the attenuation image for effective links in a small area close to the current location of the moving object. We also develop an adaptive algorithm for determining r. We evaluate the proposed approaches in terms of energy consumption and localization error in three different test areas: an open indoor area, a cluttered office, and the aisles of the University of Utah campus bookstore. Our experimental results show that using our approach, we are able to save 50% to 80% of energy. Interestingly, we find that our radius-based approach actually increases the accuracy of localization.

1.2 Simultaneous Power-Based Localization of Transmitters for Crowdsourced Spectrum Monitoring

RTI works with a small number of sensors in a limited environment. However, there are newer paradigms such as crowdsourced sensing that take advantage of the pervasive wireless devices to collect and evaluate data beyond the scale of what was previously possible. We envision a novel and scalable approach that crowdsources the sensing and localization for a very important problem: that of detecting and localizing spectrum violators. We assume a distributed set of wireless devices, e.g., smartphones, RF sensor nodes, laptops, access points and modems, etc., will participate by *sensing* the use of different bands of the spectrum over time and space and sharing their measurements with a detection and localization module in a (cloud) server, as shown in Figure 1.1. This module requests and collects spectrum usage information from a variety of sensors. It compares the spectrum usage with the allowed spectrum usage information (spectrum policies and regulations, frequency bands, locations, etc.) available in a database to determine and locate spectrum offenders. This spectrum usage database is akin to the whitespace database (e.g., an FCC-approved database that contains information on available whitespaces and their locations).

Towards the fulfillment of this vision, in this dissertation, we focus on crowdsourced localization of spectrum offenders. Due to privacy concerns and bandwidth and energy

constraints, it is undesirable for mobile sensing devices to share raw signal samples with a central server, and hence, in our work, devices collect only received power measurement. Our mobile sensing devices do not decode any signals. A key challenge in localizing spectrum offenders is that of simultaneous localization of multiple transmitters.[1] We develop a simple yet efficient and accurate approach that simultaneously localizes multiple transmitters using the power of sum of all signals received at the selected wireless devices. Our localization approach relies on the fact that, even when multiple transmissions overlap, typically the vast majority of power received by a receiver is from the nearest transmitter. Therefore, by finding local maxima in the spatially distributed RSS measurements, we can approximate the region of presence of each transmitter. We can then convert the problem of simultaneous multiple transmitter localization to a set of single transmitter localization problems and use a matrix inversion approach to find the location of each transmitter.

We experimentally evaluate our approach in two different settings: 1) an open environment with nonuniformly distributed receivers in the Orbit testbed [3] using USRP2 nodes for transmitting and receiving signals, and 2) a cluttered office with 44 uniformly distributed sensors [4]. Our experimental results show that using SPLOT we are able to localize multiple transmitters with high accuracy and in a timely manner. The highest average localization error using SPLOT measured in the open environment is 1.16 meters for up to 4 simultaneously transmitting transmitters, and the highest average localization error in the cluttered office with mobile transceivers is 2.14 meters. In comparison, the highest average localization error in Quasi EM measured in the open environment is above 6 meters. Our results also show that SPLOT is tens of minutes faster than Quasi EM. We also implement SPLOT on commodity devices and perform multitransmitter localization in a variety of indoor and outdoor experiments. We find SPLOT to significantly outperform Quasi EM in these settings as well.

1.3 Sampling for Crowdsourced Spectrum Monitoring

An important aspect of offender localization is the task of selecting a set of mobile sensing devices for RSS measurements. We build efficient sampling approaches to select

[1] For instance, a malware-based attack could simultaneously cause many devices to violate rules or jam the spectrum.

mobile sensing devices required for RSS measurements. Our goal is to select a set of wireless devices that provides maximum coverage for the monitored area considering mobility of both the sensing and the offending devices. We define and use a new metric called *degree expansion*, and propose two sampling approaches: 1) Greedy sampling and 2) Metropolis sampling. We also make our sampling efficient for handling mobility by using the information from the previous sampling interval in the current sampling interval and providing an adaptive approach to determine the number of required sensing devices in each sampling interval. Next, we enhance our sampling approach to incentivize mobile users such that we select nodes that maximize coverage but minimize the total payoff. Furthermore, our incentive mechanisms motivate mobile sensing devices to act truthfully. Our *truthful sampling* considers both the budget limit and mobility of mobile sensing devices.

There are a great number of works that have focused on selecting a set of sensor nodes to provide the maximum coverage in wireless sensor networks (WSNs) (e.g., [5–7]). There are also a few existing works that consider both incentive and the coverage problem [8–10]. However, unlike our work, none of these existing works consider mobility, truthfulness, and coverage problem all together. [11] tries to maximize the expected coverage for long-term participation that does not apply to mobile environments where mobile sensing devices can be too temporary.

We evaluate the impact of our proposed sampling approaches on the localization accuracy and show how we can maintain the localization accuracy by using a suitable sampling approach to select mobile sensing devices among all available mobile sensing devices. We also compare our sampling approaches in terms of coverage and efficiency of implementation with existing well known sensor sampling approaches; our evaluations demonstrate the efficiency of our approach.

1.4 Preserving Location Privacy

The distributed sensing in wireless networks also leads to invasion of privacy. In this dissertation, we investigate defense mechanisms against attacks where person location can be inferred using the radio characteristics of wireless links (e.g., the received signal strength, RSS, of wireless links). In these attacks, a person or a group has one or more

wireless devices (wireless access points/sensor nodes) deployed in an area in which they expect privacy, for example, their homes. An attacker can deploy a network of receivers which measure the received signal strength of the radio signals transmitted by the legitimate wireless devices, allowing the attacker to learn the locations of people moving in the vicinity of the devices, information that the attacker would not be able to know if the wireless devices did not exist. Such an attack is possible even when the network is otherwise secure against data eavesdropping.

We develop the first solution to the location privacy problem, where neither the attacker nodes nor the tracked moving object transmit any RF signals, using a game theoretic framework. In our game theoretic model, the defender (the genuine wireless network) deploys multiple transmitters in different locations and changes transmitters in some random or probabilistic fashion to minimize the chance of the attack receivers locating the people inside certain parts of the building. Figure 1.2 shows an overview of the attack due to radio network leakage. In this figure, an attacker is interested in monitoring an area of interest inside the building. The defender deploys four transmitters. When transmitter Tx 1 transmits, it would make most sense to place the attacker receivers in strategic area 1 to monitor the area of interest. When transmitter Tx 2 transmits, it would make most sense to place the attacker receivers in the strategic area 2 to monitor the area of interest. The attacker need not deploy attack receivers in all strategic areas because of cost. More importantly, the higher number of attack receivers the attacker deploys the higher the probability of it being detected (e.g., by security cameras or guards etc.). Furthermore, the attacker cannot "quickly" move and deploy attack receivers from one strategic area to another. Therefore, by appropriately changing the transmitter location the defender can defend against the attacker.

Note that we only show the transmitters and attack receivers in Figure 1.2. Movement can still be detected in the presence of other objects both inside and outside the monitored area [12]. Furthermore, while we show only one kind of transmitter, a WiFi access point, and only one kind of receivers, laptops, other wireless devices or nodes with wireless capabilities can also contribute to or be used to create radio network leakage attacks. Additionally, Figure 1.2 shows only one kind of building perimeter. Our research applies to other building perimeters as well.

We model this attacker-defender scenario as a Stackleberg game, which is a sequential game where the defender plays first, then attacker selects its best strategy by observing the defender's strategy. Our goal is to maximize the defender's benefit, i.e., maximize location privacy. We define utility and cost functions related to the defender and attacker actions. Using our utility and cost functions, we find the optimal strategy for the defender by applying a Greedy method.

We experimentally evaluate our game theoretic model in two different settings: in an open environment and a cluttered office. Our experimental results show that when using our approach, the minimum localization error for the attacker increases by $36\% - 240\%$. Higher localization error corresponds to more privacy. We expect the localization error for the attacker to be significantly higher for larger areas.

1.5 Computation Offloading in Mobile Cloud Computing

Rapid growth in mobile devices' processing and storage capabilities has also created a tremendous inexpensive amount of computing power that can provide the power of cloud. Fortunately, mobile devices opportunistically encounter many other mobile devices that are capable of lending their resources. Thus, instead of using a server cloud, mobile devices are able to offload computing tasks to nearby devices. Offloading compute tasks to nearby mobile devices rather than using a remote cloud through the mobile data cellular network lowers the latency and the burden on network backhaul. The nearby mobile devices, collectively and opportunistically, essentially provide the power of a cloud. In this dissertation, we use this notion of mobile cloud computing that corresponds to offloading of compute tasks to a group of nearby mobile devices connected by various types of links including D2D, WiFi Direct, Bluetooth, etc.

There is a growing amount of work to utilize mobile device computing power for cloud computing. Hyrax [13] uses the computational power of a network of Android smartphones in MapReduce. Mobile Device Clouds [14, 15] and Serendipity [16] are platforms for opportunistic computing where a mobile device offloads computing tasks to nearby mobile devices. NativeBOINC for Android [17] is another example of utilizing mobile devices' computing power. Recently, Habak et al. [18] proposed FemtoCloud where a controller executes a variety of tasks arriving at controller by using the computational

power of nearby mobile devices. SymbIoT [19] is another platform that uses the computational capability of all mobile devices within the same network to perform different tasks. However, all of these existing works primarily assume altruistic behavior in the distributed computing environment and do not carefully incentivize resource sharing. In addition, existing works ignore the heterogeneity among mobile devices in task allocation. Mobile devices can have different hardware/software and thus have different capabilities. For this reason, the execution time of a specific task can be different across mobile devices. Therefore, the task allocation mechanism must consider the heterogeneity of mobile devices to reduce the overall job completion time. Moreover, the task allocation needs to take into account the mobility of devices. In a mobile environment, the distributor may observe disconnection of mobile devices with assigned tasks, and new arrivals that might provide high computational capabilities. Thus, decisions should be made according to the dynamics of the environment. We propose a multidimensional auction for allocating the tasks of a job among nearby mobile devices based on their computational capabilities and also the cost of computation at these devices with the goal of reducing the overall job completion time and being beneficial to all the parties involved. We consider heterogeneity among mobile devices as well as node mobility in developing our methods. We also propose an incentive model to provide cooperation among nearby mobile devices for the application of mobile live video up streaming.

We evaluate our framework and methods using both real world and synthetic mobility traces. We use two models of compute jobs: a simple single job model, and a multiple job model that uses a Directed Acyclic graph to represent causal dependencies in a set of jobs. Our evaluation results show that our game theoretic framework improves the job completion time by a factor of 2-5 in comparison to the local execution of the job, in both the job models, while minimizing the number of auctions. Thus, our approach is beneficial for the distributor in terms of enhancing its performance. We also show that the nearby nodes that execute the distributor's tasks receive a compensation higher than their actual costs.

1.6 Summary of Contributions

Our key contributions in this dissertation include the following:

1.6.1 Localization Using Distributed Sensing

- We propose two effective energy efficient RTI approaches, ellipse-based and radius-based, for localization using RF sensor networks [20]. In addition, we propose an algorithm to tune the radius of circle adaptively over time in the radius-based approach. We perform extensive evaluations using real experimental data in three different test areas: an open indoor area, a cluttered office, and the aisles of the University of Utah campus bookstore.
- We present a simple yet efficient and accurate method, SPLOT, for simultaneous localization of multiple transmitters using crowdsourced measurements of received power [21]. We experimentally evaluate our approach in two different settings: 1) an open environment using USRP2 nodes in the Orbit testbed [3], and 2) a cluttered office [4]. We also implement SPLOT on commodity devices and perform multi-transmitter localization in a variety of indoor and outdoor experiments.
- We propose two sampling approaches: 1) Greedy sampling and 2) Metropolis sampling, to select mobile sensing devices required for RSS measurements. We enhance our sampling approach to incentivize mobile users such that we select nodes that maximize coverage but minimize the total payoff. Our incentive mechanisms consider truthfulness, the budget limit, and mobility of mobile sensing devices [21].
- We develop the first solution to the location privacy problem where the location of people moving inside a private area can be inferred using the radio characteristics of wireless links that are leaked by legitimate transmitters deployed inside the private area [22]. We model the radio network leakage attack using a Stackelberg game and find the optimal strategy for the defender by applying a Greedy method. We experimentally evaluate our game theoretic framework.

1.6.2 Computation Offloading in Mobile Cloud Computing

- We propose a multidimensional auction for allocating the tasks of a job among nearby mobile devices based on their computational capabilities and also the cost of computation at these devices with the goal of reducing the overall job completion time and being beneficial to all the parties involved [23]. We consider device and task heterogeneity as well as device mobility in developing our methods. We evaluate

our framework and methods using both real world and synthetic mobility traces.

Thesis statement: We express our thesis statement in a concise form as follows. With the aid of distributed sensing and computing in wireless networks, we propose novel, efficient approaches for localization and computation offloading and explore the related problems of incentives, and privacy, that arise while using a set of distributed wireless devices and provide solutions for them.

The rest of this dissertation is structured as follows. Chapter 2 describes our work on energy efficient radio tomographic imaging. Chapter 3 presents our work on Simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring. In Chapter 4, we describe the sampling approaches to select mobile sensing devices required for RSS measurements for crowdsourced spectrum monitoring. In Chapter 5, we investigate a location privacy attack in radio networks and provide a solution for that. Chapter 6 describes our work on using the compute power of nearby mobile devices for computation offloading. In Chapter 7, we conclude the dissertation and indicate directions for future research.

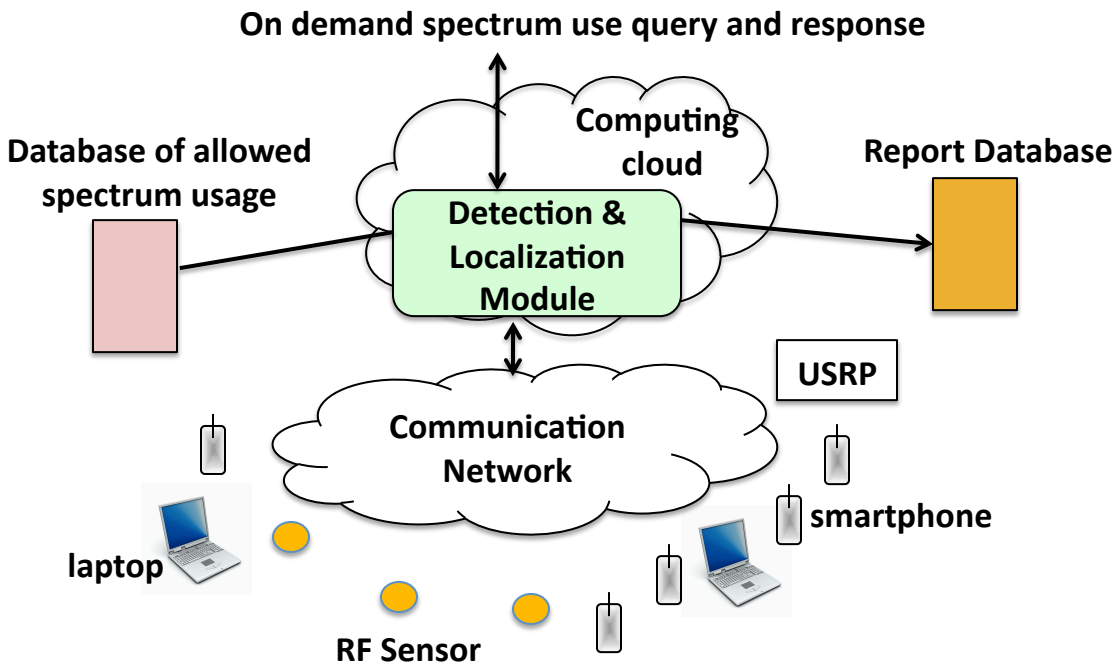

Figure 1.1. Enabling distributed set of wireless devices to detect and locate spectrum offenders.

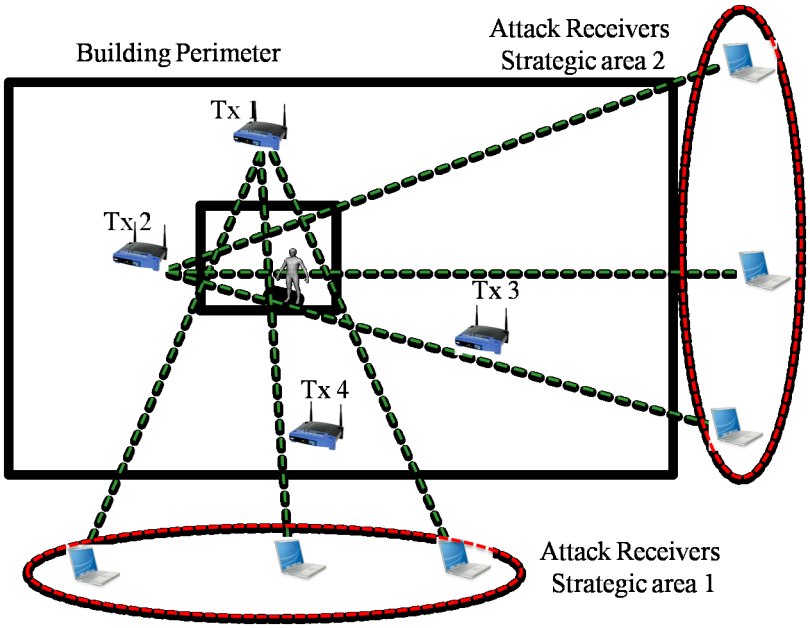

Figure 1.2. Radio network leakage attack.

CHAPTER 2

ENERGY EFFICIENT RADIO TOMOGRAPHIC IMAGING

2.1 Introduction

In this chapter, we propose two energy efficient approaches for target tracking using the sensing power of a set of distributed RF sensors.

RF sensor networks can track people and objects without requiring them to carry any wireless transmit or receive devices [1]. Moving people and objects can be located based on the changes they cause in the received signal strength (RSS) of the radio links they perturb between transmit and receive sensor nodes. This "device free" localization of physical objects is called radio tomographic imaging (RTI) [24–30]. Localization of moving people and objects using RF sensor networks has several applications including surveillance, rescue operations, and residential monitoring [2].

One critical issue which has been neglected in existing RTI approaches is energy efficiency. RTI has primarily focused on location accuracy and assumed that sensors are connected to wall power sockets or their batteries can be recharged often. When deploying RTI in outdoor settings where wall power is not an option, saving energy of sensor nodes becomes a key requirement. It might not be easy to recharge sensor nodes after deploying them. Even in indoor environments, wall power might not always be available. Therefore, it is desirable to reduce the energy consumption of RTI sensor nodes as much as possible to prolong the lifetime of the RF sensor network. In this chapter, our goal is to develop approaches to reduce the energy consumption in RTI methods without giving up accuracy.

A typical RTI setup deploys a mesh of n transceivers around the area that is to be monitored. In this setup, each node takes a turn to transmit radio signals. All the other nodes, when not transmitting, receive radio signals. The measurements of the RSS on all the $O(n^2)$ links between the transmitters and the receivers are used for tracking moving

objects in the monitored area. We develop energy efficient target tracking methods that essentially limit the number of radio links that we must measure at any given time thereby allowing us to deactivate a large number of transceivers and hence save energy on these nodes. Our key idea is to only measure those links that are near the current location of the moving object. In order to find the effective links near to the current location of the moving object, we propose two approaches. In our first approach, we only consider links that are in an ellipse around the velocity vector from the current location of the moving object. In our second approach, we only consider links that cross through a circle with radius r from the current position of the moving object. Thus, rather than creating an attenuation image of the whole area in RTI, we only create the attenuation image for effective links in a small area close to the current location of the moving object.

We propose an adaptive algorithm to change the value of r in the radius-based approach. The value of r must change over time depending on the velocity of the moving object, the number of links that cross through the circle, and the amount of error in the current location estimation. Moreover, the tracking accuracy and the energy consumption in the radius-based approach is highly dependent on the value of r. For example, if we choose a large value for r that includes all changes, then the accuracy of localization improves at the expense of relatively high energy consumption.

Our contributions in this chapter are as follows. First, we define two energy efficient approaches for localization using RTI. Our energy efficient approaches can be used with both Shadowing-based RTI and variance-based RTI [24–30]. Second, we introduce an adaptive algorithm to change the radius in the radius-based approach. Third, we evaluate the proposed approaches in terms of energy consumption and localization error in three different test areas: an open indoor area, a cluttered office, and the aisles of the University of Utah campus bookstore. Our experimental results show that using our approach, we are able to save 50% to 80% of energy. Interestingly, we find that our radius-based approach for energy efficiency actually increases the accuracy of localization.

The rest of this chapter is organized as follows. Section 2.2 contains the relevant related work. In Section 2.3, we briefly describe two basic approaches (Shadowing-based RTI, and variance-based RTI) for localization using radio tomographic imaging. Section 2.4 represents the energy efficient approaches in details. In Section 2.5, we describe three con-

ducted experiments and evaluate the results. Finally, the concluding remarks is provided in Section 2.6.

2.2 Related Work

RF sensor networks estimate the location of people and objects using the changes in RSS measured on the links of a wireless network. A growing body of research has developed approaches to improve the robustness of RF sensing to the challenges of the multipath radio channel. Experimental systems have demonstrated locating a person in offices or homes with average errors from 17 cm to 1 m [27, 28, 31–33], even across an entire office building floor [34], while tracking multiple people [35–38], even through exterior walls [29, 39, 40]. Moreover, measurements from standard wireless devices have been shown experimentally to enable reliable gesture recognition [41], breathing rate estimation [42], and fall detection [43, 44]. These experimental results are impressive despite the fact they are predominantly tested in the 2.4 GHz band where the REM system experiences interference from WiFi.

Several general approaches exist for using RSS in an RF sensor network for localization. One is radio tomographic imaging (RTI), which estimates a map of the activity in the area of deployment, computed from the changes in mean [24–28] or variance [29, 30] of RSS. Another perspective is to use machine learning to estimate position, using a prerecorded set of training data with a person in each position [34, 37, 44]. Another approach is to use line crossing information directly to infer position, either using a geometrical model [31, 36] or a statistical model for RSS given person location (and statistical inversion) [32, 33].

The energy consumption of RF sensor networks has been addressed by very few works, to our knowledge. Sensor nodes in RF networks have limited supply of energy and it is often difficult to recharge them after deployment. Thus, in order to increase the lifetime of network, energy consumption should be reduced. One way to reduce the energy required for data collection is to have sensors locally process RSS data and decide when a link is crossed, and forward data only when a link is detected as crossed [45]. This idea is complementary to this proposed work, which would turn off sensors' RSS data collection if the link is not expected to be crossed due to current human positions. Compressed sensing (CS), which estimates tomographic images using fewer link measurements, has been tested

for tomographic building structure imaging [46, 47] and for open-area attenuation-based RTI [48]. These papers explore sensing strategies that are fixed, even if random, not a function of the current positions of people. In contrast, we present an adaptive strategy. Future work might combine the benefits of both CS and adaptive strategies.

In contrast to energy efficiency in *RF* sensor networks, research in energy efficiency in wireless sensor networks (WSN) is significantly more mature [49–51]. A general WSN uses transmission and reception purely for data communication, not to measure the environment as in an RF sensor network. In an RF sensor network, it is insufficient to find a low-energy route from source to destination. Instead, we have a tradeoff between activating transmitter and receivers to minimize tracking error, and allowing them to sleep to minimize energy consumption. The adaptive algorithm explored in this chapter addresses this fundamental tradeoff.

2.3 Background on Radio Tomographic Imaging

Radio tomographic imaging is the process of imaging the attenuation caused by physical objects moving in an RF sensor network. This image can be used to find the location of the moving objects or people within the area of deployed RF network.

In an RF network with n static sensor nodes, there are $L = n(n-1)$ directed links. Let vector $y = [y_1, y_2, \ldots, y_L]^T$ be the measurement for all links. Also, let $x = [x_1, x_2, \ldots, x_M]^T$ be the vector of voxels values, where M is number of voxels. Then, y can be modeled as:

$$y = Wx + n \tag{2.1}$$

where, n is $L \times 1$ vector that represents the noise level of L links. W is a $L \times M$ matrix where w_{ij} indicates how the voxel j's attenuation affects link i.

The weight matrix, W, is modeled by an ellipse [25, 28, 29]. In this model, an ellipsoid with foci at the transmitter and receiver locations determines the weight for each link. If voxel j falls outside of link i's ellipsoid, w_{ij} is set to zero. Otherwise, w_{ij} is set to the constant, which is inversely proportional to the square root of the link length, as follows:

$$w_{ij} = \begin{cases} \frac{1}{\sqrt{d}} & d_{rj} + d_{tj} < d + \lambda \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

Here, d is the length of link i, d_{rj} and d_{tj} are length of line from the center of voxel j to the

receiver and transmitter locations of link i, respectively, and λ represents the width of the ellipse.

In the shadowing-based RTI, y is the change in RSS mean values and in variance-based RTI, y is the windowed variance of RSS on each link. The image vector x is estimated from y. However, finding y is an ill-posed inverse problem. Thus, regularization is required to reduce the noise in the image. A regularized least square approach [24, 30] is used for both shadowing-based RTI and variance-based RTI:

$$\hat{x} = \Pi y \tag{2.3}$$

$$\Pi = (W^T W + \sigma_N^2 C_x^{-1})^{-1} W^T \tag{2.4}$$

in which σ_N^2 is the noise variance. The prior covariance matrix C_x is obtained by using an exponential spatial decay:

$$[C_x]_{jl} = \sigma_x^2 e^{-d_{jl}/\delta_c} \tag{2.5}$$

where δ_c tunes the amount of smoothness in the image, σ_x^2 is the variance of voxel attenuation, and d_{jl} is the length of line between voxel j's center and voxel l's center.

2.4 Energy Efficient RTI Approaches

In this section, we describe two energy efficient approach for RTI that limit the number of radio links that we must measure at any give time thereby allowing to deactivate a large number of transceivers and hence save energy on those nodes. In the basic RTI approach, all transceivers are active at all times.

To reduce the energy consumption, instead of forming the image of attenuation for the whole monitored area, we only construct the image of attenuation for the small area near the current position of the moving object. We expect a temporal dependency in the moving pattern of object. Therefore, the next location of the moving object is likely to be close to its current location. Additionally, the moving object only changes the RSS of the links that are near the moving area. In fact, as the distance of a link from the moving object increases, it provides less information about the attenuation caused by the moving object. Thus, only links that are within a certain distance from the current location of the moving object contribute to the formation of the attenuation image. We refer to these links as *effective*

links. Our aim is to save energy by measuring only the effective links and deactivating those transceivers that do not construe these links.

To find the effective links, we use two approaches. In the first approach, the effective links are those that cross through the ellipse around the velocity vector from the current location of the moving object. In the second approach, effective links are within the circle of radius r from the current location. We explain these two approaches in Sections 2.4.1 and 2.4.2, respectively. Then, in Section 2.4.3, we describe a scheduling policy for activating and deactivating appropriate transceivers to reduce the energy consumption.

2.4.1 Ellipse-based Approach

Using the fact that a typical moving object changes its velocity and its direction of movement smoothly and gently instead of abruptly, in the ellipse-based approach, we only consider an ellipse around the velocity vector from the current position of the moving object to form an attenuation image for localization. Figure 2.1 shows the ellipse where $\overrightarrow{v}$ and θ are the velocity vector and direction of movement, respectively. The point c is the current location of the moving object.

To estimate the velocity vector, we use the history of movements over a window of time t. The velocity vector at time $t+1$ is a function of the moving object locations during $[1, t]$ as follows:

$$\overrightarrow{v}_{t+1} = \overrightarrow{v}_t + \epsilon \tag{2.6}$$

$$\epsilon \sim N(0, \sigma^2) \tag{2.7}$$

$\overrightarrow{v}_t$ is obtained by dividing the path length by the time taken to traverse the path. A path is the longest straight line that the moving object can traverse without changing the direction or pausing. Since the object need not necessarily move in a straight line and there are likely to be some deflections around the straight line, we are likely to see some small paths around the straight line. To address these deflections, we fit a straight line between two points with distance greater than l_{min} such that the other points between these two points are a distance less than w from the line.

We first run the basic RTI approach for t seconds. Using the location histories of the moving object over time t, we estimate the velocity vector at time $t+1$. After that, we only consider links that are in the ellipse around the velocity vector. Again, in time period

$t+2$, we set the window for estimating the velocity vector by 1. Given that we use the ellipse-based approach from time $t+1$ onwards, our estimation of the velocity vector can contain errors. This error can gradually increase and the ellipse that we use can move far away from the actual location of the moving object.

To tackle this potential for increase in error, we periodically (with time $T > t$) compare the result of localization of our approach with the result of localization of the basic RTI approach. If the difference between the positions obtained by the two approaches is greater than a threshold, we run the basic RTI approach for time t to obtain a more accurate history of movement of the object. Although, this reestimation of the history reduces the localization error, it also increases the energy consumption because we run the basic approach more often.

The ellipse approach can reduce the number of measured links significantly and, consequently, it can save energy. However, it is expected to work well only in scenarios where the changes in the mobility pattern are not drastic. In cases with drastic movements, e.g., an object moving back and forth or an object changing its direction of movement significantly because of the obstacle in the area, the localization error of the ellipse approach can be high. In a high mobility environment, the ellipse approach can consume more energy because of the need to run the basic RTI approach every T time units. Essentially, the ellipse-based approach can result in high localization errors or high energy use when the tracked object exhibits an unpredictable movement pattern. For robustness in these mobility scenarios, we propose the radius-based approach in the next section.

2.4.2 Radius-based Approach

In the radius-based approach the effective links are those that are in a circle with radius r from the current position of the moving object. Using a circle instead of ellipse reduces the error of localization in case of high changes in the mobility pattern. In the radius based approach, we start with the basic RTI approach by measuring all links. However, after finding the current location of the moving object, we only consider links that are in the circle with radius r from the current location.

One important challenge in the radius-based approach is the determination r. An improper value of r can increase the energy consumption or the localization error substan-

tially. Furthermore, the value of r should be adapted dynamically depending on (i) the number of sensor nodes that are deployed, (ii) the obstacles (e.g., bookshelf, table, sofa) that are placed in the area, (iii) the number of links that cross the circle, (iv) the velocity of movement, and (v) amount of error in location estimation. Unlike the ellipse-based approach, we find the value of r in the real time without using history of movement. Thus, the localization error does not increase over time. In the next section, we describe our adaptive algorithm for the selection of r.

2.4.2.1 Adaptive Algorithm for Radius Selection

In our adaptive algorithm for radius selection, we compare the location estimated using the radius-based approach with that estimated using the basic RTI approach every T time units. We change the value of r based on the difference between the location estimates and also based on the current state of the algorithm. Figure 2.2 shows the three states that we use in our adaptive algorithm to change the value of r. The algorithm starts in an exponential decrease, ED, state and then it goes to the linear increase linear decrease, LILD, state and then to the recovery state.

- ***ED***: The first state of algorithm is ED. In this state, we set the initial value of r to r_{max}. r_{max} is the radius of circle that covers the whole area of experiment and also all links, i.e., at the first step of ED both the radius-based and basic RTI approaches operate in a similar fashion. Then, in each time period T, we estimate the location using the radius-based approach and compare it with that obtained from the basic RTI approach. If the difference in the estimation is within a threshold, we reduce the value of r by half, otherwise we go to the LILD state. Let r_{k-1} denotes the value of r after $k-1$ time units, the next value of r in the ED state will be:

$$r_k = \max(\alpha * r_{k-1}, r_{min}) \tag{2.8}$$

  Here, $\alpha = 0.5$ is the exponential factor and r_{min} is the minimum value of r.

  The ED state finds the range of r which depends on several factors. We start with $r = r_{max}$ to cover the whole area. However, due to temporal dependency and the fact that the moving object only changes the RSS of neighboring links, r_{max} is much greater that the optimal value for r. Thus, in each time interval we exponentially

reduce the current value of r when the difference in location estimates (in comparison to basic RTI) is low. When we go to the LILD step after k time periods, the range of r is determined to be from r_{k-1} to r_k.

- ***LILD***: Given that we determine the range of r, $[r_{k-1}, r_k]$, quickly through exponential decrease in the ED state, we now adjust the value of r much more slowly, by increasing or decreasing its values linearly, to move closer to its optimal value. The value of r in this state is obtained as follows:

$$r_k = \begin{cases} \max(r_{k-1} + \beta, r_{min}) & \text{Invalid Result} \\ \max(r_{k-1} - \beta, r_{min}) & \text{Valid Result} \end{cases} \tag{2.9}$$

  An invalid result in the above equation refers to the situation when the difference in location estimation using the radius-based and basic RTI approaches is greater than a threshold. A valid result is one where the difference in location estimation is less than or equal to the threshold. Every T time units, we increase the value of r linearly if the result is invalid and decrease the value of r linearly if the result is valid. Thus, in LILD, the value of r fluctuates between r_{k-1} and r_k.

  In the LILD state, it is possible that we observe consecutive invalid results. If the number of consecutive invalid results, despite linearly increasing r, is greater than a predefined threshold then we conclude that we are not converging to the right r. This means that the current location of the moving object is far from its true location and even by increasing the value of r linearly in consecutive time periods, we are unable to find the location of the moving object correctly. In such a situation, stay in the LILD state will dramatically decrease the accuracy of localization. To avoid this, when we receive a certain number of consecutive invalid results, we go to the ED state again and set r to r_{max}. In other words, we start with the basic RTI approach to find the current location and reduce the value of r exponentially to find its right range all over again.

- ***Recovery***: Although, in the ED state, we decrease the value of r exponentially, it takes a long time for ED state to find the proper value for r and consequently the sensor nodes consumes more energy. Thus, we should reduce the number of times that we go to the ED state. For this, we add the recovery state in the return path from the LILD to the ED state. Recovery state gives us a less drastic approach to find the

proper value of r before going to the ED state.

In this state, we double the current value of r and then compare the result of localization after one time period T. If the result is valid then we go to the LILD state, otherwise we go to the ED state.

Figure 2.3 shows the changes of the value of r in these three states over time. We first start with the ED state by setting the r to $r_{max} = 8m$. Then, we reduce the value of r to half in each time slot. In this figure after four time units, the algorithm is able to find the range of changes for r. In the LILD state, we change the value of r linearly. Finally, we double the value of r in the recovery state.

2.4.3 Scheduling

In this section, we explain our scheduling policy for activating and deactivating the RF transceivers in our energy efficient approaches.

In the basic RTI approach, all sensor nodes are on at all times whether they transmit or receive signals. Figure 2.4 shows the scheduling policy for the basic RTI approach. In this figure, the deployed RF sensor network has 4 sensor nodes. To estimate the image of attenuation in the basic RTI approach, we must measure the RSS on all links in a full mesh topology. Considering the fact that in RF sensor networks the RSS of the link from a to b can be different from the RSS of the link from b to a, in the basic RTI approach we must measure $L = 12$ links.

Our scheduling policy for activating or deactivating sensor nodes is based on the Spin protocol [52]. The Spin protocol uses a token-based approach to prevent multiple sensor nodes from transmitting at the same time. In this protocol, the sensor nodes transmit in time division multiple access (TDMA) fashion with an order identified by their node IDs. When one sensor node is transmitting, all the other nodes are in the receiving mode. Figure 2.4 shows one round of scheduling in the basic RTI approach. In the first time slot, $t = 1$, the first node is in transmitting mode (TX), and all the other nodes (2, 3, and 4) are in the receiving mode (RX). At the end of one round, in case multiple channels are used, the sensor nodes switch synchronously to the next frequency channel defined by the user and use the same scheduling as the previous channel. As shown in Figure 2.4, in all time slots, all nodes are either in the transmitting or in the receiving mode.

Our energy efficient approaches are based on the fact that not all links in the full mesh topology are effective. Independent of how we find the effective links (using the ellipse-based or the radius-based approach), Figure 2.4 shows one round of the scheduling policy for the graph with reduced links ($L = 6$). In the first time slot, $t = 1$, node 1 is in the transmitting mode, node 2 and 3 are in the sleeping mode (deactivated), and node 4 is in the receiving mode. As this figure shows, in one round of an energy efficient approach, we can save 50% energy by increasing the number of times that sensor nodes are in the sleeping mode from 0 to 6.

As shown in Figure 2.4, in energy efficient approaches, reducing the number measured links does not change the number of nodes that are in transmitting mode. This happens because in most real cases, sensor nodes are placed uniformly along the perimeter. For scheduling the receivers, at the beginning of each round, the sink node sends a small control packet to all nodes. This packet determines the receiving nodes in each time slot. Let n be the number of sensor nodes, then size of the control packet for each node is n bits. A bit is set to 1, if the sensor node is in the receiving mode in that time slot. Otherwise, it is set to 0. Note the control packet is the same for all available channels and the sink node only sends it once for all channels, in case multiple channels are used for RTI.

If the sensor nodes are placed nonuniformly, then there might be some cases where the number of nodes that are in transmitting and receiving modes is reduced. In this case, the scheduling policy is entirely determined by the control packet that the sink node sends to all nodes. The control packet in this case indicates three modes (transmitting, receiving, sleeping) for each node in each time slot. Thus, the size of control packet for each node is $2n$ bits in a network with n nodes as 2-bits are required to represent three modes.

Note that the energy efficient approaches determine which bits should be set in the control packet based on the RSS of the measured links and the steps of these approaches. Once the control packet is created, it is sent to the sink node for distribution to the sensor nodes. The time complexity of the energy efficient approach is no worse than the basic RTI approach that is used for real time localization. In fact, the time complexity of the energy efficient approach is better than the basic approach. This is because, in most cases, the actual monitored area in the energy efficient approach is much smaller than the monitored area in the basic RTI approach.

2.5 Evaluation

To evaluate the energy efficient approaches, we conduct experiments in three different areas: an open environment, a cluttered office area, and a bookstore. In this section, we first describe these three areas and then present the evaluations of our energy efficient approaches in terms of energy consumption and localization error in these three areas.

2.5.1 Experiment Areas

- ***Open Environment:*** In the open environment, there are no objects or obstructions in the monitoring area. Figure 2.5 shows the layout of this experiment. As shown in this figure, 30 sensor nodes are deployed along the perimeter of a $70m^2$ area at the height of one meter from the floor. The sensor nodes transmits on channels 11, 15, 18, 22 and 26. The marker points in Figure 2.5 show the true positions of the moving person. The person starts moving at point A, then moves along the straight lines from A to B, B to C, C to D, and D to A. Finally, the person stops moving at point A. At each location, the person stands for $20s$. We also measure the RSS of all links when no person is present in the monitoring area for $60s$ and use it for RTI calibration.
- ***Cluttered Office:*** This experiment is done in a cluttered area where there are several metallic obstructions such as desks, chairs, and monitors. In this experiment, 14 sensor nodes are deployed inside of a $52m^2$ area at the height of one meter from the floor, as shown in Figure 2.6. The sensor nodes transmit on channels 11, 16, 21, and 26. The marker points in Figure 2.6 show the true locations. The person moves along the path $ABDCEFGHCEGBA$. As can be seen in the movement path, in this experimental setup, the person changes direction of movement more often than in the open environment. Using these two experiments, we are able to compare the results of ellipse and radius-based approaches in conditions where there are too many changes in the movement pattern (cluttered office) with the case where the changes in the movement pattern are minimal (open environment).
- ***Bookstore:*** This experiment is performed in the University of Utah Bookstore in a $55m^2$ area. As in the case of the office environment, the bookstore is cluttered with shelves, tables, and books. There are 34 sensor nodes that are positioned in the area, as shown in Figure 2.7. The gray rectangles in this figure are shelves. The sensor

nodes transmit only on one channel. In one experiment (which we call Exp. 1), a person moves along the path $ABCDA$ twice and in the another experiment (which we call Exp. 2), the person moves along the path $EFBAE$ twice.

In the first two experiments (open environment, and cluttered office), the sensors are TI CC2531 USB dongle nodes [53] and in the third experiment, bookstore, the sensor nodes are TelosB [54].

2.5.2 Experimental Results

In this section, we evaluate the energy efficient approaches in terms of energy consumption and the localization error. For evaluation, we compare our work with both Shadowing-based RTI and variance based RTI approaches. We use the approach that is proposed in [28] from the shadowing based approaches and VRTI [29] from the variance based RTI approaches. In [28], the authors use channel diversity to improve the accuracy of localization in RTI.

In order to evaluate the energy efficient approach, we use two metrics: average error of location estimation, and energy consumption ratio. The energy consumption ratio is obtained by dividing the total energy consumption in the energy efficient approach by the total energy consumption in the basic approach (e.g., Shadowing-based RTI, and variance-based RTI). The total energy consumption in an RF network with n nodes is obtained from the following formula.

$$E = \sum_{t=1}^{T} \sum_{i=1}^{n} E_{Tx}(i,t) + E_{Rx}(i,t) \tag{2.10}$$

Here, E_{Tx} and E_{Rx} are energy consumptions in the transmitting and receiving modes respectively. We set both E_{Tx} and E_{Rx} to $4.5J$ in all experiments. Table 2.1 shows the value of parameters used in these three experiments.

Figure 2.8 shows the energy consumption of the ellipse, radius-based, and the multi-channel RTI approaches in the open environment. This figure shows that both ellipse and radius based approaches can reduce the energy consumption ratio from 1 to 0.2. In other words, both energy efficient approaches save 80% of energy compared to the multichannel RTI approach. Also, this figure shows that there is not too much difference between the energy consumption ratio of ellipse and radius-based approach (see the yellow area).

Figure 2.9 compares the energy consumption ratio of the three approaches (the ellipse, radius-based, and multichannel RTI) in a cluttered office area. This figure shows that the energy efficient approaches save around 60% of energy which is a little less than the energy savings in the open environment. This is because in the office area there are only 14 nodes which is significantly less than the number of nodes in the open area. The reduction in the number of measured links in the office area is less than that in the open area and therefore, we save less energy in the office area.

Figure 2.10 shows the average error of location estimation in the ellipse, radius-based and multichannel RTI approaches for both open and cluttered office areas. In both areas (open environment and office), the error of radius based approach is slightly less than the multichannel RTI approach. Since in the radius-based approach we only construct the image of attenuation for a circle around the current position, the noise in the other places does not effect the result of localization. However, in multichannel RTI, we consider the whole area for constructing the image of attenuation and sometimes the noise in the environment corrupts the image and increases the error of localization. In addition, as shown in Figure 2.10, the average error of localization in the ellipse approach is slightly higher than the average error in multichannel RTI approach for the open environment and it is a lot higher than the error of multichannel RTI approach in the office environment. As we discussed earlier, one drawback of the ellipse approach is propagation of error. This propagation has high impact when we have frequent significant changes in the movement pattern. In the office area, the person changes direction of movement more frequently compared to the open environment. Thus, the error of localization increases significantly in the office area.

Figure 2.11, Figure 2.12, and Figure 2.13 show the comparison of ellipse, radius-based, and VRTI approaches in the bookstore. Figure 2.11 and Figure 2.12 show that by using the VRTI approach with ellipse and radius-based approach, we can save up to 50% of energy on average compared to the basic VRTI approach. However, the amount of reduction in the energy consumption is less than that in the open environment where we use multichannel RTI with the energy efficient approaches. This happens because in the VRTI approach, we need the RSS of links for a couple of rounds to be able to compute the variance. Therefore, in the energy efficient approaches over each T time units, when we compare the basic

approach with the energy efficient approach, we need to turn on all sensor nodes for a couple of rounds. This increases the energy consumption in this scenario. However, we still cut down the energy usage by half.

As shown in Figure 2.11 and Figure 2.12, sensor nodes in Exp. 2 consume more energy than in Exp. 1. As mentioned in the previous section, in Exp. 1, the person moves along the path $ABCDA$ that covers the whole area. However, in Exp. 2, the person moves along the path $EFBAE$ that is only a part of the whole area that is surrounded by lots of sensor nodes. Therefore, the reduction in the number of measured links in Exp. 2 is less than the reduction in Exp. 1. As a result, in Exp. 2, we consume more energy compared to that in Exp. 1. Also, in the bookstore experiment there is a higher difference in the energy consumption ratio of nodes. This happens because in the bookstore environment the sensor nodes are placed nonuniformly.

Table 2.2 shows the average location estimation error, $\overline{e}_{loc}$ and the total energy consumption ratio, $\overline{E}_R$, in all the experiments. If we only consider the energy consumption ratio, the ellipse approach is better that the other approaches. However, the ellipse approach does not perform well in terms of localization. As we can see in this table, in all experiments the localization error in the ellipse approach is higher than the basic and the radius-based approaches and in some cases such as the office area the localization error is significantly higher than the error in the other approaches. In terms of localization, the radius-based approach performs better than the others. Also, it can save 50% energy in the worst case which is great. Figure 2.14 shows this comparison. In this figure, the x axis represents the total energy consumption ratio, and the y axis represents the normalized localization error. Our goal is to find an approach with points on the left bottom corner of the figure. This figure shows that the basic approach is on the right and close to the bottom. Our energy efficient approaches when applied to the basic approach, moves the points from the right to the left. Also, we decrease the distance from the bottom in the radius based approach.

2.6 Conclusion

We introduced two effective energy efficient RTI approaches, ellipse-based and radius-based, for localization using RF sensor networks. In both energy efficient approaches, our

aim was to save energy by reducing the number of links that we must measure to form an image of attenuation. In the ellipse-based approach, we only considered links in an ellipse around the velocity vector of the current position of the moving object and in the radius-based approach, we used links in a circle around the current position of the moving object. In addition, we proposed an algorithm to tune the radius of circle adaptively over time. We performed extensive evaluations using real experimental data from three different settings. Our experimental results showed that our energy efficient approaches can save 50% to 80% of energy without seriously degrading localization accuracy. Interestingly, our radius-based approach even increased the accuracy of localization in comparison to the basic RTI approach.

RTI works with a small number of sensors in a limited environment. However, there are newer paradigms such as crowdsourced sensing that take advantage of the pervasive wireless devices to collect and evaluate data beyond the scale of what was previously possible. In the next chapter, we exploit crowdsourcing for the spectrum monitoring application.

Table 2.1. Parameters used in the experiments

Parameter	Description	Open	Office	Bookstore
P	Pixel width(m)	0.15	0.5	0.5
λ	Excess path length limit of ellipse weighting(m)	0.02	0.02	0.02
σ_x	Voxels variance(dB)	0.05	0.1	0.1
σ_N^2	Noise variance(dB)	1	1	1
δ_c	Correlation coefficient	4	4	4
m	Number of used channels	4	4	1
α	Exponential factor	0.5	0.5	0.5
β	Linear factor(m)	0.5	0.5	0.5
T	Time periods for RTI	50	50	50
r_min	Min radius(m)	0.5	0.5	0.5
t	Window size for ellipse	5	5	5

Table 2.2. Comparison of different approaches

	Open		Office		Bookstore Exp.1		Bookstore Exp.2	
Approaches	$\bar{e}_{loc}(m)$	$\overline{E}_R$	$\bar{e}_{loc}(m)$	$\overline{E}_R$	$\bar{e}_{loc}(m)$	$\overline{E}_R$	$\bar{e}_{loc}(m)$	$\overline{E}_R$
Ellipse	0.172	0.1611	1.667	0.2862	1.7571	0.2715	0.8013	0.2871
Radius-based	0.1544	0.1899	0.7481	0.3998	0.9368	0.4823	0.7264	0.5029
Basic Approach	0.1693	1	0.8993	1	1.0245	1	0.7928	1

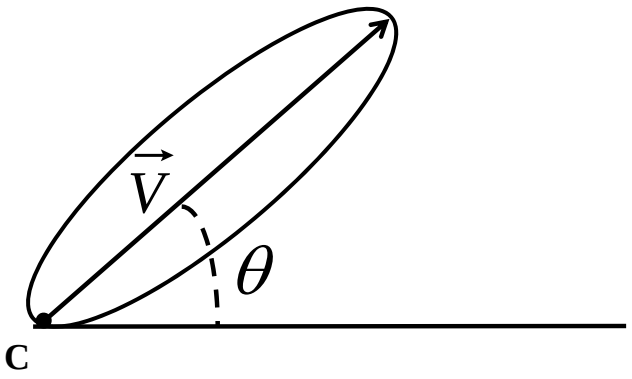

Figure 2.1. An ellipse around the velocity vector from the current location of the moving object.

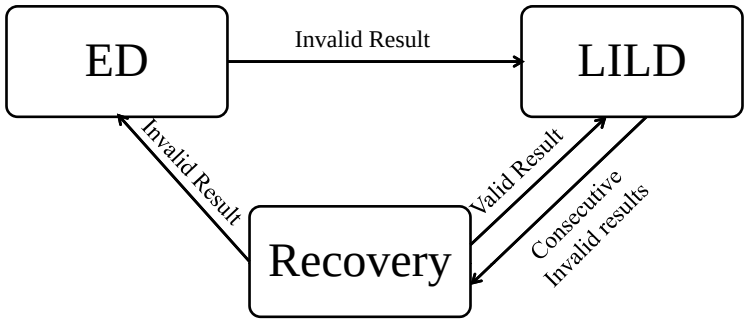

Figure 2.2. Adaptive algorithm for selection of r.

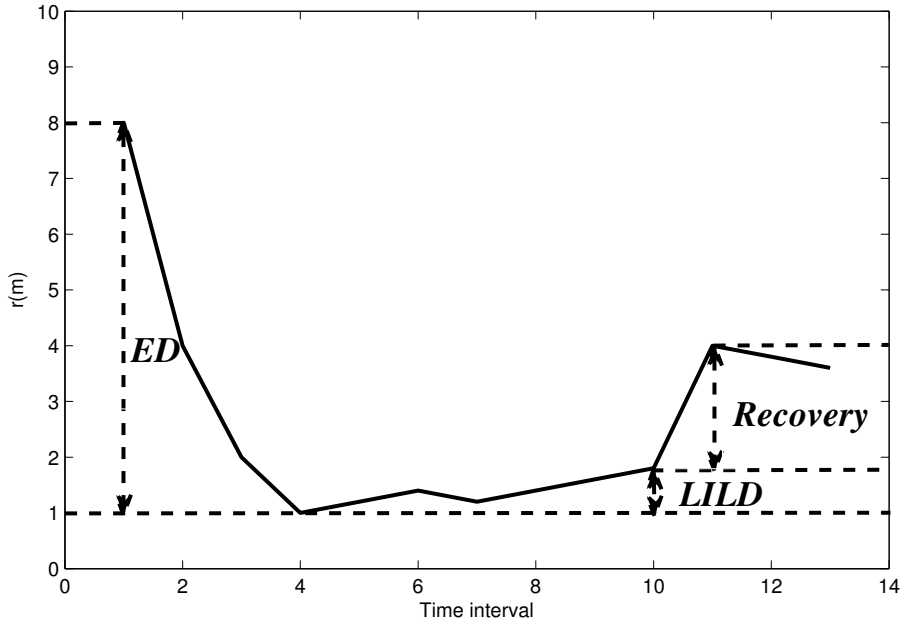

Figure 2.3. The value of r in each time period T.

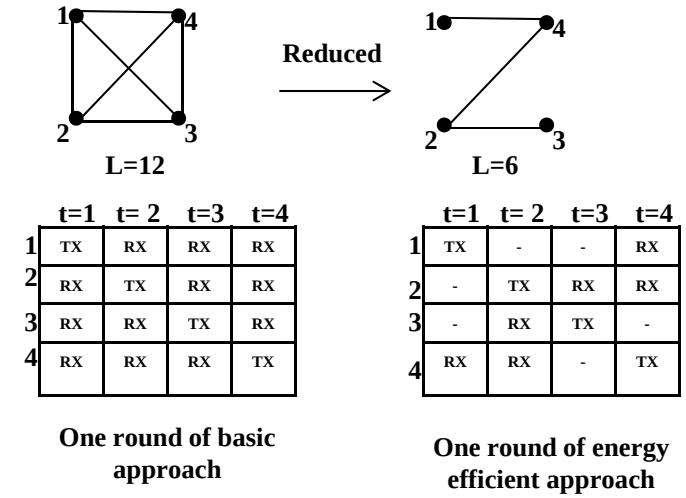

Figure 2.4. The scheduling policy for basic approach and energy efficient approach in an RF network with 4 sensor nodes.

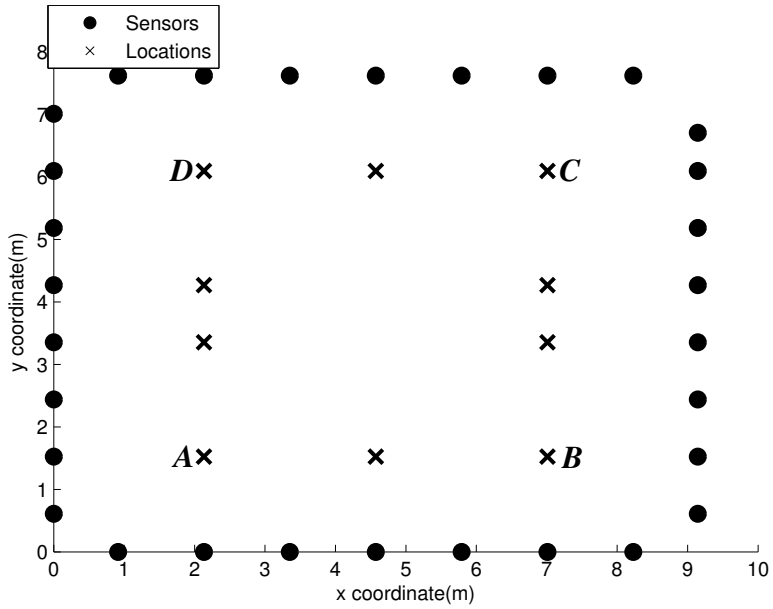

Figure 2.5. The layout and sensor nodes positions for open environment.

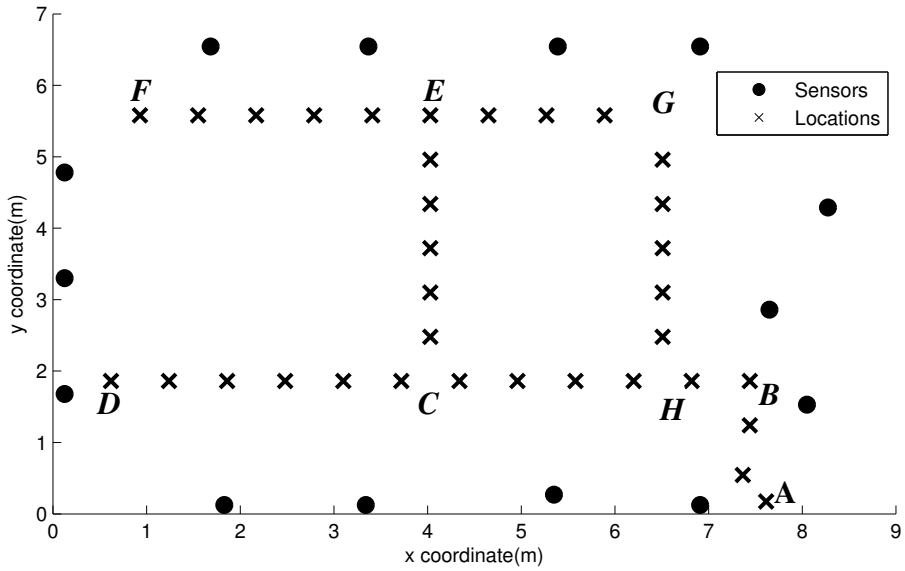

Figure 2.6. The layout and sensor nodes positions for cluttered office.

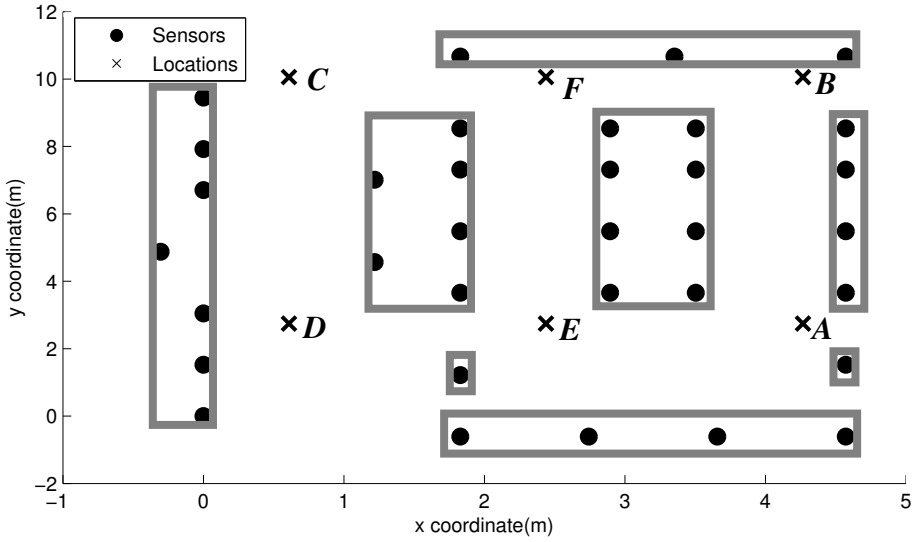

Figure 2.7. The layout and sensor nodes positions for bookstore.

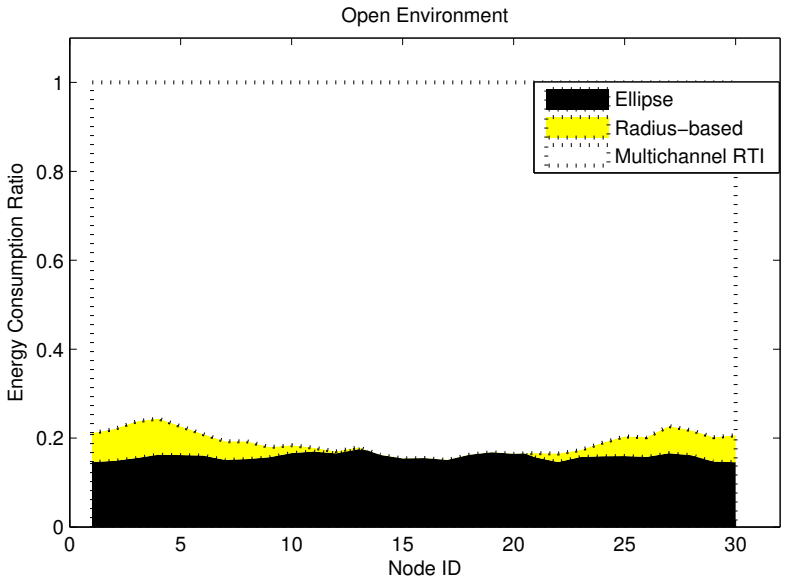

Figure 2.8. The energy consumption ratio for ellipse, radius-based, and multichannel RTI approaches in an open environment.

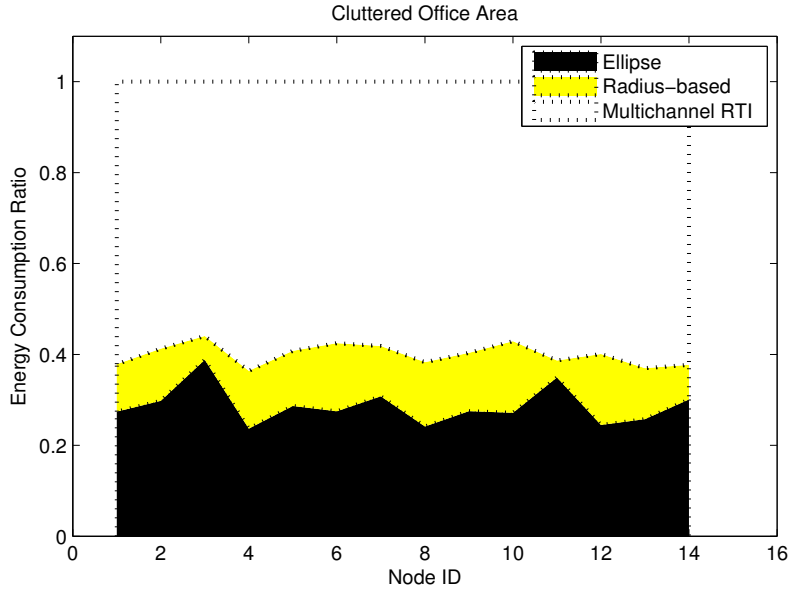

Figure 2.9. The energy consumption ratio for ellipse, radius-based, and multichannel RTI approaches in a cluttered office.

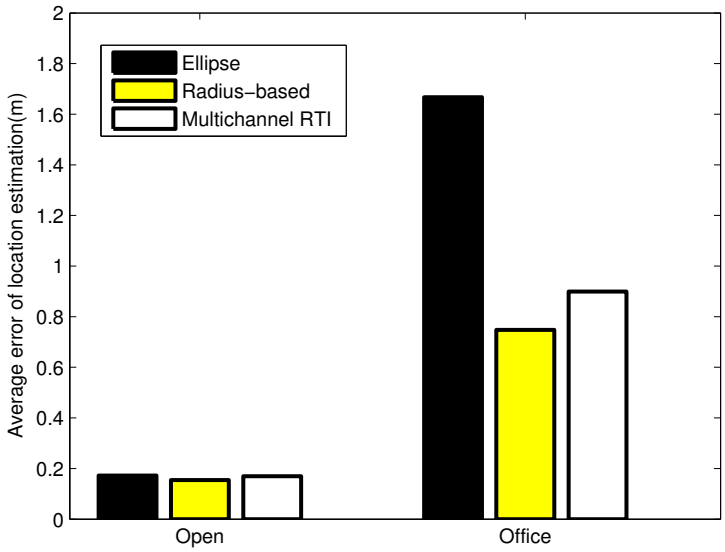

Figure 2.10. The average error of location estimation for ellipse, radius-based, and multi-channel RTI approaches in the open environment and the cluttered office.

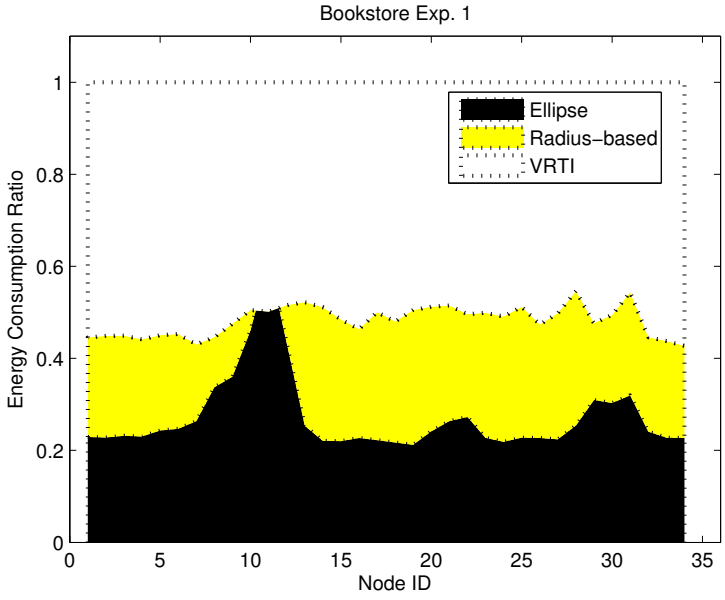

Figure 2.11. The energy consumption ratio for ellipse, radius-based, and VRTI approaches in bookstore Exp1.

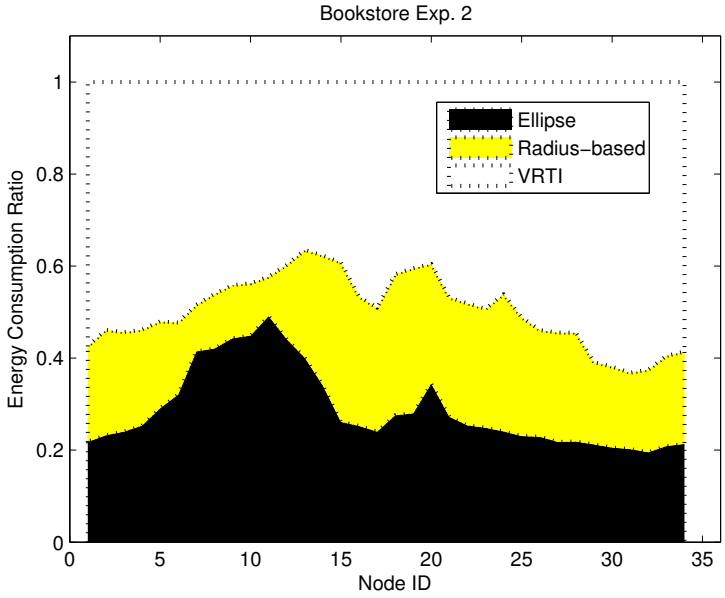

Figure 2.12. The energy consumption ratio for ellipse, radius-based, and VRTI approaches in bookstore Exp2.

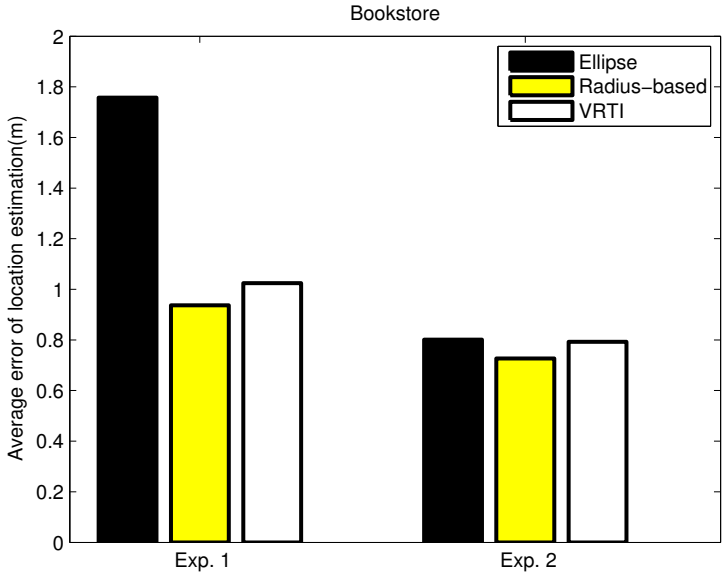

Figure 2.13. The average error of location estimation for ellipse, radius-based, and VRTI approaches in bookstore.

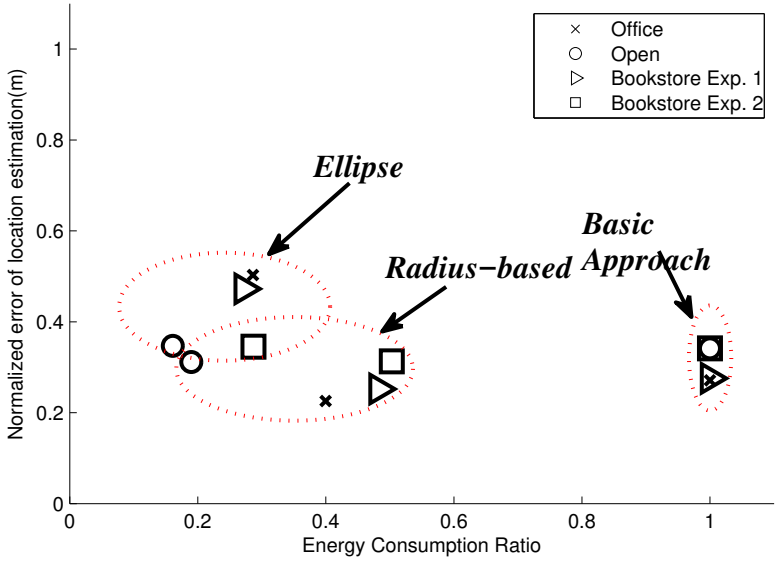

Figure 2.14. Comparison between ellipse, radius-based, and basic approaches in terms of the total energy consumption ratio and localization error.

CHAPTER 3

LOCALIZATION FOR CROWDSOURCED SPECTRUM MONITORING

3.1 Introduction

When software defined radios (SDRs) become ubiquitous, i.e., in the hands and pockets of average people, it will be easy for a selfish user to alter his radio(s) to transmit and receive data on unauthorized spectrum, for example, using an off-limits band, or transmitting/receiving on a channel when another device has priority. In addition, SDRs infected by a computer virus or malware could exhibit illegal spectrum use without the user's awareness. The U.S. Federal Communications Commission has an enforcement bureau which detects violations via complaints and extensive manual investigation. The mechanisms used currently for locating spectrum offenders are a time consuming, human-intensive, and expensive proposition. A violator's illegal spectrum use can be too temporary and mobile to be detected and located using existing processes. We envision a novel approach that crowdsources the sensing and localization of spectrum offenders. We assume a distributed set of wireless devices, e.g., smartphones, RF sensor nodes, laptops, access points and modems, etc., will participate by *sensing* the use of different bands of the spectrum over time and space and sharing their measurements with a detection and localization module in a (cloud) server. This module requests and collects spectrum usage information from a variety of sensors. It compares the spectrum usage with the allowed spectrum usage information (spectrum policies and regulations, frequency bands, locations, etc.) available in a database to determine and locate spectrum offenders. This spectrum usage database is akin to the whitespace database (e.g., an FCC-approved database that contains information on available whitespaces and their locations).

Towards the fulfillment of this vision, in this chapter, we focus on crowdsourced localization of spectrum offenders. Due to privacy concerns and bandwidth and energy

constraints, it is undesirable for mobile sensing devices to share raw signal samples with a central server, and hence, in our work, devices collect only received power measurement. Our mobile sensing devices do not decode any signals. A key challenge in localizing spectrum offenders is that of simultaneous localization of multiple transmitters.[1] We develop a simple yet efficient and accurate approach that simultaneously localizes multiple transmitters using the power of sum of all signals received at the selected wireless devices. While localization of a transmitter based on power measurements taken by a network of sensors has been widely studied (see [55] for a survey), we argue that the existing work on WSNs cannot be simply adapted for localization of multiple transmitters using crowdsourcing. If two devices transmit at the same time, the received signal is a phasor sum of the signals from both. Simultaneous transmission in the same channel can be a consequence of an attacker's violation of spectrum access rules or an intentional effort to jam. In either case, the signals may be impossible to separate, particularly when receivers report only power measurements to the server. Furthermore, it is important for crowdsourcing-based localization to account for the mobility and changing availability of user devices.

Many existing methods assume that if multiple transmitters are to be located, their signals can be separated at the receivers [55]. Note that even if all receivers were sophisticated enough to perform this blind source separation, a smart adversary could simply transmit signals that are not blind separable. When multiple signals cannot be separated, the few published methods [56–58] that are able to localize multiple transmitters from power measurements have high time complexity and do not consider the mobility and temporal availability of transmitters and receivers. For example, Quasi EM [58], a statistical approach to localize multiple transmitters assumes that the transmitters and receivers are static and that the number of transmitters is known a priori. There is existing work on locating multiple transmitters using mobile robots [59]. However, this work is not applicable in our setting where mobile users move without the network control. We need a localization algorithm which minimizes time complexity without significantly compromising the localization accuracy in dynamic environments in order to detect and locate unauthorized transmitters.

---

[1] For instance, a malware-based attack could simultaneously cause many devices to violate rules or jam the spectrum.

We present a method for simultaneous power-based localization of transmitters (SPLOT) for crowdsourced spectrum monitoring. SPLOT considers the temporal availability and mobility of both receivers and transmitters and makes no assumptions about the number of transmitters. SPLOT relies on the fact that, even when multiple transmissions overlap, typically the vast majority of power received by a receiver is from the nearest transmitter. Therefore, by finding local maxima in the spatially distributed RSS measurements, we can approximate the region of presence of each transmitter. We can then convert the problem of simultaneous multiple transmitter localization to a set of single transmitter localization problems and use a matrix inversion approach to find the location of each transmitter. Notably, we only consider an approximate region of each transmitter that is a confined area around each local maximum and thus, improve the efficiency of localization in terms of time complexity, accuracy, and scalability.

We experimentally evaluate our approach in two different settings: 1) an open environment with nonuniformly distributed receivers in the Orbit testbed [3] using USRP2 nodes for transmitting and receiving signals, and 2) a cluttered office with 44 uniformly distributed sensors [4]. Our experimental results show that using SPLOT we are able to localize multiple transmitters with high accuracy and in a timely manner. The highest average localization error using SPLOT measured in the open environment is 1.16 meters for up to 4 simultaneously transmitting transmitters, and the highest average localization error in the cluttered office with mobile transceivers is 2.14 meters. In comparison, the highest average localization error in Quasi EM measured in the open environment is above 6 meters. Our results also show that SPLOT is tens of minutes faster than Quasi EM. We also implement SPLOT on commodity devices and perform multitransmitter localization in a variety of indoor and outdoor experiments. We find SPLOT to significantly outperform Quasi EM in these settings as well.

The rest of this chapter is organized as follows. Section 3.2 represents the localization approach in detail. In Section 3.3 and Section 3.4, we describe the experimental setup and the implementation, respectively. Concluding remarks are provided in Section 3.5.

3.2 Localization

In this section, we describe our approach to simultaneously locating multiple transmitters in the areas near the mobile sensing devices (also referred to here as receivers). The localization problem considered here is challenging for several reasons. First, there may be multiple transmitters, and the number of transmitters is unknown. Second, each measurement of received power is an unknown combination of the received powers from each transmitter. Third, the number and the locations of both transmitters and receivers might change from one sample to the next. Fourth, each link from transmitter to receiver experiences multipath fading, which is known to complicate RSS-based localization. We do not use time-of-arrival (TOA) methods because they require recording and sharing users' sampled signals, which violates our privacy model. We do not use angle-of-arrival (AOA) methods because they require additional radio frequency (RF) hardware. We require an efficient and accurate localization approach that can locate all available transmitters in a dynamic environment using only power measurements.

3.2.1 Methodology

Our localization methodology assumes that receivers (mobile sensing devices) have been selected using our sampling approaches and that these receivers are spread geographically across the monitored area. Let K denote the unknown numbers of transmitters and $\theta = \{\theta_1, \theta_2, \ldots, \theta_K\}$ represent their unknown two-dimensional locations. Our problem is to determine K and θ based on observed received powers reported by L receivers, $y = \{y_1, y_2, \ldots, y_L\}$.

Our localization approach relies on two observations. First, receivers that are located near the transmitter observe generally higher power than the receivers that are distant from the transmitter. The second observation is that the observed RSS at each receiver is primarily affected by the nearest transmitter. To validate this observation, we compare the RSSs observed in selected receivers when there is no transmitter, when there is only one transmitter, and when we add multiple transmitters in different locations. Our results show that when there is a transmitter near a receiver, the RSS observed by this receiver increases substantially. However, there is a small growth on the observed RSS by the receiver when adding more transmitters at more distant locations.

These two observations allow us to reduce the problem of locating multiple, unknown numbers of transmitters to that of localizing a set of single transmitters as follows. First, we find the local maxima of RSSs observed by receivers that are greater than a predefined threshold. The predefined threshold is set to the minimum RSS that a receiver observes when there is a transmitter near it. By defining this threshold, we are able to separate the local maxima due to the presence of a transmitter from the local maxima due to the different fade levels at nearby receivers.

With the knowledge of the local maxima, our localization problem can be reduced to finding K transmitters, where K is equal to the number of local maxima. Instead of locating K transmitters in the entire monitored area, we divide the problem into K single transmitter localizations. For each local maximum, we locate a single transmitter. However, we confine the area to the small area around the local maximum and only use the RSSs that are observed by the receivers in this area for localization. Restricting the area to the small area around the local maximum reduces the time complexity of localization approach, and more importantly, confines the localization area and thus increases the accuracy of localization. This is because the noise in the measurements at receivers in other areas has little impact.

After reducing the multiple transmitter localization problem to a set of single transmitter localization, we locate each single transmitter in a small area around the local maximum using a single transmitter localization method. Specifically, in this chapter, we use a matrix inversion approach for single transmitter localization designed for higher efficiency and localization accuracy (see Section 3.3.3).

3.2.1.1 Single Transmitter Localization

In this section, we apply a linear model and perform inversion to localize the transmitted power. Our approach is designed to accurately locate a source with unknown transmit power. In this linear model, we estimate the power field, i.e., the power transmitted vs. position. This power field is then used to find the location of an unknown transmitter.

Given a local maximum of RSSs observed by receivers, we find a transmitter located in a confined area around the local maximum. The confined area is a circle of radius R from the local maximum. Let vector $y = [y_1, y_2, \ldots, y_L]$ be the received powers (in linear units

of Watts) observed by L receivers located in the confined area. Also, let $x = [x_1, x_2, \ldots, x_Q]$ be the power field, where x_i represents the transmit power sent by a transmitter in voxel i. We select a grid of Q voxels that fill the confined area. We have a forward model for y,

$$y = Wx + n, \tag{3.1}$$

where n is an $L \times 1$ vector that represents the noise and fading contributing to the L RSS measurements, and W is an $L \times Q$ matrix where W_{ij} indicates the gain that would be experienced on the channel between a transmitter at voxel j and receiver i, if there is a transmitter in voxel j. The weight value W_{ij} is inversely related to the distance of the voxel and the receiver. We model the weight as,

$$W_{ij} = \begin{cases} d_{ij}^{-n_p}, & d > minPL \\ minPL^{-n_p}, & \text{otherwise} \end{cases} \tag{3.2}$$

Here, d_{ij} is the Euclidean distance from the center of voxel j to the receiver i. n_p is the path loss exponent and $minPL$ is the minimum path length.

A power field estimate $\hat{x}$ is estimated from y. However, finding $\hat{x}$ is, in general, an ill-posed inverse problem. We use a regularized least square approach to compute an estimate,

$$\hat{x} = \Pi y \tag{3.3}$$

$$\Pi = \left(W^T W + \sigma_N^2 C_x^{-1}\right)^{-1} W^T, \tag{3.4}$$

where σ_N^2 is the noise variance, W^T is the transpose of matrix W, and the prior covariance matrix C_x is obtained by using an exponential spatial decay function,

$$[C_x]_{jl} = \sigma_x^2 e^{-f_{jl}/\delta_c}. \tag{3.5}$$

Correlation distance constant δ_c describes the distance at which two voxels have correlation coefficient $1/e$, σ_x^2 is the variance of the transmit power field, and f_{jl} is the length of the line between the centers of voxels j and l.

Finally, the transmitter location is estimated to be the center of voxel with maximum value of $\hat{x}$.

3.2.1.2 Dynamic Localization

We also address the problem of locating mobile transmitters. We consider two dynamic cases: 1) the number and locations of transmitters are changing, and 2) the number and location of both transmitters and receivers are changing.

One approach to locate these dynamic cases is to repeat the multiple transmitter localization procedure every time new received power measurements are made. This approach is not efficient since changes are likely to be small from one time interval to the next. Thus, in this section, we present methods to use the coordinates from the previous estimate in the current localization problem.

- ***Dynamic transmitters:*** When changes happen only in the number and location of *transmitters*, our dynamic localization works as follows.

  The localization module uses three data sources at time t: 1) the RSS measurements of the selected receivers in time $t-1$, which we denote y^{t-1}; 2) the RSS measurements of the receivers in time t, y^t; and 3) the locations of detected transmitters in time $t-1$, which we denote θ^{t-1}. The first step is to estimate the number of transmitters, which we accomplish by comparing y^{t-1} with y^t. If there is no significant change, the localization module sets $\theta^t = \theta^{t-1}$. If the RSS increases significantly for some receivers, the localization module performs multiple transmitter localization for the area that is covered by these receivers to find new transmitters. If the RSS decreases significantly for some receivers, the localization module removes previous transmitters if there are receivers with significant decrease in the RSS within the confined area (circle of radius r) around the previous transmitters. The set of transmitters at time t, θ^t is the union of the remaining transmitters from the previous time and the newly added transmitters.

  Note that by significant change, increase or decrease, in RSS, we mean any changes in the RSS that is greater than a predefined threshold. The predefined threshold is determined by the localization module depending on the environment. This predefined threshold is equal to the minimum change in the RSS measurements of receivers when at least one transmitter is added to or removed from the environment.

  One drawback of the above approach is the propagation of error. Since the localization module estimates current transmitter locations using the previous results,

the error can be carried forward from one time interval to the next, and after some time, the detected transmitters may be far from the actual transmitters. To deal with this problem, we periodically reinitialize by performing the multiple transmitter localization algorithm without any previous time estimates.

- ***Dynamic transmitters and receivers:*** When the number and the location of receivers also change over time, we cannot compare two consecutive RSS measurements on the same link, as the change in receiver position will cause significant RSS changes. Even by mapping the receivers in the previous time, $t-1$, to the nearest receiver available in time t and inversely relating the RSS to the distance between mapped receivers, we are not able to compare the RSSs in two consecutive time slots. This is due to the fact that RSS may be different for receivers with the same distance from the transmitter due to both shadowing and small-scale fading. Therefore, while the number or locations of receivers are changing, we perform multiple transmitter localization without using the previous time information. However, we limit the recalculation to at most every T seconds.

3.3 Experimental Setup

To evaluate our sampling and localization approaches, we conduct our experiment in two different areas: an open environment and a cluttered office area. In this section, we describe these two areas and the mobility settings for transmitters and receivers. We also explain the evaluation metrics. The values of parameters used in our localization and sampling are listed in Table 3.1.

3.3.1 Test Environment

- ***Open environment:*** In the open environment, there are no objects or obstructions in the area. This experiment is performed on the Orbit testbed [3], using the USRP2 nodes to transmit and receive signals. Figure 3.1 shows the layout of this experiment. As shown in this figure, 14 receivers and 4 transmitters are placed nonuniformly inside of a 20 by 20 m area. The transmitters send a sinusoidal continuous wave (CW) signal, and the receivers measure received signal strength (RSS) at the same frequency.

- ***Cluttered office:*** The office area is cluttered with desks, bookcases, filing cabinets, computers, and equipment. The collected data is a public data set [4]. In this experiment, 44 sensors are placed randomly in a 14 by 13 (m) area as shown in Figure 3.2. Unlike the open environment, in the cluttered office the sensors are randomly distributed in the environment. Here, the transmitters are transmitting sequentially, and as this is a previously collected public experimental data set, we cannot change it to have multiple transmitters transmitting simultaneously. To approximate what the RSS would have been if multiple transmitters had transmitted simultaneously, we use the sum of linear received powers measured from each transmitter when transmitting sequentially. We justify this approximation as follows. First, we note that the expected value of the power in sum of multiple signals is the sum of the powers of those individual signals [60]. Second, we perform a validation experiment in the Orbit testbed. For this validation, we randomly select two USRP2s and set them to transmit. We denote the receiver linear power measurements as RSS(TX1) and RSS(TX2) when the transmitters transmit sequentially, and RSS(TX1,TX2) when they transmit simultaneously. Figure 3.3 shows the comparison of RSS(TX1)+RSS(TX2) vs. RSS(TX1,TX2). The data shows, in fact, that the linear approximation is very accurate, almost always within 0.5 dB.
- ***Changes in number and locations of transmitters and receivers:*** To approximate mobility despite the fixed locations of mobile sensing devices in the two testbeds, we turn on and off the mobile sensing devices in the open environment and the cluttered office. To model the on and off states of each transmitter, we use a two state continuous time Markov chain that is able to model the bursts of device availability (on state) [61], with state 0 indicating Off (or not transmitting) and state 1 indicating On (transmitting). To calculate the probabilities, we need the time interval between two decisions on whether or not to perform state transitions which is modeled to be exponentially distributed with parameter λ, the bursts of transmitting for mobile sensing devices (the expected number of consecutively transmitting), b, and the probability that the mobile sensing device is transmitting, p (for more information see [61]). We can find the on and off states of each device once we determine the probabilities.

3.3.2 Evaluation Metrics

1. *Localization error:* The localization error, ϵ_l, is equal to the root mean squared error of the best assignment between estimated and actual transmitter locations [62].

$$\epsilon_l(\theta, \hat{\theta}) = \frac{1}{|\hat{\theta}|} \min_{a \in A} \left(\sum_{i=1}^{|\theta|} \sum_{j=1}^{|\hat{\theta}|} a_{i,j} d(\theta_i, \hat{\theta}_j)^2 \right)^{\frac{1}{2}} \tag{3.6}$$

   where A is the set of all permutations between the set of actual transmitter locations θ and the set of estimated transmitter locations $\hat{\theta}$, and $d(\theta_i, \hat{\theta}_j)$ is Euclidean distance between the ith actual and jth estimated transmitter locations. When $|\hat{\theta}| = |\theta| = 1$, single transmitter, the localization error is equal to $d(\theta, \hat{\theta})$.

2. *Cardinality error:* The cardinality error, ϵ_c, is the fraction of time during which the number of estimated transmitters differs from the actual number of transmitters [63].

3. *OSPA metric:* The optimal subpattern assignment (OSPA) metric, ϵ_p, penalizes the error in the number of estimated transmitters with a constant g that is measured in meters [64]. The OSPA metric is obtained by the following formula when $|\theta| \leq |\hat{\theta}|$,

$$\epsilon_p(\theta, \hat{\theta}) = \left(\frac{1}{|\hat{\theta}|} \min_{a \in A} \sum_{i=1}^{|\theta|} d^g(\theta_i, \hat{\theta}_{a_i})^2 + g^2 \left(|\hat{\theta}| - |\theta| \right) \right)^{\frac{1}{2}}, \tag{3.7}$$

   where $d^g(\theta, \hat{\theta}) = \min(g, d(\theta, \hat{\theta}))$. When $|\theta| > |\hat{\theta}|$, the OSPA metric is obtained by inverting θ and $\hat{\theta}$ in (3.7).

3.3.3 Results

As mentioned in Section 3.2, after finding the local maxima, we perform localization in each subarea using the matrix inversion. This approach is simple and efficient in terms of time complexity. Based on our evaluation, the running time of the matrix inversion approach for a single transmitter in the open environment is 0.1 second. However, the running time of the well-known maximum likelihood estimator (MLE) [55] is around 1 second.

In this section, we first evaluate the matrix inversion approach in terms of the localization error. We compare the localization error of the matrix inversion approach with that obtained from the MLE. Table 3.2 shows the localization error in the open environment for each transmitter. As shown in this table, the matrix inversion approach performs better in terms of the localization error. The average localization error of matrix inversion

is 1.11 meters. In comparison, the average error of the MLE approach is 2.02 meters. Also, the variance of the localization error of the MLE approach is 4.1 square meters which is much higher than 0.18 square meters, the variance of the localization error of the matrix inversion approach. We also evaluate the localization error of one transmitter in the cluttered office. We find that the average and variance of localization error among all possible single transmitters are 1.6 meters and 0.61 square meters, respectively, in the matrix inversion approach. However, the average and variance of the localization error in the MLE approach are 1.77 meters and 1.55 square meters, respectively. Given the benefits of the matrix inversion approach in terms of both time complexity and the localization error, we select this approach to localize a single transmitter.

Next, we analyze the performance of SPLOT for multiple transmitters. To create the changes in number and locations of transmitters, we use the two state continuous time Markov chain and find the on and off state of each transmitter for 1000 seconds. The results are obtained by taking average of 100 different trials of transmitters in on and off states over 1000 seconds. We assume that only one transmitter is turned on at any instant, although once turned on at different time instants, multiple transmitters can be transmitting simultaneously. However, we do consider the scenarios where multiple transmitters can be simultaneously turned off.

- ***Number of transmitters:*** We consider the impact of maximum number of transmitters on the performance of SPLOT. We set the maximum number of transmitters to 1, 2, 3, and 4 in the open environment. Figure 3.4 shows the changes in the average localization error of SPLOT and Quasi EM with the maximum number of transmitters in open environment. To find the localization error of Quasi EM, we provide the actual number of transmitters as an input parameter. Also to find average localization error, we run 1000 trials of Quasi EM and take the average over these trials. The average localization errors shown in Figure 3.4 are over different combinations of transmitters in the open environment.

  Figure 3.4 shows that the average localization error increases substantially with the increase in the number of transmitters in the Quasi EM approach. However, the average localization error of SPLOT is at or less than 1 meter, even though SPLOT also estimates the number of transmitters (unlike Quasi EM, SPLOT is not provided

the number of transmitters as an input). Interestingly, we do not observe any penalty for increasing the number of transmitters being located.

Next, we compare the running time of SPLOT with that of Quasi EM in the open environment. Table 3.3 shows the running time of these two approaches for a trial of 100 seconds of the transmitters being in the on and off states. The Quasi EM is run only for one time. Table 3.3 shows that the running time of SPLOT is around 0.5 second. However, the running time of Quasi EM is above 200 seconds and the increase in the maximum number of transmitters substantially degrades the running time of Quasi EM degrades. We also expect the running time of Quasi EM to degrade more with increasing number of mobile sensing devices.

We also evaluate SPLOT in terms of average cardinality error and average OSPA error. Table 3.4 shows the average OSPA error and the average cardinality error of SPLOT with increasing maximum number of transmitters. This table shows that an increase in the maximum number of transmitters increases the average cardinality error. The average cardinality error in the worst case is around 0.14 when the maximum number of transmitters is 4. The OSPA metric considers both the localization error and the cardinality error in one metric. Table 3.4 shows that the average OSPA error increases by a very small amount even when the cardinality penalty is set to a very high value ($g = 5m$). This is because the fraction of times that the number of estimated transmitters $|\hat{\theta}|$ and the number of actual transmitters $|\theta|$ differ is very small.

- ***Impact of transmitters locations:*** Next, we analyse the effect of transmitter locations on the localization approach. To find the impact of transmitter locations, we use the cluttered office data with the maximum number of transmitters equal to 2. We consider different combinations of two transmitters such that the Euclidean distance between two transmitters varies from 3.5 meters to 18 meters. Figure 3.5 and Figure 3.6 show the relationship between the transmitters' Euclidean distance and the average localization error. Here, each data point is obtained by averaging over 100 different trials, each with its own randomly generated transmitter on and off chains. Also, note the different scales on the y-axes of the two plots. Figure 3.6 shows a linear correlation between the transmitters' distance and the localization error in the Quasi

EM approach. However, in SPLOT, there is no apparent correlation between the distance between transmitters and the localization error. The correlation coefficients between distance and average localization error in the Quasi EM and SPLOT are 0.72 and -0.02, respectively.

Regarding the cardinality error, there is a small correlation between the distance between transmitters and the cardinality error in SPLOT. Our evaluations show that the correlation coefficient of distance and average cardinality error in SPLOT is around -0.35. With increasing distance, SPLOT is more successful in finding the local maxima and converting the multiple transmitters localization to a set of single transmitter localizations. Therefore, the average cardinality error decreases with increasing distance between transmitters. Similarly, there is also a small correlation of -0.1 between the distance and the OSPA error.

- ***Comparison of SPLOT and Quasi EM in the cluttered office:*** With the help of our experimental results, we have shown, earlier in this section, that SPLOT performs better than Quasi EM in the open environment. However, the number of transmitters and mobile sensing devices in the open environment is small and there is no significant noise or obstruction in the environment. In this section, we compare SPLOT with the Quasi EM in the cluttered office environment, where we have 44 nodes that are located uniformly in the cluttered office and we can select any of these nodes as a transmitter or a mobile sensing device. Figure 3.7 shows the CDF of average localization error in the cluttered office for different combination of transmitters with maximum number of two transmitters. The average localization error on the x-axis corresponds to the localization error for each combination of transmitters obtained by averaging over 100 different trails. Figure 3.7 shows that the average localization error of SPLOT is significantly less than that of Quasi EM. Furthermore, the average localization error for any combination of transmitters is 4.54 meters in Quasi EM which is much higher than the average localization error in SPLOT.

3.4 Implementation

To further investigate its accuracy, we use commodity devices to implement SPLOT. Our mobile sensing device consists of a commodity smartphone/tablet that connects to

an inexpensive Realtek dongle (RTL-SDR for brevity) [65] via a USB cable. The RTL-SDR acts as a sensing unit and collects raw In-phase/Quadrature(I/Q) samples while the smartphone/tablet acts as a computing unit that receives I/Q samples from RTL-SDR and computes the RSS value. RTL-SDR operates in 25MHZ-1750MHZ with a sample rate up to 2.4MHZ. For transmitters, we use BaoFeng BF-F8HP (BF) [66]. BF is an FM transmitter that is able to transmit VHF in 136MHZ-174MHZ and UHF in 400MHZ-520MHZ with up to 8W power. We build an Android app that is able to measure spectrum in real time by specifying the frequency range and the sampling rate. The app records the I/Q samples obtained from the RTL-SDR and computes the RSS value. In our setup, the app generates a (time, location, RSS) tuple every second. The location is the GPS coordinates for outdoor experiments. For indoor experiments, the app finds the location by indoor fingerprinting localization [67].

3.4.1 Data Gathering

We have 30 users participate in our experiments for carrying both the transmitters and mobile sensing devices. They are of different ages, body shapes, and heights. Each user has its own Android device (smaprtphone/tablet) with our app installed and an attached RTL-SDR device. We collect data in different indoor and outdoor areas with at least two transmitters. The areas of our experiments are at least 30 m by 30 m^2 in size.

To determine the location and the transmission time of transmitters, a user that carries the transmitter also carries a mobile sensing device (smartphone/tablet connected to RTL-SDR). Our app on the smartphone/tablet records the transmission time and the location of transmitter every second. In some experiments, the transmitters are transmitting continuously, while in the other experiments, we give the transmitters a transcript for transmission. The transcript shows the time of transmission for each transmitter. We create the transcript to allow us to experiment with a different number of active transmitters at different times. In all experiments, we configure the transmit power to 1W and the frequency band to 443MHZ.

---

[2]We make the experiment area small to evaluate SPLOT in an environment where mobile sensing devices are receiving signal from both transmitters and there is strong interference.

3.4.2 Test Environment

We perform different indoor, outdoor experiments for both stationary and mobile scenarios.

- ***Engineering Building:*** We perform two experiments on the third floor of an engineering building. In both experiments, there are at most 8 mobile sensing devices[3] that are placed in four corridors of a square area of 40 m by 40 m. Two transmitters that are located in two opposite corridors. In the first experiment (Experiment A), the mobile sensing devices are static while the transmitters transmit continuously and move along the corridors for 7 minutes at normal walking speed. In the second experiment (Experiment B), the mobile sensing devices move randomly in different corridors. This experiment takes 8 minutes and the transmitters use a transcript for transmission.
- ***Food Court:*** We perform two experiments in a university food court area. The food court is an indoor area of 30 m by 50 m. In both experiments, there are 6 mobile sensing devices that are located uniformly along the food court and both the transmitters and the mobile sensing devices are static. Also, in both experiments the transmitters use a transcript for transmission. In the first experiment (Experiment C), there are two transmitters that are located on two ends of the food court at first. Then, we gradually reduce the distance between the transmitters. In the second experiment (Experiment D), there are three transmitters located in three different corners of the food court.
- ***Outdoor Area:*** We perform two experiments in an outdoor area of size 30 m by 50 m that is a part of a campus where both static (buildings, trees) and mobile obstacles (pedestrians) are present during the experiment. In both experiments there are at most 8 mobile sensing devices. In the first experiment (Experiment E), both transmitters and mobile sensing devices are static and the transmitters use a transcript for transmission. In the second experiment (Experiment F), both transmitters and mobile sensing devices are moving. The transmitters transmit continuously and move

[3]Users are not able to run the app for the duration of the experiment time for different reasons such as a battery issue, etc.

around a circle for 7 minutes at normal walking speed. The mobile sensing devices also move around the circle while maintaining a distance from the transmitters.

3.4.3 Results

We evaluate SPLOT using the data from our implementation and compare its performance with that of Quasi EM. Figure 3.8 shows the average localization error of SPLOT and Quasi EM in experiments A to F. Figure 3.8 shows that the average localization error of SPLOT is substantially less than the Quasi EM. The average localization error decreases in both SPLOT and Quasi EM when both transmitters and the mobile sensing devices are static (Experiment C, D, E). The average localization error of Quasi EM increases significantly when transmitters and mobile sensing devices are mobile (Experiment A, B, F). In comparison, the average localization error of SPLOT is less than 5 meters.

We also evaluate SPLOT in terms of average cardinality error and average OSPA error for experiments A to F. Table 3.5 shows the average OSPA error and the average cardinality error of SPLOT. This table shows that the average cardinality error increases when both transmitters and mobile sensing devices are mobile. Also, the average cardinality error in Experiment D is greater than that in Experiment C and E because of an increase in the number of transmitters. Table 3.5 also shows that the average OSPA error increases by a very small amount even with high cardinality penalty ($g = 5m$).

Figure 3.9 shows the changes in the average localization error of SPLOT and Quasi EM for different number of transmitters in Experiment D. Figure 3.9 shows that the average localization error increases with the increase in the number of transmitters in the Quasi EM approach. However, the average localization error of SPLOT is at or less than 4 meters, even though SPLOT also estimates the number of transmitters (recall that Quasi EM assumes that the number of transmitters is known).

Finally, we analyse the effect of transmitter locations on the localization approach. To find the impact of transmitter locations, we use Experiment C with number of transmitters equal to 2. We change the location of one transmitter such that the Euclidean distance between two transmitters varies from 6 meters to 45 meters. Table 3.6 shows the relationship between the transmitters' Euclidean distance and the localization error. Table 3.6 shows that there is no apparent correlation between the distance between transmitters and the

localization error in SPLOT. However, the localization error decreases in the Quasi EM with decreasing distance between transmitters. This is because Quasi EM estimates the locations of both transmitters close to one of the transmitters.

3.5 Conclusion

We presented a method for simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring (SPLOT). SPLOT considered the temporal availability and mobility of both receivers and transmitters and made no assumptions about the number of transmitters. We experimentally evaluated our approach in two different settings. We also implemented SPLOT on commodity devices and performed multitransmitter localization in a variety of indoor and outdoor experiments. Our results demonstrated the efficiency and accuracy of our approach.

In the next chapter, we build sampling approaches to select mobile sensing devices required for localization.

Table 3.1. Evaluation parameters.

Parameters	Values	Descriptions
δ_p	1.5	Pixel width(m)
σ_x^2	0.5	Voxels variance(dB)
σ_c	1	Correlation coefficient
$minPL$	1.5	Minimum path length(m)
n_p	2	Path loss exponent
λ	1	Parameter of exponential distribution in Markov chain model
b	2	Bursts of transmitting
p	0.5	Probability of transmitting
R	4	Confined area radius(m)
k	20	Number of iterations in Metropolis

Table 3.2. Localization error in Matrix inversion and MLE approaches for one transmitter in the open environment with no mobility.

Transmitter	Localization error(m)	
	Matrix inversion	MLE approach
TX1	1.41	1.01
TX2	0.5	1.03
TX3	1.11	0.97
TX4	1.41	5.08

Table 3.3. Running time of SPLOT and Quasi EM in the open environment.

Maximum number of transmitters	Running time (second)	
	SPLOT	Quasi EM
1	0.5	211
2	0.5	985
3	0.7	2281
4	0.6	4169

Table 3.4. Average OSPA error, $\bar{\epsilon_p}(m)$, and average cardinality error, $\bar{\epsilon_c}$, of SPLOT in the open environment.

Maximum number of transmitters	$\bar{\epsilon_p}(m)$			$\bar{\epsilon_c}$
	$g=1$	$g=2$	$g=5$	
1	0.79	1.01	1.01	0
2	0.91	1.17	1.17	0
3	1.04	1.36	1.45	0.05
4	1.14	1.51	1.74	0.14

Table 3.5. Average OSPA error, $\bar{\epsilon_p}(m)$, $g = 5m$, and average cardinality error $\bar{\epsilon_c}$, of SPLOT for experiments A to F.

Experiment	A	B	C	D	E	F
$\bar{\epsilon_p}(m)$	5.39	5.05	4.19	5.03	3.79	6.38
$\bar{\epsilon_c}$	0.18	0.11	0.04	0.16	0	0.06

Table 3.6. Average localization error, $\bar{\epsilon}_l(m)$, of SPLOT and Quasi EM versus distance between two transmitters in experiment C.

	Distance (m)	45	41	36	29	18	9	6
$\bar{\epsilon}_l(m)$	SPLOT	3.38	3.82	3.67	4.92	2.9	3.68	1.16
	Quasi EM	18.72	17.34	14.43	11.31	8.7	4.96	5.66

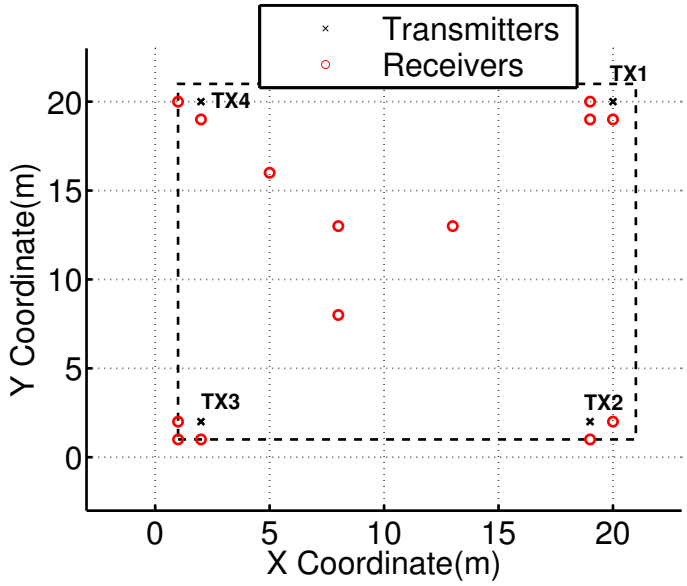

Figure 3.1. The experiment layout for an open environment.

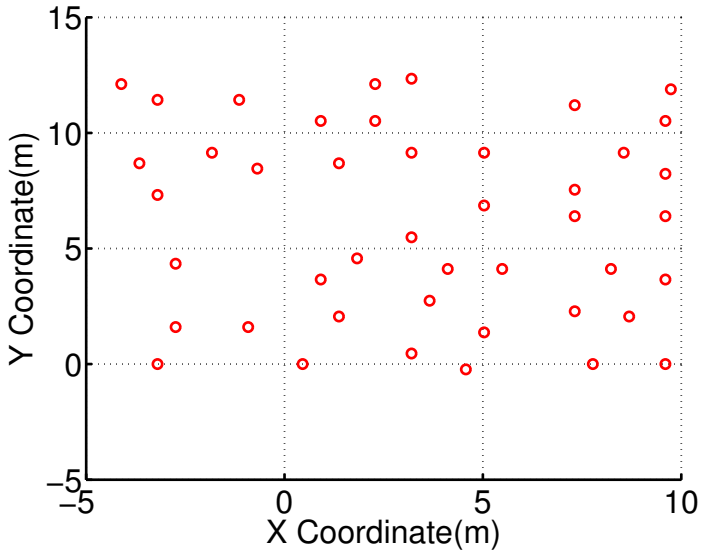

Figure 3.2. The experiment layout for a cluttered office.

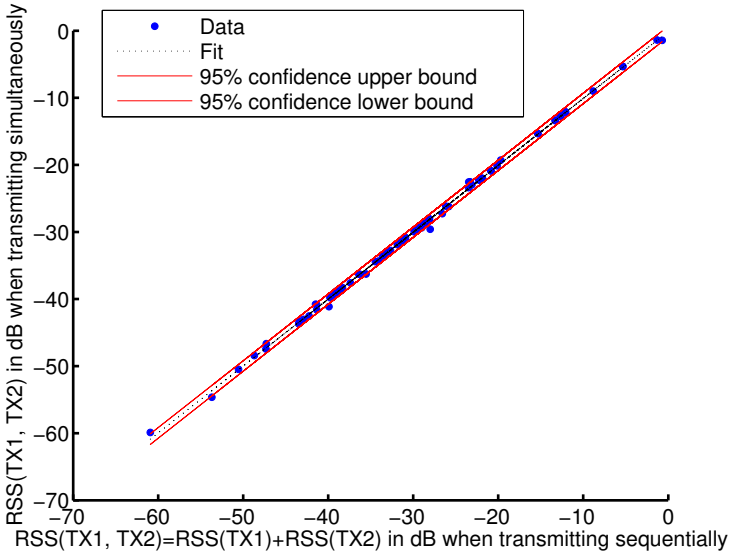

Figure 3.3. Correlation between the RSSs received by receivers when transmitting simultaneously and when transmitting sequentially in the Orbit testbed.

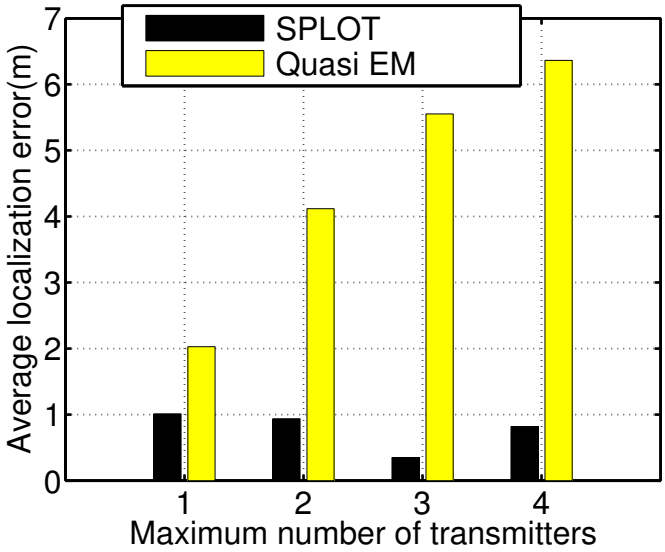

Figure 3.4. Average localization error versus the maximum number of transmitters in the open environment for Quasi EM and SPLOT.

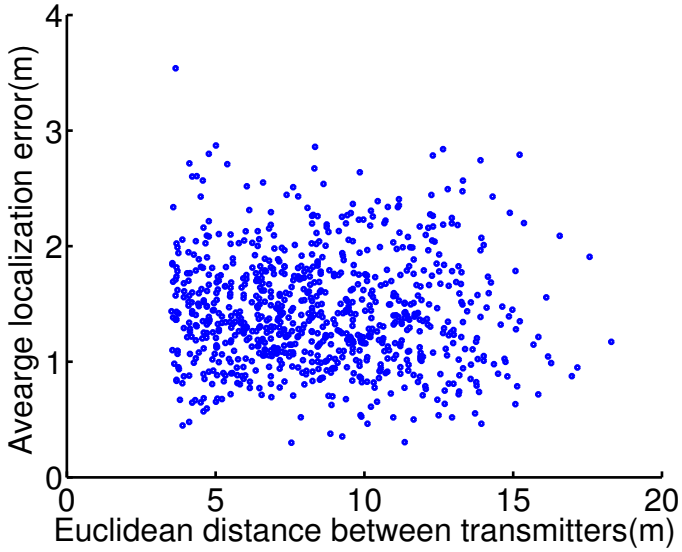

Figure 3.5. Impact of transmitters locations in SPLOT for cluttered office.

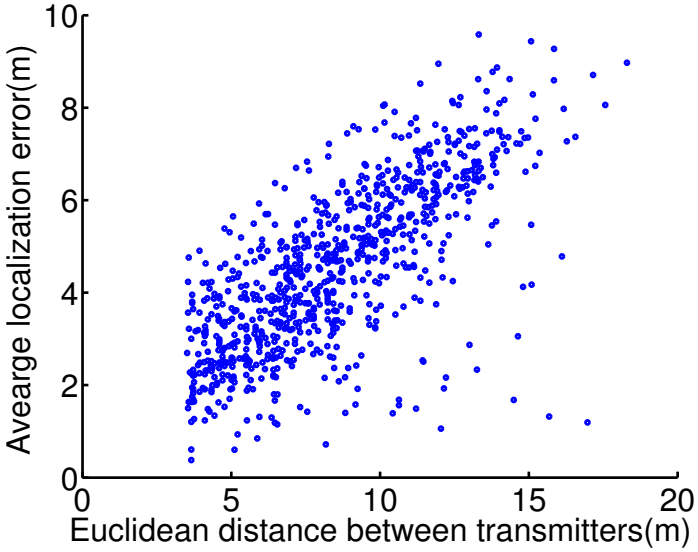

Figure 3.6. Impact of transmitters locations in Quasi EM for cluttered office.

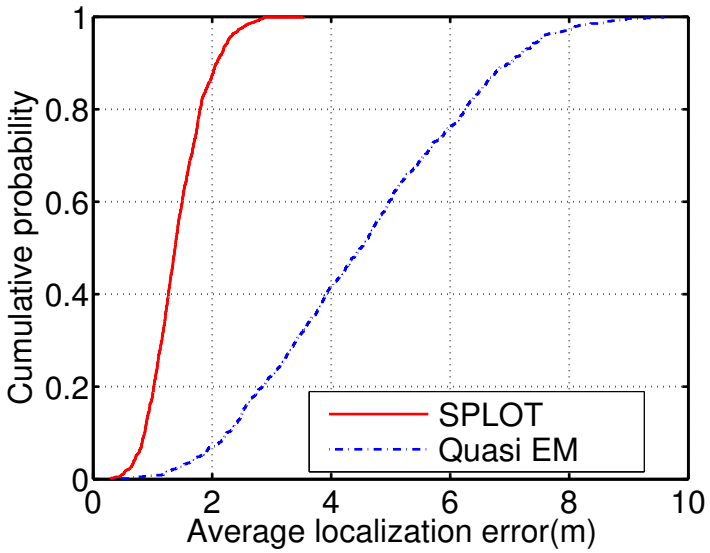

Figure 3.7. CDF of average localization error (m) in the cluttered office with maximum two transmitters.

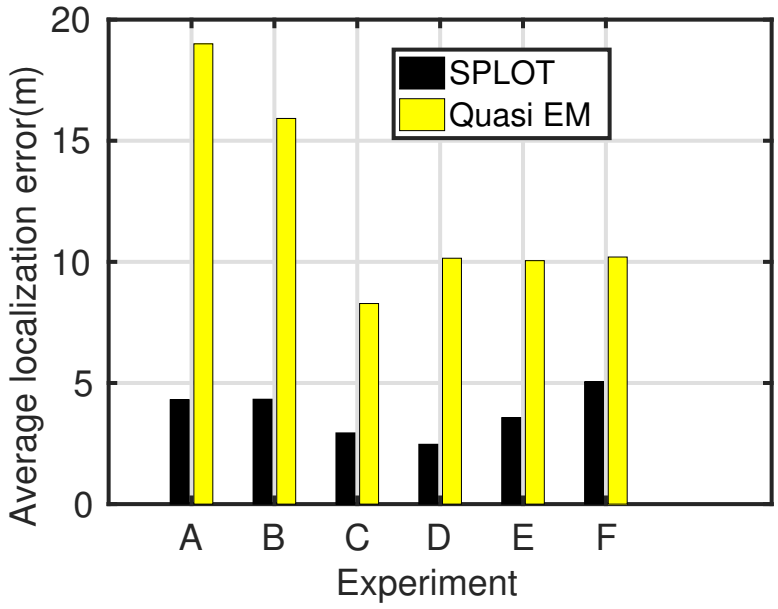

Figure 3.8. Average localization error of SPLOT and Quasi EM in experiments A to F.

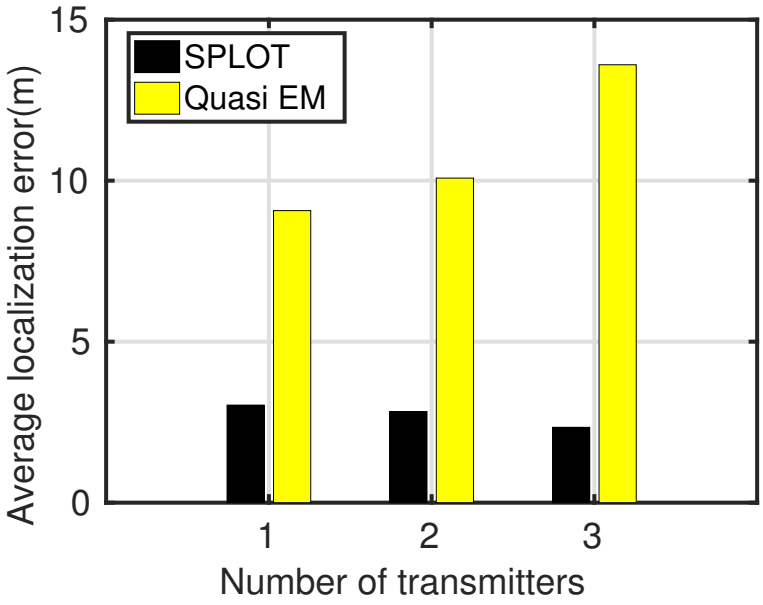

Figure 3.9. Average error versus the number of transmitters in experiment D for SPLOT and Quasi EM.

CHAPTER 4

SAMPLING FOR CROWDSOURCED SPECTRUM MONITORING

4.1 Introduction

In this chapter, we build efficient sampling approaches for crowdsourced spectrum monitoring. An important aspect of offender localization is the task of selecting a set of mobile sensing devices for RSS measurements. The selection approach must consider the following: (i) maximum spatial coverage of the monitored area, and (ii) computational efficiency, i.e., select the nodes in a timely manner.

We define and use a new metric called *degree expansion*, which represents the amount of overlap in the sensing ranges of mobile sensing devices. A high degree expansion amounts to selecting receivers from new uncovered geographic areas and thus maximizing coverage. Using this metric, we propose two sampling approaches: 1) Greedy sampling and 2) Metropolis sampling. We also make our sampling efficient for handling mobility by using the information from the previous sampling interval in the current sampling interval and providing an adaptive approach to determine the number of required sensing devices in each sampling interval. While good samaritans can be recruited for monitoring spectrum, mobile users need not participate in crowdsourcing for selfish reasons (including depletion of batteries and use of their processing resources) unless they receive some payoff as a compensation. We enhance our sampling approach to incentivize mobile users such that we select nodes that maximize coverage but minimize the total payoff. Furthermore, our incentive mechanisms motivate mobile sensing devices to act truthfully. Our *truthful sampling* considers both the budget limit and mobility of mobile sensing devices.

There are a great number of works that have focused on selecting a set of sensor nodes to provide the maximum coverage in wireless sensor networks (WSNs) (e.g., [5–7]). There are also a few existing works that consider both incentive and the coverage problem [8–10].

However, unlike our work, none of these existing works considers mobility, truthfulness, and coverage problem all together. [11] tries to maximize the expected coverage for long-term participation that does not apply to mobile environments where mobile sensing devices can be too temporary.

We evaluate the impact of sampling on the localization accuracy and show how we can maintain the localization accuracy by using a suitable sampling approach to select mobile sensing devices among all available mobile sensing devices. We also compare our sampling approaches in terms of coverage and efficiency of implementation with existing well known sensor sampling approaches.

The rest of this chapter is organized as follows. Section 4.2 provides some preliminaries. Section 4.3 and Section 4.4 describe the Greedy and the Metropolis sampling approaches. In Section 4.5, we make our sampling approaches efficient for handling mobility. Section 4.7 present our truthful sampling approaches. In Section 4.6, we describe our evaluation results. The concluding remarks are provided in Section 4.8.

4.2 Preliminaries

We assume that the mobile sensing devices know their locations. Let T be the time interval after which the sampling is repeated. At the beginning of each time interval T, the central controller sends the sensing request for that time interval to the mobile sensing devices. Then, mobile sensing devices voluntarily inform the central controller about their current locations and their sensing ranges. The sensing range of a mobile sensing device is the maximum radius around their current location for a given transmit power that they are able to measure. Given the current locations of mobile sensing devices, the central controller constructs a weighted graph $G = (V, E, W)$ where $v \in V$ represents mobile sensing devices (nodes), $e_{i,j} \in E$ denotes a link between node i and node j, and $w_{i,j} \in W$ represents the weight of link $e_{i,j}$. There is a link between node i and node j, $e_{i,j} \in E$, if their sensing ranges overlap. $w_{i,j} \in [0, 1]$ shows the amount of overlap between the sensing range of node i and node j and is obtained from the following formula:

$$w_{i,j} = \begin{cases} \frac{(r_i + r_j - d_{i,j})}{(r_i + r_j)} & d_{i,j} \leq r_i + r_j \\ 0 & otherwise \end{cases} \tag{4.1}$$

Here, r_i and r_j are parameters related to the sensing ranges of node i and node j respectively. $d_{i,j}$ denotes the Euclidean distance between two nodes.

We define the following:

Definition1. S is set of nodes where $S \subset V$.

Definition2. $N(S)$ is the *neighborhood* of S that includes all nodes in $V - S$ that have a link to S.

Definition3. $d_G(v)$ is *degree* of node v in graph G. $d_G(v)$ is equal to sum of the weights of links that connect to v.

Definition4. The degree expansion, $DX(S)$, of a set S is $DX(S) = \sum_{i \in N(S)} \min_{j \in S} w_{i,j}$.

To select mobile sensing devices that maximize the coverage, we develop algorithms for solving the optimization problem below:

$$\arg\max_{S \subset V} DX(S) \tag{4.2}$$

where, S is the cardinality constraint. This optimization problem is NP-Hard. It can be reduced to the *set cover* problem [68]. In the following sections, we propose two algorithms to obtain approximate solutions.

4.3 Greedy Algorithm

The Greedy approach is shown in Algorithm 1. As we can see, the selection is based on a Greedy heuristic that selects the mobile sensing devices based on their marginal contributions to the degree expansion. The mobile sensing device with maximum degree expansion is selected first. Then, the mobile sensing devices are selected one by one iteratively based on their marginal contributions on degree expansion until S mobile sensing devices are selected. By iteration j, $j-1$ mobile sensing devices are selected (denoted by A_{j-1}), and the marginal contribution of mobile sensing device $i \in V - A_{j-1}$, if represented by $m_{i|A_{j-1}}$ is equal to $d_G(i) - \sum_{j \in N(A_{j-1}) \cup A_{j-1}} w_{i,j} - \sum_{j \in A_{j-1}} w_{i,j}$. The mobile sensing device with maximum marginal contribution to the degree expansion is selected as the winner of the j iteration. I.e., $v \in argmax_{i \in V - A_{j-1}} m_{i|A_{j-1}}$. To simplify the notation, we replace $m_{i|A_{j-1}}$ by m_i in the rest of this chapter.

Algorithm 1: Greedy Algorithm

input : $G = (V, E, W)$
S // cardinality constraint
output: A // selected mobile sensing devices

$A = \emptyset$;
$v = argmax_{v \in G} d_G(v)$;// the mobile sensing device with maximum degree is selected first
$A = A \cup v$;
while $|A| < S$ **do**
  Select new node $v \in V - A$ where
  $v \in argmax_{i \in V - A_{j-1}} m_{i|A_{j-1}}$
  $A = A \cup v$;
end

4.4 Metropolis Algorithm

Metropolis is a Markov Chain Monte Carlo (MCMC) method to sample and evaluate probability distributions [69]. Recently, Metropolis sampling has been applied to subgraph sampling [70]. In this section, we use the idea of subgraph sampling to approximate the optimization problem of Equation 4.2.

Overview: Given a graph $G = (V, E, W)$, the idea of Metropolis algorithm is to create a subgraph of size $S < |V|$ in each iteration. The first set of mobile sensing devices is selected randomly and the subsequent sets of mobile sensing devices are constructed by removing a node from and adding a new node to the subgraph. We choose a *quality measure*, described below, to quantify the degree expansion of samples. The acceptance or rejection of new mobile sensing devices is based on this quality measure. By selecting mobile sensing devices until convergence is achieved while keeping the selected mobile sensing devices with the maximum quality measure, we obtain a subgraph (selection of S mobile sensing devices out of N users) that approximately optimizes the degree expansion. We describe the pseudo-code of Metropolis in Algorithm 2.

Quality metric: Given a selected mobile sensing device A, the maximum possible degree expansion of graph $G = (V, E, W)$ is $\sum_{v \in V - A} d_G(v)$. This implies that the selected nodes, A, have links to all the other nodes, $V - A$ (i.e., $N(A) = V - A$). As a result, a normalized quality measure is equal to: $q(A) = \frac{DX(A)}{\sum_{v \in V - A} d_G(v)}$. The quality measure determines the degree expansion. A higher quality measure corresponds to a higher degree expansion.

Algorithm 2: Metropolis Algorithm

input : $G = (V, E, W)$
S, M // cardinality constraint and number of iterations
output: A_{opt} // selected mobile sensing devices

// Initial sample, S nodes selected randomly.
$A_{current} = random(V, S)$;
$A_{opt} = A_{current}$;
for $i = 1$ *to* M **do**
  $v = rand(A_{current}, 1)$;
  $w = rand(V - A_{current}, 1)$;
  $A_{new} = \{A_{current} - v\} \cup \{w\}$;
  $\alpha \in rand[0, 1]$;
  if $\alpha < \frac{q(A_{new})}{q(A_{current})}$ **then**
    $A_{current} = A_{new}$;
    if $q(A_{current}) > q(A_{opt})$ **then**
      $A_{opt} = A_{current}$;
    end
  end
end

4.5 Mobility Aware Sampling

Mobile sensing devices can frequently change their locations making the sampling and the RSS measurements unreliable. To account for the mobility of sensing devices, we should repeat the sampling often. Let the time interval after which the sampling is repeated be T. We note that the sampling is not actually performed over the time interval T but at the beginning of each interval. To reduce the time taken for node selection across repeated sampling, we use the information from previous sampling interval in the current sampling interval as follows.

1. Instead of constructing the whole graph again, we only update the constructed graph after each time interval T.
2. We select some of the nodes for the current sampling interval, from those selected in the previous sampling interval.

Besides the above optimizations to reduce the time taken for node selection, to account for mobility and possibility of erroneous or missed measurements, we use an adaptive approach to determine S (number of selected nodes) after each time interval T. The pseudo-code of our mobility aware Sampling approach is described in Algorithm 3.

At the beginning of first time interval, we select S sensing devices using one of the sampling approaches (Greedy sampling or Metropolis sampling). S^1_{opt} represents the selected sensing devices for the first time interval. Then, at the beginning of each time interval, we perform mobility aware sampling (the value of T depends on the mobility in the environment, for environments with high mobility the value of T should be smaller than that in environments with lower mobility). In mobility aware sampling after each time interval T, the selected sensing devices report their measurements along with their current locations to the central controller. The central controller decides on sampling, how many devices and how they are locally distributed, using the reported information and also the constructed graph $G = (V, E, W)$ in the previous time interval.

How many devices? The central controller uses a Linear Increase and Linear Decrease (LILD) approach to determine the number of required sensing devices, S. The number of required sensing devices, S, increases linearly if the received measurements are *unreliable* because of mobility. The central controller determines unreliable measurements by comparing the current and previous locations of selected nodes. A measurement is deemed unreliable if the difference in locations is more than a prespecified threshold.

Which devices? After selecting the number of nodes using the LILD approach, the central controller must determine which sensing devices should be selected for measurement in the next time interval T. We make this decision based on the graph $G = (V, E, W)$ constructed in the previous time interval and the current locations of previously selected sensing devices. Let S^t_{opt} and S^{t-1}_{opt} represent the selected sensing devices in the current, t, and the previous time interval, $t-1$, respectively. Also, let $\theta^t(S^{t-1}_{opt})$ denote the locations of S^{t-1}_{opt} in the current time interval. We update the graph G using the current locations of previous nodes, $\theta^t(S^{t-1}_{opt})$. This update is only related to the positions of previous nodes, S^{t-1}_{opt}, in the graph. We use the updated graph in our localization module for selection of nodes in the current time interval, S^t_{opt}.

The graph G that we construct is expected to change little across adjacent time intervals. Therefore, the nodes selected in a time interval can still provide high coverage in the subsequent time interval. Therefore, we can select some of the nodes from the selected nodes from the previous time interval. For this purpose, we map the previous locations of selected nodes in the previous time interval, $\theta^{t-1}(S^{t-1}_{opt})$, to the current locations

Algorithm 3: Mobility Aware Sampling

input : $G = (V, E, W)$
S, T // Initial number of required samples and Time interval between two consecutive sampling
output: S_{opt} // selected nodes in each time slot T

1 Set S^1_{opt}, selected nodes for the first time interval, using (Greedy or Metropolis sampling) with $G = (V, E, W)$ and S as input parameters.
2 Set $time = 1$;
3 **while** *(1)* **do**
4 **if** *time mode* $T = 0$ **then**
5 Select the required number of nodes, S, by using LILD approach.
6 Update G for current time interval, t, using the current locations of previous nodes $\theta^t(S^{t-1}_{opt})$.
7 Map the current location and previous location of S^{t-1}_{opt} based on distance.
8 S^t_{opt} are selected from S^{t-1}_{opt} if their current locations are nearest to their mapped previous locations and their distance is less than a threshold.
9 **if** $|S^t_{opt}| <= S$ **then**
10 Select the remaining nodes using Greedy or Metropolis sampling.
11 **end**
12 **else**
13 Select S nodes from $|S^t_{opt}|$ using Metropolis sampling with subgraph $G' = (S^t_{opt}, E', W')$.
14 **end**
15 **end**
16 $time = time + 1$;
17 **end**

of these sensing devices, $\theta^t(S^{t-1}_{opt})$. This mapping is based on the distance between the current locations, $\theta^t(S^{t-1}_{opt})$, and the previous locations, $\theta^{t-1}(S^{t-1}_{opt})$, of nodes. Then, we select those nodes from the previously selected nodes whose current locations is nearest to their mapped previous locations and their distance is less than a threshold (we use a threshold of 1 meter).

After selecting some nodes from the previously selected nodes, if the number of selected nodes is less than the required nodes (the *if* statement in line 9 of Algorithm 3), we perform Greedy sampling or Metropolis sampling for selecting the rest of nodes. In Metropolis sampling, instead of selecting all required nodes randomly in the initial state, we use the current selected nodes, S^t_{opt}, and select the remaining required nodes randomly.

Thus, the initial state of Metropolis sampling is not completely random, which reduces the number of iterations required in the Metropolis sampling. When the number of selected nodes is greater than the required nodes, we prune the nodes using Metropolis sampling. However, instead of using the whole graph G, we only use the subgraph of G' that we construct from the selected nodes (see line 13 of Algorithm 3).

Recovery: It is possible that we observe consecutive linear increase in the required number of nodes S. If the number of consecutive linear increases is greater than a predefined threshold, we use the current locations of all available sensing devices to obtain a fresh graph from scratch for accurate localization.

4.6 Truthful Sampling

A naive way to provide incentives for the mobile sensing devices is to reward them such that the reward amount is greater than their declared cost of participation. The problem with this approach is that mobile sensing devices can over-report their costs and receive higher rewards. Therefore, the algorithm for selecting mobile sensing devices should also motivate them to truthfully report their costs. In this section, we enhance the Greedy algorithm to consider both cost and the degree expansion in selecting mobile sensing devices and add truthfulness.

We design a time efficient payment mechanism for the case where the sampling is not optimal. Then, we propose the budget feasible version of the Greedy algorithm. Finally, we propose a truthful mobility aware sampling approach that prevents selfish mobile sensing devices from cheating with regard to their availability for the sampling task. In truthful sampling, in addition to location and the measuring range, the mobile sensing devices also inform the central controller about their bids that can be equal or greater than their actual costs of collecting data, c_i.

Given graph $G = (V, E, W)$, and a cardinality or a budget constraint, the central controller selects winners and determines the payment, p_i, for each winner. We assume that all mobile sensing devices act rationally and selfishly, and their main goal is to maximize their own profits, not to harm others. Also, each user has a utility of $p_i - c_i > 0$ if selected and 0 otherwise.

4.6.1 Truthful Greedy Algorithm

Given the cardinality constraint S, the central controller wants to select S mobile sensing devices that maximize the degree expansion and minimize cost while considering individual rationality (provides the required incentives for selfish players to participate in the game) and incentive compatibility (truthfulness). This amounts to solving the following optimization problem.

$$\max_{S \in V} \sum_{i \in S} \frac{m_i}{c_i}$$

$$s.t.$$

$$\forall i, c_i', p_i(c_i) - c_i \geq p_i(c_i') - c_i \tag{4.3}$$

$$\forall i, p_i(c_i) - c_i \geq 0 \tag{4.4}$$

Here, Constraints 4.3 and 4.4 provide incentive compatibility and individual rationality.

Our allocation mechanism must consider both cost and the degree expansion. Our payment mechanism must ensure both incentive compatibility (IC) and individual rationality (IR). The best known payment mechanism for providing IC and IR is the well-known Vickrey-Clarke-Groves (VCG) mechanism [71]. Unfortunately, VCG fails to provide IC and IR when the allocation mechanism is not optimal. The problem of selecting a set of mobile sensing devices that maximize the total weight is equivalent to *set-cover* problem which is NP-hard. We use a Greedy algorithm to approximate the optimal solution. However, we still need a truthful payment mechanism that employs an approximated Greedy algorithm to select mobile sensing devices. Towards this goal, we rely on Myerson's characterization of truthful mechanisms [72].

Theorem 4.1. ***(Myerson 1981).*** *A mechanism with allocation mechanism A and payment mechanism P is truthful if and only if the following holds:*

- *A* ***is monotone:*** *The allocation mechanism keeps selecting the mobile sensing device i as a winner if it independently decreases its declared cost, c_i.*
- *P* ***pays winners the threshold amounts:*** *Paying each winner the maximum declared cost.*

First, we need to show the submodularity for $\frac{m_i}{c_i}$ which implies that:

$$\frac{m_1}{c_1} \geq \frac{m_2}{c_2} \geq \ldots \geq \frac{m_S}{c_S} \tag{4.5}$$

We show the submodularity by contradiction. Suppose mobile sensing device i and mobile sensing device l are selected by the central controller where $i < l$, i.e., the mobile sensing device i selected before the mobile sensing device l. Let us assume that $\frac{m_i}{c_i} < \frac{m_l}{c_l}$. According to the sampling mechanism, $[i] = argmax_{j \in V - A_{i-1}} \frac{m_j}{c_j}$. That means $\frac{m_{i|A_{i-1}}}{c_i} \geq \frac{m_{l|A_{i-1}}}{c_l}$. Given that $A_{i-1} \subset A_{l-1}$, $m_{l|A_{i-1}} \geq m_{l|A_{l-1}}$. Therefore, we have $\frac{m_{i|A_{i-1}}}{c_i} \geq \frac{m_{l|A_{i-1}}}{c_l} \geq \frac{m_{l|A_{l-1}}}{c_l}$, which is a contradiction with our assumption. As a result, Equation 4.5 is true. Next, we determine the payment using the Myerson's conditions. The key point is that we have to find the maximum value for the cost that a mobile node can declare and still win. We can find this threshold amount for each winner i by setting $V' = V - \{i\}$ and run the Greedy algorithm until i is no longer selected. Let k be the index of the last mobile sensing device where $\frac{m_k}{c_k} < \frac{m_i}{c_i}$. Then, we obtain the payment to mobile sensing device i from the following formula:

$$p_i(c_i) = max_{1 \leq j \leq k} \frac{c_j m_i^j}{m_j} \tag{4.6}$$

where m_i^j is the marginal contribution of mobile sensing device i on the degree expansion in iteration j.

Lemma 4.1. *The payment mechanism provides individual rationality.*

Proof. If a mobile sensing device is selected by the sampling mechanism, then the utility is equal to $\frac{c_j m_i^j}{m_j} - c_i$. Given that $\frac{m_i^j}{c_i} \geq \frac{m_j}{c_j}$, the utility of the winner is always positive. Also, if the mobile sensing device is not selected by the sampling mechanism, the utility of the device is zero.

Lemma 4.2. *The payment mechanism provides incentive compatibility for the declared cost.*

Proof. Let c_i, c_i' be the declared cost of mobile sensing device i when it is being truthful and not truthful, respectively. We must consider four cases here. First, if mobile sensing device i is the winner, by over-reporting, $c_i' > c_i$, or under-reporting, $c_i' < c_i$, the utility of mobile sensing device decreases. Thus, the mobile sensing device has no incentive to lie. Second, if the mobile sensing device i is a loser it is still not selected even if it over-reports or under-reports its cost. Then, the utility is still zero and once again there is no incentive to lie. Third, if mobile sensing device i is a winner, it is not selected by over-reporting its cost. Then, the utility will be zero. Thus, the mobile sensing device has no incentive to

lie. Fourth, if the mobile sensing device i is a loser and it is not selected by over-reporting its cost. Otherwise, it contradicts the submodularity. However, the mobile sensing device can be selected by under-reporting its cost. Let us assume that the mobile sensing device l is selected by the Greedy algorithm when the mobile sensing device i acts truthfully. This means $\frac{m_l}{c_l} \geq \frac{m_i}{c_i}$. Since the maximum threshold for the mobile sensing device i is at least $\frac{m_l}{c_l}$, the payment to player i is at least $\frac{m_i^l c_l}{m_l}$ meaning that the utility is at most zero.

4.6.2 Budget-feasible Truthful Greedy Algorithm

We now consider a limit, B, on the total budget available to the controller. Given this limit, we must select a set of mobile sensing devices that maximize the marginal contribution on degree expansion. We formulate the following optimization problem. To select the mobile sensing devices with the budget limit constraint, similar to [73], we check the following condition at each iteration of the Greedy algorithm, and stop the algorithm whenever the following condition no longer holds.

$$c_i \leq B \frac{m_i^j}{\sum_{j \in A_j} m_j} \tag{4.7}$$

Let $\{1, \ldots, k\}$ be the largest subset that respects condition 4.7. To find the amount of payment for mobile sensing device $i \leq k$, we remove this player from the set of mobile sensing devices, $V' = V - \{i\}$, and then run the Greedy algorithm. In each iteration of Greedy algorithm, we check the budget limit constraint, Equation 4.7. Let k' be the last iteration of the Greedy algorithm for $V' = V - \{i\}$ that Equation 4.7 holds and mobile sensing device i is still a winner, $\frac{m_{k'}}{c_{k'}} \leq \frac{m_i^{k'}}{c_i}$. To simplify the notation, we write the proportional share of mobile sensing device i in iteration j, $\rho_i(j) = B \frac{m_i^j}{\sum_{j \in A_j} m_j}$ and $c_i(j) = \frac{c_j m_i^j}{m_j}$. The payment for user i in the budget feasible mechanism is:

$$p_i = max_{j \leq k'}(min(\rho_i(j), c_i(j))) \tag{4.8}$$

Lemma 4.3. *The budget feasible mechanism provides individual rationality.*

Proof. Since the payment has the maximum value, we need to show the following for a $j \leq k'$.

1. $c_i \leq \rho_i(j)$

2. $c_i \leq c_i(j)$

The first condition is the budget limit constraint, Equation 4.7, that holds. The second condition is related to the threshold payment and shows that the mobile sensing device i is still the winner. Otherwise, the mobile sensing device i is not selected

Lemma 4.4. *The budget feasible mechanism provides incentive compatibility for the declared cost.*

Proof. To prove incentive compatibility, we need to show that p_i is the maximum cost that mobile sensing device i can declare and still be selected by the allocation mechanism. Let r be the index in $j \leq k'$ such that $min(\rho_i(j), c_i(j))$ is maximized. For the case where $\rho_i(r) < c_i(r)$, declaring higher cost puts the mobile sensing device i after first k' users. Therefore, the mobile sensing device i will not be selected by the sampling mechanism. Otherwise, if there is a j where $j \leq k'$ such that $\rho_i(j) > \rho_i(r)$, considering the maximality of r, $c_i(j) < \rho_i(r) < \rho_i(j)$. This implies that mobile sensing device i by increasing its cost gets placed after mobile sensing device j and thus, will not be selected. If $\rho_i(r) \geq c_i(r)$, a higher cost places the mobile sensing device i after r and since r is the maximum index in k', mobile sensing device i will not be selected. Otherwise, if there is a j, $j < k'$ where $c_i(j) > c_i(r)$, by maximality we have $c_i(j) > \rho_i(j)$. i will not be selected because of the budget limit constraint, Equation 4.7.

4.6.3 Mobility Aware Budget-feasible Truthful Greedy Algorithm

In this section, we consider the mobility of sensing devices in our proposed truthful sampling. We propose a truthful mobility aware sampling that prevents the selfish mobile sensing devices from lying about their availability for the sensing task. In this setting, when the central controller probes the nearby mobile sensing devices for sampling over the time interval T, the mobile sensing devices reply by declaring their locations, costs, and their availabilities. Based on this information, the central controller selects a set of mobile sensing devices that maximize the marginal degree contribution under the budget limit and also are available during the time interval T. By providing incentives for the mobile sensing devices to declare the availability truthfully, the controller only selects among those devices that are available over the time interval T.

The sampling approach in the mobility-aware truthful mechanism is the same as one in

the previous subsection. However, in the mobility aware approach, the central controller first selects the mobile sensing devices that are available during the time interval T and then applies the Greedy budget limit on the available mobile sensing devices. The payment mechanism in this case should provide incentive for mobile sensing devices to declare their availability times truthfully. Therefore, in the mobility aware truthful mechanism, the threshold payment depends on cost and the availability time of mobile sensing devices. Let t be number of time intervals T that mobile sensing device i participates in the sampling, then the payment to mobile sensing device i is the maximum of threshold payment for each time slot T. Formally,

$$p_i = max_t(p_i(t)) \tag{4.9}$$

The above payment ensures the truthfulness of mobile arrival and departure times by removing the time dependency. Let a_i, d_i denote the arrival and departure times of device i, respectively. Let a_i', d_i' be the declared arrival and departure time for the mobile sensing device i. Considering the fact that the mobile sensing device cannot declare early arrival and late departure, $[a_i', d_i'] \in [a_i, d_i]$. Thus, the mobile sensing device i can only decrease the value of t. Since we take the maximum among all threshold payments, the payment to mobile sensing device i when it lies about the availability time is no greater than that when it reports truthfully.

4.6.4 Time Complexity

Now, we show that truthful sampling can be done in polynomial time. For the Greedy algorithm, the time complexity is $\mathcal{O}(S \log N)$. Recall that, S, N denote the cardinality constraints and the number of available mobile sensing devices, respectively. For the Greedy truthful algorithm, the sampling time complexity is $\mathcal{O}(S \log N)$. We also need to run the sampling mechanism S more times for payment determination. Thus, the time complexity is $\mathcal{O}(S^2 \log N)$. This time complexity is the same for the budget-feasible algorithm. Finally, for the mobility-aware algorithm, the payment is the maximum threshold payment for each time slot. Therefore, the overall time complexity is $\mathcal{O}(tS^2 \log N)$ where t is number of time intervals T that mobile sensing devices are available.

4.7 Evaluation

In this section, we evaluate the proposed sampling approaches. First, we study the impact of the Greedy, the Metropolis sampling, and mobility aware sampling approaches on the localization accuracy. Next, we compare our sampling approaches in terms of coverage and efficiency of implementation with existing well known sensor sampling approaches. Finally, we evaluate the impact of cost on the Greedy truthful sampling.

4.7.1 Impact of Sampling on Localization Accuracy

The number of available mobile sensing devices plays a key role in the accuracy of localization. Selecting a large number of mobile sensing devices for measurement increases the communication overhead between the mobile sensing devices and the localization module. It also increases the time taken and the energy consumption. On the other hand, selecting a very small number of mobile sensing devices decreases the accuracy of our localization approach in terms of the localization error, the cardinality error, and the OSPA error. In this section, we examine the impact of the number of mobile sensing devices on efficiency of our localization approach and show how we can maintain the localization accuracy by using a suitable sampling approach to select mobile sensing devices among all available mobile sensing devices. We also evaluate the impact of the mobility aware sampling approach on the localization. For evaluation, we use the data from the open environment and the cluttered office that we described in Section 3.3.1.[1] To create mobility for both transmitters (offending sources) and receivers (or mobile sensing devices) in the open environment and the cluttered office, we change the number and locations of transmitters and receivers using a two state continuous time Markov chain and find the on and off state of each transmitter and receiver for 1000 seconds. The results in this section are obtained by taking average of 100 different trials of transmitters and receivers in on and off states over 1000 seconds. The use of this Markov chain to activate or deactivate transmitters and receivers is similar to its use in Chapter 3, as described in Section 3.3.1.

Table 4.1 shows the average localization error and the average cardinality error when the maximum number of transmitters is 3 in the open environment with the number of

[1] We are not able to evaluate our sampling approaches using the implementation data (Section 3.4) because of the small number of mobile sensing devices.

mobile sensing devices reduced from 14 to 12, 9, and 6 for three sampling approaches for selecting mobile sensing devices (random, Greedy, and Metropolis). We make the following observations. First, the average localization error changes by about 0.25 meters when reducing the number mobile sensing devices from 12 to 6 for all sampling approaches and the average localization error is close to 1 meter even for 6 mobile sensing devices (see Table 4.1). Table 4.1 also shows that the average localization error does not change too much by selecting mobile sensing devices randomly. This table also shows that the average cardinality error increases with reduction in the number of mobile sensing devices. This is possibly because the number of selected mobile sensing devices is not enough to cover the whole area and the localization approach is not able to detect the transmitters located in the uncovered area. Moreover, the results of random sampling in Table 4.1 show that the location of mobile sensing devices directly effects the cardinality error. In other words, we need to have enough mobile sensing devices to cover the whole area to be able to detect all available transmitters. Third, the results of average cardinality error for Greedy sampling show that by reducing the number of mobile sensing devices and selecting a good sampling approach, we can still have a small cardinality error. Finally, this table also show that for the open environment where nodes are distributed nonuniformly, the Greedy sampling performs better than the Metropolis sampling approach.

Next, we evaluate the impact of number of mobile sensing devices in the cluttered office. The maximum number of transmitters is 2 and we vary the number of mobile sensing devices between 5 to 40 with increments of 5. Table 4.2 shows average localization and cardinality errors in the cluttered office. We make the following observations. First, the average localization error varies from 1.42 to 2.24 meters. However, the average cardinality error increases relatively substantially with reduction in the number of mobile sensing devices. Second, for the cluttered office area where the mobile sensing devices are located uniformly in the environment, Metropolis sampling performs slightly better than Greedy sampling. Third, Table 4.2 shows that by reducing the number of mobile sensing devices from 42 to 25 and using either of Metropolis or Greedy sampling approaches, we can keep the localization error close to 2 meters and the cardinality error to about 0.06. Fourth, the results of Table 4.2 show that 5 mobile sensing devices is not enough to cover the whole area in the cluttered office environment. As this table shows, even by selecting

the Metropolis sampling the average cardinality error is 0.43 if we only select 5 mobile sensing devices. As we discussed earlier, the determination of the number of mobile sensing devices required is done by the localization module. Generally, there is always a trade off between the accuracy of of the localization algorithm and the communication and processing overheads. Depending on the importance of each of these parameters and the localization results, the localization module makes a decision about the number of required mobile sensing devices.

Next, we evaluate the proposed mobility aware sampling approach. We use three approaches for selecting receivers in this mobile environment. In the base approach, we assume that the localization module probes the environment every second and therefore, all selected receivers report their measurements to the localization module every second. Probing the environment every second has high communication and processing overhead. As more efficient alternatives, we consider probing the environment every 20 seconds and use mobility aware sampling (mobility aware Greedy or mobility aware Metropolis) to select the required receivers in each time slot ($T = 20$ seconds) with initial sample size $S = 20$. However, in terms of localization accuracy, the base approach may perform better than the two more efficient alternatives. In this section, we evaluate the localization efficiency of the base approach and the two alternatives.

Figure 4.1 compares the localization accuracy of the base approach, and the mobility aware Greedy and the mobility aware Metropolis alternatives in the cluttered office with maximum two transmitters. Figure 4.1 shows the CDF of average localization error for three approaches. It is clear that the base approach performs better than the other approaches. However, the difference between the base and other two approaches is not very high. For instance, the 80th percentile average localization error of the other two approaches is about 0.5 meters.

Figure 4.2 shows the CDFs of average cardinality error for the three approaches. Here too, we can see a pattern similar to the CDFs of average localization error. Thus, by using mobility aware sampling, we are able to keep the localization accuracy close to the base approach and reduce the overhead[2] of probing the environment every second. Table 4.3

[2]We do not evaluate the exact overhead.

shows average value of localization error, $\bar{\epsilon_l}(m)$ and average cardinality error, $\bar{\epsilon_c}(m)$, for different combinations of transmitters in the cluttered office. The results of this table are consistent with Figure 4.1 and Figure 4.2. As Table 4.3 shows, by using mobility aware sampling where we reduce the number of selected receivers and do the probing every $T = 20$ seconds instead of every second, we can still keep the localization and cardinality error small.

4.7.2 Efficiency of the Proposed Sampling

We compare our proposed Greedy sampling in terms of coverage and efficiency of implementation with Greedy Distance and Greedy Coverage approaches [74]. In Greedy Distance, we select a sensing device that is furthest away from the current locations of the selected devices. However, in Greedy Area, we select a sensing device that covers the most area not already covered by the selected devices. To find the overlapping areas in Greedy Area, we divide the area into grids, a square of $0.2(m) \times 2(m)$. The coverage area is equal to the number of grids that are covered by the sensing range of selected devices.

For evaluation, we consider both uniform and nonuniform placements of sensing devices in a square area. To create nonuniform placement of sensing devices, we use a simple growth model that starts in an initial state, e.g., having one sensing device in three clusters. Then, every step we add a new node to one of the clusters. The probability of selecting of a cluster for a node is as follows.

$$p_i = \frac{S_i}{\sum_k S_k} \tag{4.10}$$

where p_i is the probability that cluster i is selected, S_i is the size (number of nodes) of cluster i and k is number of clusters. The above equation is called the *preferential attachment rule* which means a node is more interested in bigger clusters. The way that nodes choose their clusters results in power law distribution for cluster sizes. In other words, there will be a few large-sized clusters. This, in fact, reflects the notion of hot spots that many people visit in their daily movements.

We vary four parameters, the number of selected nodes, S, the number of available nodes, the area size, and the number of clusters for nonuniform placement. The sensing range of each node is selected randomly from a uniform distribution between 10 to 40 meters and the number of clusters for nonuniform distribution is set to 20. We use two

metrics to evaluate our sampling approaches, the coverage ratio, and the running time. The coverage ratio is equal to number of grids that are covered by the selected nodes divided by the total number of grids. All results are obtained by taking average of 100 different runs.

Figure 4.3, Figure 4.5, Figure 4.7, and Figure 4.9 show the coverage ratios of three sampling approaches (Greedy, Greedy Distance, and Greedy Area) for different values of S, the number of available nodes, the area size, and the number of clusters using both uniform and nonuniform distributions for sensing devices' placement. We make the following observations. First, as the number of selected nodes (S) increases, the coverage ratio increases in all sampling approaches. By increasing the number of available nodes and the size of area, the coverage ratio decreases in all sampling approaches. Second, Greedy Area has the maximum coverage. Our proposed Greedy sampling performs close to Greedy Area and better than the Greedy distance approach.

Figure 4.4, Figure 4.6, Figure 4.8, and Figure 4.10 show the running times of three sampling approaches (Greedy, Greedy Distance, and Greedy Area) for different values of S, the number of available nodes, the area size, and the number of clusters. These figures show that the running time of Greedy sampling is slightly less than Greedy Distance and it is a lot less than the running time of Greedy Area. As we can see, the Greedy area approach increases the coverage ratio at the cost of increasing the running time. Another weakness of the Greedy area approach is its memory usage (we need to store the number of grids that are covered by the selected nodes) and as we increase the area size this memory usage increases. Note that we can decrease both the memory usage and the completion time of the Greedy area approach by increasing the size of the grid at the cost of decreasing the coverage ratio for this approach. However, our approach performs close to Greedy Area in terms of coverage ratio without compromising the time complexity.

4.7.3 Evaluation of Truthful Sampling

In truthful sampling, we consider both cost and the coverage for selecting sensing devices. In this section, we study the impact of cost on the coverage ratio. We use three different distributions for the cost, the uniform distribution $[0, 1]$, the normal distribution with $\mu = 0$ and $\sigma = 1$ and the exponential distribution with $\lambda = 1$. We consider 500

sensing devices that are nonuniformly distributed in a 1000 by 1000 meters area and the number of selected nodes (S) varies between 25 and 125. The sensing range of each node is selected randomly between 10 to 40 meters. The number of clusters and the grid size are set to 20 and 0.2 meter, respectively.

Figure 4.11 shows the coverage ratios of Greedy approach and the truthful Greedy approach with three different distributions for the cost. This figure shows that there is not much difference between the coverage ratio of the Greedy sampling and the truthful Greedy sampling. The truthful Greedy sampling with uniform distribution for the cost has the least coverage ratio. This is because the probability of selecting a higher value for the cost in the uniform distribution is greater than that in the normal and the exponential distributions. Therefore, in the Greedy truthful sampling with uniform distribution for the cost, more likely, a node with a higher degree expansion will have a high value for the cost and hence, not selected by the sampling.

4.8 Conclusion

In this chapter, we defined and used a new metric called degree expansion and proposed two sampling approaches: 1) Greedy sampling , and 2) Metropolis sampling. We also made our sampling efficient for handling mobility by using the information from the previous sampling interval in the current sampling interval and providing an adaptive approach to determine the number of required sensing devices in each sampling interval. Next, we enhanced our sampling approach to incentivize mobile users such that we selected nodes that maximize coverage but minimize the total payoff. Our incentive mechanisms considered truthfulness, the budget limit, and mobility of mobile sensing devices.

We evaluated the impact of our proposed sampling approaches on the localization accuracy and showed how we can maintain the localization accuracy by using a suitable sampling approach to select mobile sensing devices among all available mobile sensing devices. We also compared our sampling approaches in terms of coverage and efficiency of implementation with existing well known sensor sampling approaches. Our evaluations demonstrated the efficiency of our approach.

The distributed sensing for localization in wireless networks also leads to an invasion

of privacy. In the next chapter, we investigate an attack on location privacy and provide a defense mechanism for that.

Table 4.1. Average localization error (m), $\bar{\epsilon}_l(m)$, and average cardinality error, $\bar{\epsilon}_c$, of SPLOT with different numbers of mobile sensing devices and three transmitters in the open environment.

Number of	Random		Greedy		Metropolis	
mobile sensing devices	$\bar{\epsilon}_l(m)$	$\bar{\epsilon}_c$	$\bar{\epsilon}_l(m)$	$\bar{\epsilon}_c$	$\bar{\epsilon}_l(m)$	$\bar{\epsilon}_c$
14	0.35	0.05	0.35	0.05	0.35	0.05
12	0.89	0.04	0.86	0.02	0.89	0.03
9	0.95	0.14	1.03	0.05	0.97	0.09
6	1.16	0.33	1.09	0.07	1.12	0.23

Table 4.2. Average localization error (m), $\bar{\epsilon}_l(m)$, and average cardinality error, $\bar{\epsilon}_c$, with different numbers of mobile sensing devices and two transmitters in the cluttered office.

Number of	Greedy		Metropolis	
mobile sensing devices	$\bar{\epsilon}_l(m)$	$\bar{\epsilon}_c$	$\bar{\epsilon}_l(m)$	$\bar{\epsilon}_c$
42	0.35	0.05	0.35	0.05
40	0.86	0.02	0.89	0.03
35	1.03	0.05	0.97	0.09
30	0.35	0.05	0.35	0.05
25	0.86	0.02	0.89	0.03
20	1.03	0.05	0.97	0.09
15	1.09	0.07	1.12	0.23
10	1.03	0.05	0.97	0.09
5	1.09	0.07	1.12	0.23

Table 4.3. Evaluation of SPLOT with mobile transmitters and mobile sensing devices in the cluttered office with time interval $T = 20$ seconds and initial sample size $N = 20$ and maximum two transmitters.

Approach	$\bar{\epsilon}_l(m)$	$\bar{\epsilon}_c$
Base	1.66	0.03
Mobility aware Greedy	2.14	0.12
Mobility aware Metropolis	2.13	0.11

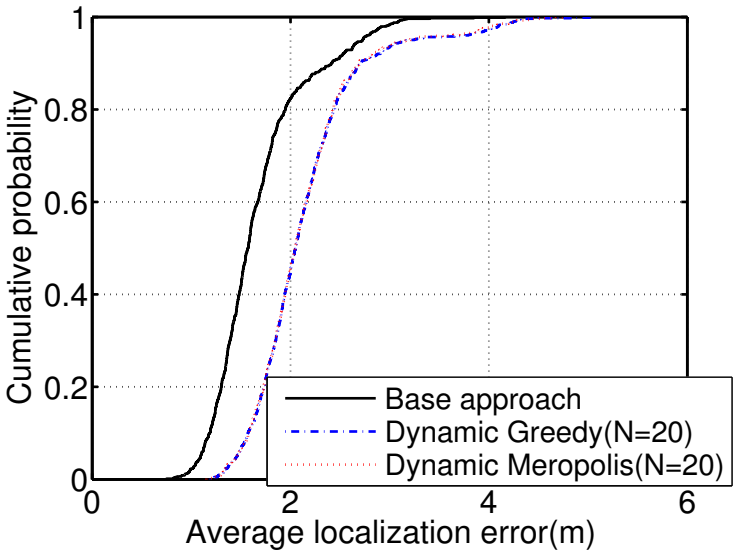

Figure 4.1. CDF of average localization error (m), with mobile transmitters and mobile sensing devices in the cluttered office and maximum two transmitters.

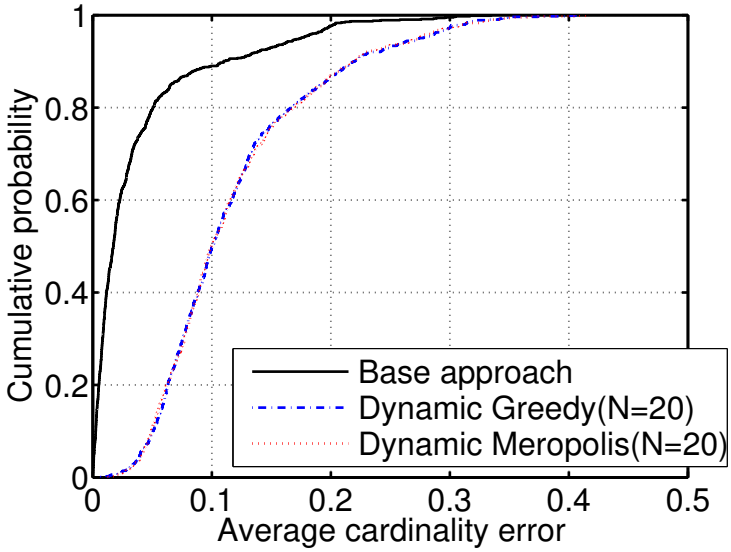

Figure 4.2. CDF of average cardinality error, with mobile transmitters and mobile sensing devices in the cluttered office and maximum two transmitters.

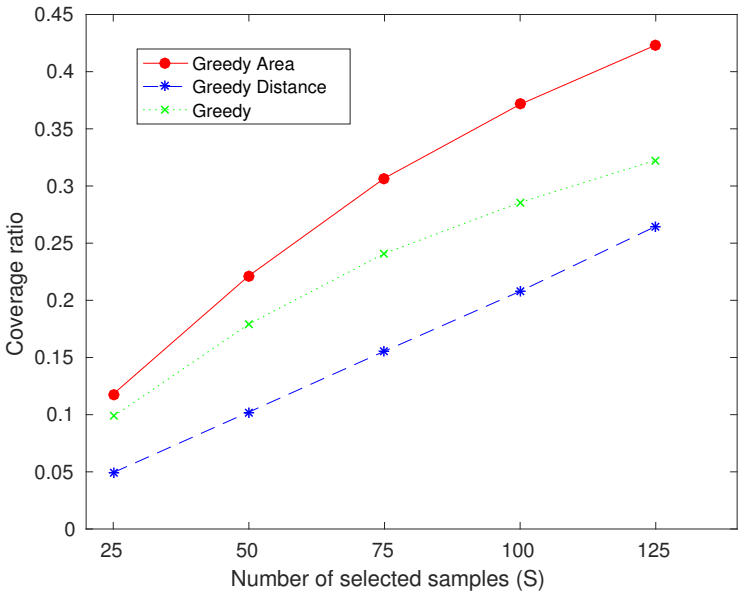

Figure 4.3. The coverage ratio versus the number of selected nodes (S).

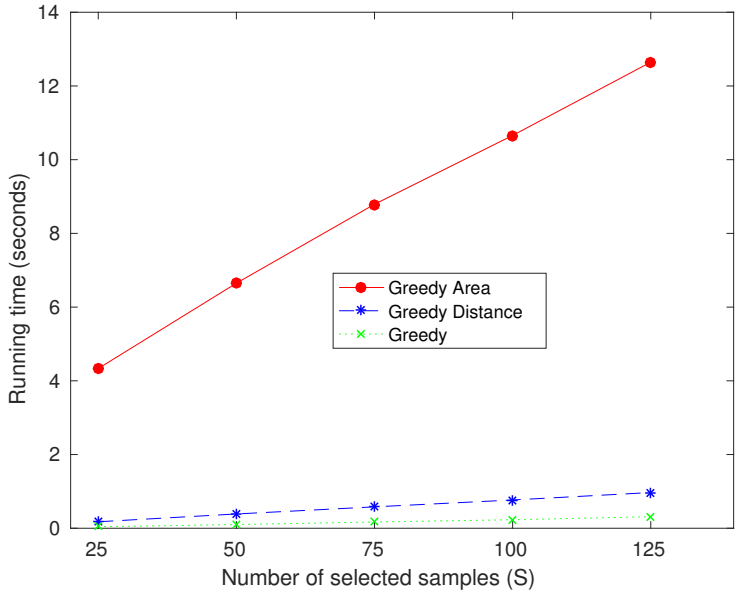

Figure 4.4. The running time versus the number of selected nodes (S).

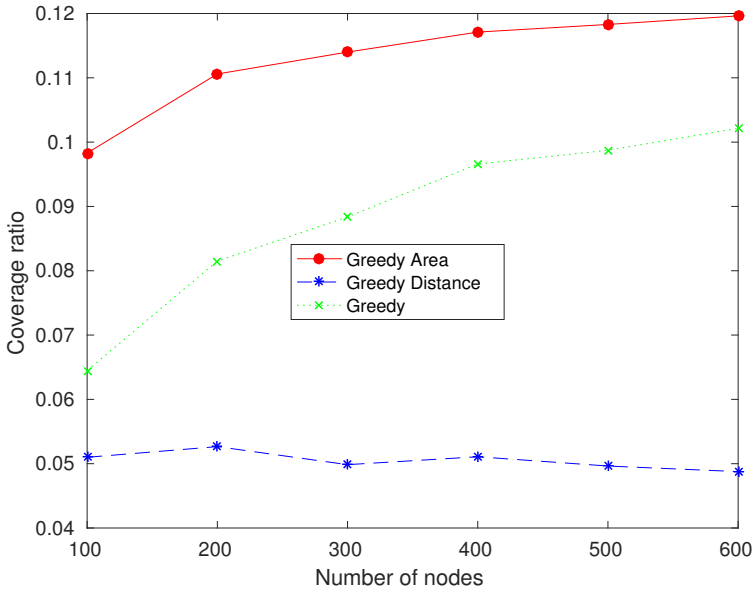

Figure 4.5. The coverage ratio versus the number of available nodes.

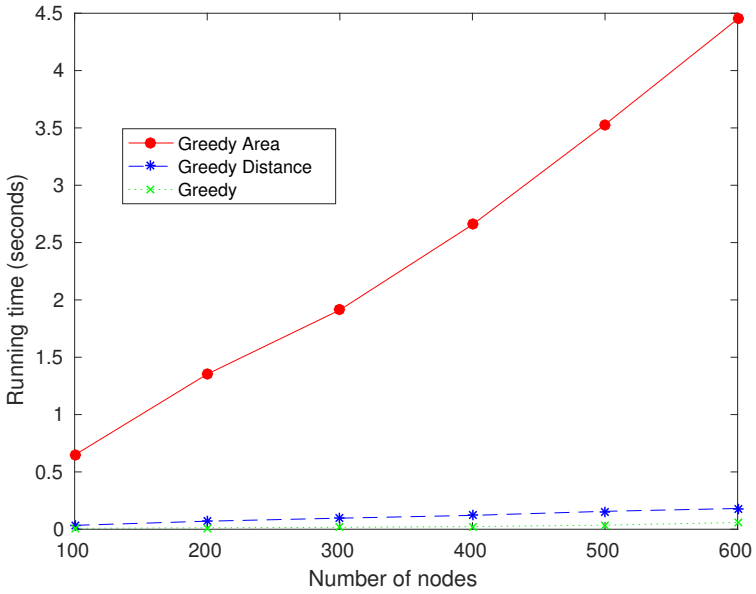

Figure 4.6. The running time versus the number of available nodes.

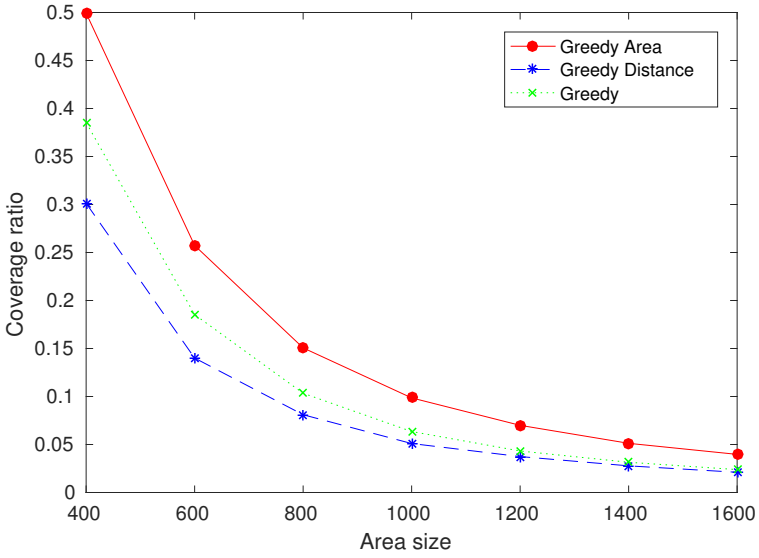

Figure 4.7. The coverage ratio versus the area size.

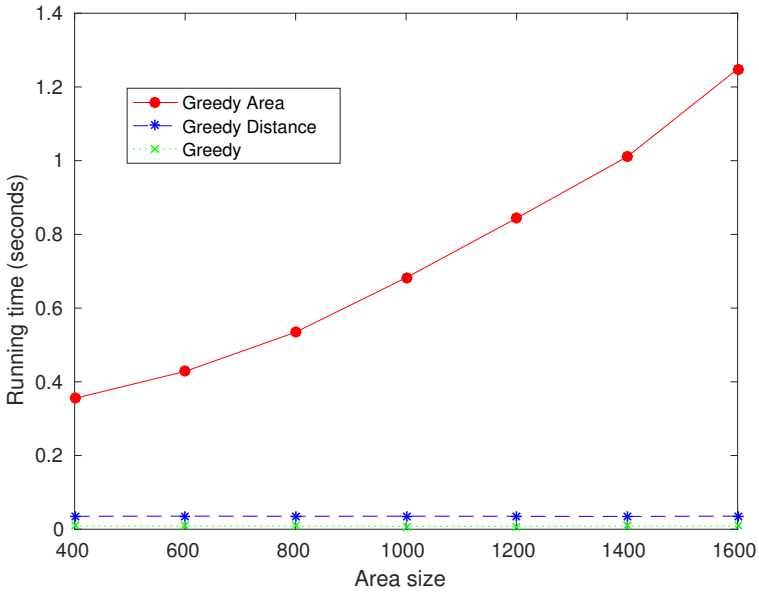

Figure 4.8. The running time versus the area size.

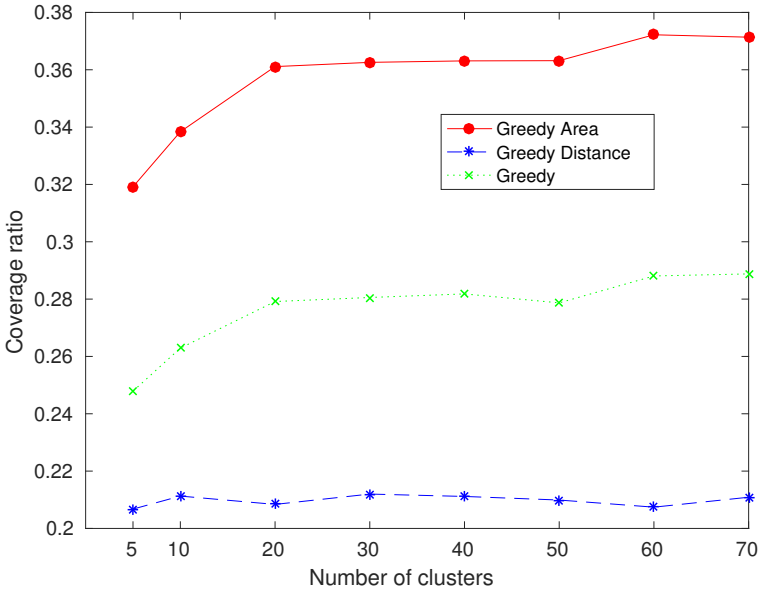

Figure 4.9. The coverage ratio versus the number of clusters for nonuniform placement of nodes.

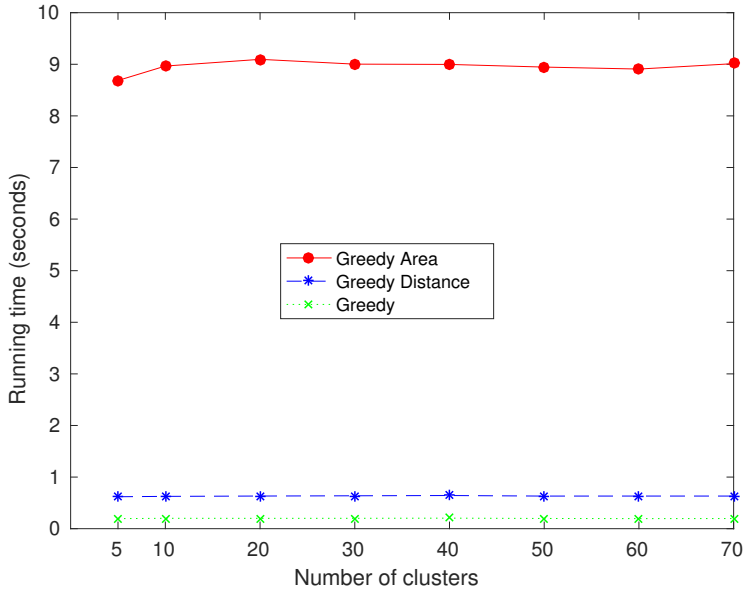

Figure 4.10. The running time versus the number of clusters for nonuniform placement of nodes.

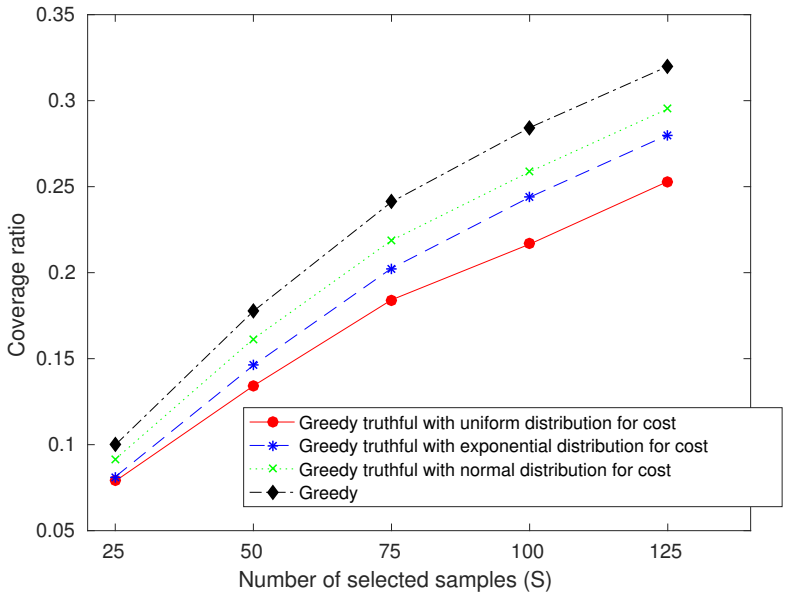

Figure 4.11. Impact of the distribution of cost in Greedy sampling coverage ratio versus the number of selected nodes (S).

CHAPTER 5

PRESERVING LOCATION PRIVACY IN RADIO NETWORKS

5.1 Introduction

Location of a wireless device or a human represents an important piece of information about the device or the human. This information can be used by adversaries for potentially dangerous life threatening attacks. In this chapter, we develop the first solution to the location privacy problem, where neither the attacker nodes nor the tracked moving object transmit any RF signals, using a game theoretic framework.

Wireless devices in a wireless network create a radio wave field in and around the area in which the network is deployed. Moving objects and people can disturb the field in ways that can be measured at locations in and outside of the deployment area [25, 67, 75]. Essentially, the radio network information is leaked well beyond the perimeter in which the radio network is deployed. In our research, we investigate defense mechanisms against attacks where person location can be inferred using the radio characteristics of wireless links (e.g., the received signal strength, RSS, of wireless links). In these attacks, a person or a group has one or more wireless devices (wireless access points/sensor nodes) deployed in an area in which they expect privacy, for example, their home. An attacker can deploy a network of receivers which measure the received signal strength of the radio signals transmitted by the legitimate wireless devices, allowing the attacker to learn the locations of people moving in the vicinity of the devices, information that the attacker would not be able to know if the wireless devices did not exist. Such an attack is possible even when the network is otherwise secure against data eavesdropping.

Consider a scenario where military personnel setup a base in an area surrounded by a tall concrete wall. Among the other facilities on the base, there are various wireless networks used for voice, video, and data communications among the personnel, on and

off the base. Security protocols, e.g., encryption, are used such that an adversary cannot eavesdrop on the data communicated. However, an attacker sets up a network of receivers, in locations outside of the wall of the base, which measure various characteristics of any signals that these receive and can infer where the people inside the base are and choose those areas to bomb for causing maximum damage.

In a different scenario, paparazzi might use the radio network leakage to learn where in a celebrity's house people are located, and be able to know beforehand from which gate and when the celebrity may exit. Having a wireless network leaking the positions and number of people in the area of the network is generally problematic in a variety of different contexts where people rightfully expect such information to be private. The potential for invasion of privacy is significant. Fundamentally, any transmitted radio signal interacts with the radio propagation environment in a way that can be measured at a receiver. By using multiple distributed receivers and observing the changes over time, an eavesdropper can estimate where the changes in the environment are occurring, and infer human or other moving object locations. Essentially, a wireless network *leaks information* about the locations of people in the vicinity of that network to anyone who wishes to and is capable of listening.

In order to motivate our research, we show a temporal plot of RSS measured by a receiver outside of a building wall, in Figure 5.1. One can automatically identify periods in which a person is crossing the line between transmitter and receiver by comparing the short and long term variance. In general, environmental noise causes very little variation in the RSS of a wireless link. However, human presence in the vicinity of the link causes a high temporal variation. Thus, if we monitor the variance in the RSS of each link and observe a very high short term variance in some link, we can infer that a human is obstructing the line of sight path of the link.

There is a growing amount of work ([76][77][78][12]) that shows how radio signals can be used for obtaining location information of moving objects that are not transmitting any radio signals themselves. Adib et al. [76] have developed WiVi to track moving humans through walls. In a follow up work [77], the authors propose an approach to track the 3d location of the moving object through walls. In another work [78], the reflection of wireless signals from a human body is used to identify human gestures. More recently,

Banerjee et al. [12], have demonstrated how humans can be tracked through walls without transmitting any signals from the attack nodes. One could possibly consider using defensive jamming [79] to corrupt the transmitter signals and preserve location privacy. However, in [12] the authors demonstrate that even by adding noise to the radio network, the attacker is still able to locate persons within the building.

In this chapter, we develop the first solution to the location privacy problem above, where neither the attacker nodes nor the tracked moving object transmit any RF signals, using a game theoretic framework. In our game theoretic model, the defender (the genuine wireless network) deploys multiple transmitters in different locations and changes transmitters in some random or probabilistic fashion to minimize the chance of the attack receivers locating the people inside certain parts of the building. Figure 1.2 shows an overview of the attack due to radio network leakage. In this figure, an attacker is interested in monitoring an area of interest inside the building. The defender deploys four transmitters. When transmitter Tx 1 transmits, it would make the most sense to place the attacker receivers in a strategic area 1 to monitor the area of interest. When transmitter Tx 2 transmits, it would make most sense to place the attacker receivers in the strategic area 2 to monitor the area of interest. The attacker need not deploy attack receivers in all strategic areas because of cost. More importantly, the higher number of attack receivers the attacker deploys the higher the probability of it being detected (e.g., by security cameras or guards etc.). Furthermore, the attacker cannot "quickly" move and deploy attack receivers from one strategic area to another. Therefore, by appropriately changing the transmitter location the defender can defend against the attacker.

Note that we only show the transmitters and attack receivers in Figure 1.2. Movement can still be detected in the presence of other objects both inside and outside the monitored area [12]. Furthermore, while we show only one kind of transmitter, a WiFi access point, and only one kind of receiver, laptops, other wireless devices or nodes with wireless capabilities can also contribute to or be used to create radio network leakage attacks. Additionally, Figure 1.2 shows only one kind of building perimeter. Our research applies to other building perimeters as well.

We model this attacker-defender scenario as a Stackleberg game, which is a sequential game where the defender plays first, then the attacker selects its best strategy by observing

the defender's strategy. Our goal is to maximize the defender's benefit, i.e., maximize location privacy. The defender's strategy is to probabilistically schedules transmitters. The attacker deploys attack receivers in strategic areas outside the building perimeter. The attacker has limited resources and incurs a cost in deploying attack receivers; the higher number of attack receivers the attacker deploys the higher the probability of it being detected (by security cameras or guards etc.). Therefore, the attacker tries to deploy attack receivers only in a strategic area that maximizes its chance of violating location privacy.

Game theory provides us with a methodology to allocate limited security resources to protect systems and infrastructure, taking into account the different weights of different targets and an adversary's response to any particular attack prevention strategy [80, 81]. Game theory allows for modeling situations of conflict and for predicting the behavior of participants. In situations where one of the players has the ability to enforce its strategy on the other, the game is called a Stackleberg game. In the Stackleberg game [81], the player who announces its strategy first is called the leader and the other player who reacts to the leader's strategy is called the follower. In our problem context, the leader is the defender trying to ensure that the attack receivers cannot accurately infer people's location in the area of interest, and the attacker is the follower trying to suitably place the attack receivers to maximize its utility. Our goal in this chapter is to use the Stackelberg game model to find a probabilistic scheduling for the defender while minimizing the possibility of determination of people's location by the attacker.

Our contributions in this chapter are as follows. First, we model the radio network leakage attack using a Stackelberg game. Second, we define utility and cost functions related to the defender and attacker actions. Third, using our utility and cost functions, we find the optimal strategy for the defender by applying a Greedy method. We experimentally evaluate our game theoretic model in two different settings: in an open environment and a cluttered office. Our experimental results show that when using our approach, the minimum localization error for the attacker increases by $36\% - 240\%$. Higher localization error corresponds to more privacy. We expect the localization error for the attacker to be significantly higher for larger areas. We briefly discuss the practicality of deploying our approach before concluding the chapter.

The rest of this chapter is organized as follows. Section 5.2 contains the relevant re-

lated work. Section 5.3 provides some preliminary game theoretic concepts. Section 5.4 represents our adversary model. In Section 5.5, we formulate the problem and develop our solution. Experimental results are reported in Section 5.6. Section 5.7 is devoted to practical considerations. The concluding remarks and some directions for future work are provided in Section 5.8.

5.2 Related Work

Location of a wireless device or a human represents an important piece of information about the device or the human. This information when available to adversaries can be used for privacy violation. More seriously, it can be used by adversaries for potentially dangerous life threatening attacks. There is a growing amount of work (e.g., [25, 67, 75, 82–84]) that shows how devices or humans can be localized in both benign and malicious settings. There are some interesting existing solutions for preserving location privacy as well (e.g., [85, 86]). However, a vast majority of these are for preserving the privacy of active transmitters' locations. In these systems, the wireless device (e.g., a mobile phone, RFID tag, low-power radio transceiver) that is being located is actively communicating with the surrounding network infrastructure (e.g., WiFi APs, RFID readers, and other radio transceivers).

In their recent work [12], the authors demonstrate that the presence, location, and movements of people not carrying any wireless device can still be eavesdropped by measuring the RSS of the links between the devices composing the legitimate network and few receivers positioned outside the target area. This can be achieved without requiring a complex network infrastructure or previous access to the target area for an initial calibration. This paper [12] also proposes a defense mechanism to fool the attack receivers by changing the power at the wireless transmitters. However, it is found in [12] that the attacker can compensate for the changes in the transmit power and still determine the human locations. The compensation mechanism is based on the intuition that an artificial transmit power change at a transmitter will impact all the links between the transmitter and the attack receivers, whereas genuine power changes due to human movement are likely to impact only some of the links. Thus, there is a need to seek newer, novel solutions that can effectively preserve human location privacy in spite of radio network leakage.

We propose a novel method based on a game theoretic framework to tackle the location privacy problem in spite of radio network leakage. Our game theoretic model is based on a Stackelberg game that has been used in attacker-defender scenarios [87–89]. However, our work is the first application of the Stackelberg game for the purpose of protecting location privacy in radio networks.

5.3 Game Theory Preliminaries

Game theory is a study of strategic interactions among selfish agents that yields the desired outcome by considering the preferences of agents. Stackelberg game is a type of sequential game where one player, the leader, commits to a strategy first and the other players, followers, selfishly choose their best response strategies considering the leader's strategy. This type of game is commonly used for modeling attacker defender scenarios in the security domain where the defender commits to the strategy first. Table 5.1 shows a simple example of the Stackelberg game between an attacker and a defender. The defender is the row player and the attacker is the column player. Here, d_1 and d_2 are the defender strategies, and a_1 and a_2 are the attacker strategies. In this game the best strategy for the defender is d_1. In this case, the attacker plays a_1. So, the utility of defender is 3. However, if the defender plays d_2, then the attacker plays a_2. As result, the utility of defender will be 1. As shown in this example, in the Stackelberg game, the goal is to maximize the utility of the first player, defender. To do this, the defender chooses a strategy with maximum utility by taking the attacker's best strategy into account.

If the defender plays deterministically (e.g., in Table 5.1 it plays a pure strategy d_1), then the attacker knows the exact strategy of the defender and selects the pure strategy a_1. However, if the defender plays probabilistically (by assigning probabilities to pure strategies d_1, d_2), then the attacker is not able to find the exact action of the defender in real time.

5.4 Adversary Model

We make the following assumptions about the attacker:

- The attacker is able to deploy multiple attack receivers within the transmission range of the legitimate transmitter(s) outside the area being monitored. The attacker is able

to measure the physical layer information (e.g., the received signal strength) of the links between the transmitter(s) and the attack receivers and localize humans moving inside the monitored area that cross the transmitter-receiver links.

- The attacker does not have access to the content of the packets transmitted by the legitimate network nodes. It does not depend on understanding the content of the packets.
- The defender does not know where the attacker will place its receivers and the attacker does not know the exact locations of transmitters.

5.5 Problem Formulation and Solution

We develop a game-theoretic framework for preserving location privacy in radio networks. Our framework is characterized by one defender and multiple target regions that the defender wishes to protect from a location privacy attack. We use the attack scenario shown in Figure 5.2 for our problem formulation. Figure 5.2 is a more generalized version of the attack scenario shown in Figure 1.2. Figure 5.2 shows multiple target regions. These target regions are comprised of predefined subsections of a building where people move. In this setting, the defender cannot schedule transmitters deterministically; otherwise the attacker will definitely succeed in violating the location privacy of a target region by deploying attack receivers in the best strategic area outside the building and measuring the variations in wireless signal strength. As a result, the defender should adapt an unpredictable scheduling strategy, randomizing over the transmitters with the goal of minimizing the possibility of determination of people's location by the attacker. In our work, we use the Stackelberg game to formulate the defender/attacker scenario. In this game, the defender commits to a strategy first that is optimal (maximizes its expected utility). Then, the attacker plays its best strategy considering the strategy the defender plays. The goal is to maximize the utility of defender in a way that even if the attacker selects its best strategy, the utility of attacker will be minimum. This goal essentially corresponds to maximizing the localization errors for the attacker.

We describe the notation that we use in our problem formulation in Table 5.2.

5.5.1 System Model

In our game, we have two players, a defender, d, and an attacker, a. We consider a set of transmitters, $T = \{t_1, t_2, \ldots, t_K\}$ deployed at different locations inside a building. We also consider a set of target regions, $R = \{r_1, r_2, \ldots, r_N\}$, that the defender wants to protect from a location privacy attack. In addition, we assume that set of strategic area locations $S = \{s_1, s_2, \ldots, s_M\}$ exists, where the attacker deploys its attack receivers.

The defender and the attacker strategies are dependent on the time of the day. This happens because the probability of human movements in a target region changes dependent on the time of the day. For example, in a campus environment, most students and faculty members move towards a specific place for lunch at a specific time [90] and if the attacker attacks at that time it can cause the maximum damage. To consider the variations of the attacker and the defender strategies during the time, we divide time into T slots.

In a recent work on location detection [20] using radio tomographic imaging, the authors showed that only those transmitters that have radio wave fields inside the moving area are effective in location detection. We define effective transmitters for each target region as the minimum number of transmitters that need to be turned off in order to preserve the location privacy of a specific target region. When the effective transmitters of a target region are turned off, an attacker cannot measure the change of the received signal strength caused by people movements. However, the defender cannot turn off all transmitters to protect all target regions or turn off a set of transmitters at all times to protect one target region (depending on the actual application of the transmitters). Therefore, the defender needs a strategy for turning off transmitters. The strategy should be random instead of deterministic with the goal of reducing the chance of an attacker determining the location of people inside target regions while considering the limits on the number of transmitters that can be turned off.

The pure strategy for the defender, $\vec{\sigma_d} = (\sigma_d^1, \sigma_d^2, \ldots, \sigma_d^T)$, is a row vector determining the effective transmitters of which target region are turned off in each time slot and the pure strategy for the attacker, $\sigma_a = (s_a, r_a)$, is selecting one strategic area and also the number of active receivers in the strategic region.

To make it hard for attackers to find its exact actions, the defender uses a randomized, mixed strategy, instead of a pure strategy. The defender mixed strategy, $\vec{m} = (m_1, m_2, \ldots, m_{|\sigma_d|})$,

essentially describes the probability of playing each pure strategy.

5.5.2 Utilities

The defender and attacker utilities depend on whether the attacker is able to attack or not. Let $PD(j, \sigma_a)$ denote the probability of detection of the target region j by the attacker, if the attacker plays $\sigma_a = (s_a, r_a)$. We formulate $PD(j, \sigma_a)$ as the following:

$$PD(j, \sigma_a) = I(j, s_a)D(r_a, s_a) \tag{5.1}$$

$I(j, s_a)$ in (5.1) determines if s_a, the strategic area that is selected by the attacker, is target region j's corresponding strategic area. For target region, j, s_a is a corresponding strategic area if the attacker can detect movements in target region j from that strategic area. Depending on the location of target regions inside the building, there will be multiple corresponding strategic areas for each target region.

Let $CR(j)$ denote the corresponding strategic areas for target region j. Also, let $Cov(j, s_a)$ represent the percentage of the whole area that is covered by s_a in case of movements in target region j. Then, in (5.1), $I(j, s_a)$ can be obtained by the following function:

$$I(j, s_a) = \begin{cases} Cov(j, s_a) & s_a \in CR(j) \\ 0 & otherwise \end{cases} \tag{5.2}$$

$D(r_a, s_a)$ in (5.1) represents the probability of detection if the attacker deploys r_a receivers in s_a. We obtain this probability by the following function.

$$D(r_a, s_a) = \begin{cases} (\frac{r_a}{r_{max}(s_a)})^{\alpha} & 0 < r_a \leq r_{max}(s_a) \\ 1 & r_a \geq r_{max}(s_a) \end{cases} \tag{5.3}$$

Here, $r_{max}(s_a)$ in (5.3) is the saturation point. Based on our experimental evaluations, the probability of detection increases by increasing the number of receivers. However, the probability of movement detection does not increase by increasing the number of receivers beyond this saturation point. In other words, the probability of movement detection is 1 beyond the saturation point, r_{max}.

The value of r_{max} depends on the location of strategic region, s_a. For instance, if there are some "radio" obstacles in some strategic region outside a building (e.g., a metal door), then the attacker needs to deploy more attack receivers in those areas to detect changes in the RSS. Figure 5.3 shows the value of $D(r_a, s_a)$ with $r_{max}(s_a) = 5$ and α equal to 1, 2, and 3. Based on our experiment, we set $\alpha = 2$ in our evaluation.

Besides the probability of attack, $PD(j, \sigma_a)$, the utilities of the defender and the attacker also depend on whether or not the defender turns off effective transmitters of the target region. $c_{j,t}(\vec{m})$ represents the probability that effective transmitters of target region j are turned off in time slot t under mixed strategy $\vec{m}$ and is equal to $c_{j,t}(\vec{m}) = \sum_{m_i \in \vec{m}} m_i x_{j,t}(m_i)$. m_i is the probability of playing the pure strategy i. $x_{j,t}(m_i) \in \{0, 1\}$ shows whether or not the effective transmitters of target region j are turned off in time slot t for strategy i. $x_{j,t}(m_i) = 1$, if the effective transmitters of target region j are turned off in time slot t. Otherwise, $x_{j,t}(m_i) = 0$.

We obtain the utility of the defender, $u_d(\vec{m}, \vec{\sigma_a})$, and the utility of the attacker, $u_a(\vec{m}, \vec{\sigma_a})$, when the defender plays the mixed strategy $\vec{m}$ and the attacker plays $\vec{\sigma_a}$, from the following formulas.

$$u_d(\vec{m}, \vec{\sigma_a}) = \sum_{j=1}^{N} PD(j, \sigma_a) \sum_{t=1}^{T} \left[c_{j,t}(\vec{m}) Re_d(j,t) - (1 - c_{j,t}(\vec{m})) Pe_d(j,t) \right] \tag{5.4}$$

$$u_a(\vec{m}, \vec{\sigma_a}) = \sum_{j=1}^{N} PD(j, \sigma_a) \sum_{t=1}^{T} \left[(1 - c_{j,t}(\vec{m})) Re_a(j,t) - c_{j,t}(\vec{m}) Pe_a(j,t) \right] \tag{5.5}$$

We observe from equation (5.4) that in each time slot t and for each target region j, the utility of the defender will increase by the amount of reward, $Re_d(j,t)$, if the defender turns off the effective transmitters of target region j when being attacked in time slot t. Otherwise, this utility will decrease by the amount of penalty, $Pe_d(j,t)$. Note that the defender's goal is to maximize $u_d(\vec{m}, \vec{\sigma_a})$ in an equilibrium. This means that $\vec{\sigma_a}$ should be the best strategy for the attacker when the defender plays the mixed strategy $\vec{m}$. In addition, the defender should consider the limits on the number of transmitters that can be turned off (Constraint 5.8). Otherwise, $C = 1$ is the optimal strategy for the defender. As in the case of the defender, the utility of the attacker increases by amount of reward, $Re_a(j,t)$, in case of a successful attack (effective transmitters of target region j are turned on in time slot t) and decreases by amount of penalty, $Pe_a(j,t)$, if the attacker is unable to attack the target region j in time slot t.

The amount of reward and penalty for a successful attack on target region j in time slot t, $Re_a(j,t)$ and $Pe_d(j,t)$, depends on the probability of movement in target region j in time slot t. The attacker can cause more damage to the target region with higher probability of

movement. As a result, Re_a and Pe_d are greater for a target region with a higher probability of movement.

Let P be a $N \times T$ matrix where N, and T denote number of target regions and number of time slots, respectively. In this matrix, each item, $p_{j,t}$, represents the probability of movement for target region j in time slot t. Then, $Re_a(j,t)$ and $Pe_d(j,t)$ are obtained as follows:

$$Re_a(j,t) = Pe_d(j,t) = \frac{p_{j,t}}{\sum_{j=1}^{N} p_{j,t}} \lambda_a \tag{5.6}$$

Here, λ_a is a constant tunable parameter for both reward and penalty in case of a successful attack.

$Re_d(j,t)$ and $Pe_a(j,t)$ depend on the number of receivers that are chosen by the attacker. As the number of receivers increases, the attacker's probability of detecting movements also increases. However, increasing the number of active receivers also increases the deployment cost of the attacker and the probability of being detected by security cameras or guards. Let $cost(\sigma_a)$ represent this cost that depends on the number of active receivers. Then, $Re_d(j,t) = Pe_a(j,t) = cost(\sigma_a)$. The dependence of this cost on σ_a is due to the fact that the attacker can deploy different number of receivers. We express this cost as $cost(\sigma_a) = r_a \times \lambda_c$, where, λ_c is the cost for adding one receiver.

5.5.3 Optimization Problem

Having determined the utilities of the attacker and the defender, characterized in (5.4) and (5.5), we can now find the Stackelberg equilibrium. In the Stackelberg game, the defender probabilistically turns off the effective transmitters in advance. Then the attacker makes its own choice to attack a specific target region.

To find the Stackelberg equilibrium, the defender has to calculate the best reply of attacker to each of its mixed strategies and selects the mixed strategy that maximizes the defender utility. Formally, the Stackelberg equilibrium can be obtained by solving the following optimization problem:

$$\underset{\vec{m},\vec{\sigma_a}}{\text{argmax}}(u_d(\vec{m},\vec{\sigma_a}))$$

$$s.t.$$

$$u_a(\vec{m},\vec{\sigma_a}) \geq u_a(\vec{m},\vec{\delta_a})\forall\vec{\delta_a} \tag{5.7}$$

$$\sum_{t=1}^{T}\sum_{j=1}^{N} c_{j,t}(\vec{m}) \leq mT \tag{5.8}$$

The objective function in the above optimization problem maximizes the utility of the defender. Also, in order to obtain an equilibrium, the output of the optimization problem, $(\vec{m},\vec{\sigma_a})$, should be optimal for the attacker as well. Constraint (5.7) represents this attacker optimality requirement. In other words, given the defender mixed strategy, $\vec{m}$, the attacker strategy, $\vec{\sigma_a}$, should be its best strategy. This ensures that if the first player, defender, plays $\vec{m}$, then the second player, attacker, plays $\vec{\sigma_a}$. Therefore, the solution of the above optimization problem, $(\vec{m},\vec{\sigma_a})$, results in a Stackelberg equilibrium. Constraint (5.8) states that the sum of probabilities of turning off the effective transmitters of all target regions in T time slots must be equal to mT, where $m < N$ is number of target regions that defender is able to protect in each time slots.

In order to solve the optimization problem, we use a Greedy approach that is inspired by the scheme presented in [91]. Our Greedy approach is described in Algorithm 4. In this approach, instead of finding the optimal mixed strategy, $\vec{m}$, for the defender, we find $c_{j,t}(\vec{m})$ for all target regions and all time slots. Let C be a $N \times T$ matrix where N and T denote number of target regions and number of time slots, respectively. We know $\sum_{j=1}^{N} c(j,t) \leq m$ and $\sum_{j=1}^{N}\sum_{t=1}^{T} c(j,t) \leq mT$. We initialize $C = 0$ in the first iteration of algorithm, and update the values of C for all time slots based on the Greedy strategy in each iteration. At each iteration, we add Δ to one of N target regions in the time slot that maximizes the utility of the defender in the same time slot given the attacker best strategy for current C, σ_a^* (see lines 4-14 in Algorithm 4). Δ is a small value between 0 and 1. The algorithm is repeated until $\sum_{i=1}^{n}\sum_{t=1}^{k} c(i,t) = mT$.

Time complexity: In each iteration of the Greedy algorithm, we have one loop that iterates over T, the number of time slots. Also, for each time slot, we must find the target region that maximizes the utility of defender which takes $O(N)$. Recall that N is the number of target regions. Therefore, each iteration of the algorithm takes $O(TN)$. The number of iterations over the Greedy algorithm is $\frac{1}{\Delta}$. Then, the time complexity of the

Algorithm 4: Greedy Approach

input : : $C_{N\times T} = 0, \Delta$
output: : $C_{N\times T}$

while *(1)* **do**
  $C'_{N\times T} = C_{N\times T}$;
  foreach *time slot t* **do**
    `// Initialization`
    $u_d^* = -\infty$;
    $selected\_target\_region = 0$;
    foreach *target region j* **do**
      $c'_{j,t} = c'_{j,t} + \Delta$;
      Find the attacker best strategy, σ_a^*, $\sigma_a^* \in \underset{\sigma_a}{\text{argmax}}(u_a(C', \sigma_a))$;
      if $u_d(C', \sigma_a^*) > u_d^*$ **then**
        $u_d^* = u_d(C', \sigma_a^*)$;
        $selected\_target\_region = j$;
      end
    end
    `// Update C`
    $c_{selected\_target\_region,t} = c_{selected\_target\_region,t} + \Delta$;
  end
  if $\sum_j \sum_t c_{j,t} = mT$ **then**
    exit;
  end
end

Greedy Monte Carlo algorithm is $O(\frac{1}{\Delta}TN)$. Note that the values of $\frac{1}{\Delta}$, N, and T are small.

5.6 Evaluation

In this section, we evaluate our game theoretic formulation and solution with the help of experimental results. More specifically, we wish to determine the effectiveness of the solution to our optimization problem, C, that determines the probability of turning off the effective transmitters of each target region in each time slot. We determine the effectiveness of our solution by showing that, given the optimal strategy for the defender, C, the attacker is unable to determine people location with a high accuracy.

We conduct experiments in two different areas: an open environment, and a cluttered office. In this section, we first describe an attack scenario in these two areas and then present the evaluation of our game theoretic approach in terms of average localization error in these two areas.

5.6.1 Experiment Layout

- ***Open Environment:*** In the open environment, there are no objects or obstructions in the monitoring area. Figure 5.4 shows the layout of this experiment. In this figure, there are 8 transmitters, each an RF sensor node, that are deployed inside of a $70m^2$ area at the height of one meter from the floor. Also, there are 30 attack receivers that are placed in four heterogeneous strategic areas (see ellipse in Figure 5.4) with r_{max} of 7, 8, 7, and 8 (these numbers of attack receivers are typical for localizing human inside perimeters [12]). The sensor nodes transmit on channels 11, 15, 18, 22 and 26 (multiple channels improve the accuracy of location determination [28]).

  There are four target regions that represent bounded areas of movement during a day. To take into account the heterogeneity of movements in time domain, we consider 3 time slots with different probability of movements in each target region. The dashed circles in Figure 5.4 determine the target regions. In our experiment, a single person moves in the target regions with the following probability matrix, P. Recall that $p_{j,t}$ represents the probability of movement for target region j in time slot t. Thus, the first column of matrix P, $(0.8, 0.2, 0, 0)$ represents a single person spending 80 percent of time in target region 1 and 20 percent of the time in target region 2 in time slot 1.

$$P = \begin{bmatrix} 0.8 & 0 & 0 \\ 0.2 & 0.4 & 0 \\ 0 & 0.6 & 0.3 \\ 0 & 0 & 0.7 \end{bmatrix} \tag{5.9}$$

- ***Cluttered Office:*** This experiment is done in a cluttered area where there are several metallic obstructions such as desks, chairs, and monitors. In this experiment too, 8 transmitters are deployed inside of a $52m^2$ area at the height of one meter from the floor, as shown in Figure 5.5. The sensor nodes transmit on channels 11, 16, 21, and 26. Also, there are 14 attack receivers that are placed in four heterogeneous strategic areas with r_{max} of 4, 3, 4, and 3. In this experiment, one person moves in three target regions (see dashed circles in Figure 5.5) and there are three time slots with the following probability matrix.

$$P = \begin{bmatrix} 0.8 & 0 & 0 \\ 0.2 & 0.2 & 0 \\ 0 & 0.8 & 1 \end{bmatrix} \tag{5.10}$$

As explained before, P represents the probability of movement.

In both experiments (open environment, and cluttered office), the sensor nodes composing the network are TI CC2531 USB dongle nodes [53]. While we only consider one person in our experiments, our framework can also work when multiple persons move in the monitored area. In this case, the amount of reward and penalty for a successful attack, $Re_a(j,t)$ and $Pe_d(j,t)$, depend on both the probability of movement and the number of moving people.

5.6.2 Experimental Results

To evaluate our framework, we first find the effective transmitters for each target region. As mentioned before, the effective transmitters for a target region are the minimum number of transmitters so that when these are turned off, the attacker is unable to detect movements in the target region. For example in Figure 5.4, transmitter 1 and 6 are the effective transmitters of target region 1.

Table 5.3 shows the localization error in open environment when a person moves in target region 1. In order to localize the moving person, we use the multichannel RTI approach proposed in [28] (using multiple channels increases the accuracy of localization). The first row shows the average localization error when all 8 transmitters are turned on and attack receivers are deployed in different strategic areas. The average localization error in this row shows that strategic areas corresponding to target region 1 are 1, 2, 4. Among these strategic areas, strategic area 4 has the maximum coverage. The second row of Table 5.3 shows the average localization error when the effective transmitters of target region 1 are turned off. This row shows that the average localization error increases between 1-3.5 meters by turning off transmitter 1 and transmitter 6 that are effective transmitters of target region 1. Note that strategic area 3 is not suitable for target region 1, and turning off the effective transmitters of target region 1 does not change the average localization error for this strategic area. The third row of Table 5.3 shows the average localization error when we turn off any other pair of transmitters except transmitter 1 and transmitter 6. This row shows that the average localization error does not change much in comparison to that when we turn off the effective transmitters of target region 1.

We find the effective transmitters and corresponding strategic areas for each target

region in the open environment and the cluttered office setting using the same reasoning as above. If we have more than one option for effective transmitters of a target region, then we select the effective transmitters that have minimum overlap with the effective transmitters of other target regions. After finding the effective transmitters and corresponding strategic areas for each target region, we focus on an attack scenario where an attacker tries to determine the location of a moving person within a target region.

First, we find the optimal strategy of the defender, C, for the two scenarios (open environment and cluttered office). In both experiments, we set Δ and m to 0.001 and 1, respectively. Recall that Δ is the increment in the probability of turning off effective transmitters; and m is the number of target regions that defender is able to protect. We also set λ_a and λ_c to 1, 0.5, respectively. Recall λ_a is constant parameter for a successful attack and λ_c is the cost of adding one receiver. Our methods work for a wide range of values of the above parameters. Here, we show results only for some specific values.

After finding the optimal strategy for the defender, C, which determines the probability of turning off the effective transmitters of each target region in each time slot, we randomly sample C to find a specific schedule for turning off the effective transmitters of each target region in each time slot. Given the optimal strategy for the defender, we evaluate the average localization error when the attack receivers are deployed in each strategic area.

Table 5.4 shows the average localization error for each strategic area in the cluttered office scenario. The first row of the table shows the average localization error when all transmitters are turned on. The values in this row show that the attacker can detect the movement with minimum localization error of 2.5 meters from strategic location 4 when all transmitters are active in all time slots. The second row of this table shows the average localization error due to the optimal scheduling determined by our game theory framework for each strategic area. The values in the second row of Table 5.4 show that the minimum localization error for the attacker is 3.4 meters if the attacker deploys attack receivers in strategic area 4. Thus, the attacker's best strategy is to deploy attack receivers in strategic area 4. Table 5.4 shows that when the optimal scheduling policy determined by our game theoretic framework is used, even if the attacker plays its best strategy, the localization error increases by 36% in a $52m^2$ area. Very importantly, we expect the difference in localization errors to increase with the increase in the size of the monitored

area.

Table 5.5 shows the average localization error in the open environment that is bigger in size than the cluttered office. This table shows that the minimum localization error when all transmitters are turned on is 1.2 meter and it occurs in strategic location 3. Since there is no obstruction or object in the open environment and because of using more attack receivers in each strategic area, the localization error here is less than that in the cluttered office. However, when using the optimal strategy determined by our game theoretic framework, the minimum localization error increases by 240% (from 1.2 to 4.1 in strategic area 3). As we expect, the difference in localization errors is increased by increasing the size of the monitored area.

So far, we have compared the localization error due to the optimal scheduling determined by our game theory framework with that when all transmitters are active in all time slots. Essentially, by turning off the effective transmitters according to our optimal schedule, we reduce the number of transmitters and consequently, increase the localization error.

To demonstrate the effectiveness of our game theory framework, we measure the localization error when the defender uses a simple *random* strategy for turning off the effective transmitters of each target region in each time slot. Table 5.6 shows the average localization error for each strategic area using a random strategy for the defender. As shown in this table, the minimum localization error for the attacker in the cluttered office is 2.6 meters if the attacker deploys attack receivers in strategic area 4. This implies that the percentage of increase in the localization error decreases from 36% when using our optimal strategy to 4% when using the random strategy. For the open environment, the minimum localization error decreases from 4.1 when using the optimal strategy to 2.6 when using the random strategy, i.e., the percentage of increase in the localization error decreases from 240% to 116%. Table 5.7 shows the minimum percentage of increase in localization error using the optimal strategy and the random strategy in the open environment and the cluttered office. This table shows that the optimal strategy performs better than the random strategy in both experiments. We believe that the random strategy will perform even worse than the optimal strategy with the increase in the number of target regions, strategic areas, and transmitters.

5.7 Practical Considerations

In this section, we briefly discuss the practicality of our game theoretic approach and solution.

First, while we have designed and evaluated our methods with specific utility and cost functions, our approach allows the defender to determine its strategy for other utility and cost functions considering the capabilities of the defender as well as the attacker. Thus, our formulation can be used for a variety of adversarial prowess and behavior. Second, the functionality to change wireless transmitters can be deployed in existing enterprise networks by implementing the transmitter schedule on a *controller node* that control wireless access points [92]. Third, a controller, having complete knowledge of the schedule, can also transfer the state of active "associations" on the current access point to the next one being scheduled over a wired high speed network (e.g., Ethernet). This state also includes any keys associated with the secure transmission between the clients inside the monitored area and the current access point. Thus, the choice of the actual value of the time slot between transmitter changes would depend on the overhead of this state transfer. If we choose a high value for the time slot, we minimize this overhead. However, we also disregard the heterogeneity in human movements during the time. Based on our experiences, we recommend a value of tens of minutes for the time slot. The state transfer across access points every tens of minutes over a wired network will not result in any significant overhead. The transmitters of the defender must use the same service set identifier (SSID) so that the change of transmitters is transparent to the nodes associated with them. Even with the centrality of the controller node, there is a small chance of packet loss during the state transfer across access points. However, we expect this loss to be not significant in comparison to other wireless loss.

Note that we considered the connectivity constraint in our game theoretic model by limiting the number of target regions that the defender can protect in each time slot, i.e., we limited the number of transmitters that can be turned off in each time slot. We can also add more connectivity constraints to our model. For example, if some transmitters can not be turned off for the consecutive time slots, we can add this constraint to our model by pruning the defender strategy space, i.e., all pure strategies of the defender that are contradicted by this constraint will be removed from the defender strategy space.

5.8 Conclusion

We investigated an attack on location privacy where the location of people moving inside a private area can be inferred using the radio characteristics of wireless links that are leaked by legitimate transmitters deployed inside the private area. We modeled the radio network leakage attack using a Stackelberg game and used a Greedy method to obtain the optimal strategy for the defender. Our experimental results showed that our game theoretic solution significantly reduces the chance of an attacker finding the location of people inside a perimeter. In the future, we will implement our framework in a WiFi network testbed inside our department to further study and demonstrate the practicality of our approach. We will also measure any performance degradation experienced by genuine receivers inside the building as a result of scheduling transmissions through different access points.

Beside distributed sensing, the proliferation and rapid growth of wireless devices and their capabilities also have created a huge opportunity to harness the computing power of nearby wireless devices for computation offloading. In the next chapter, we harness the computing power of nearby wireless devices for efficient computation offloading.

Table 5.1. An example of Stackelberg game payoff table

	a_1	a_2
d_1	(3, 1)	(1, 0)
d_2	(4, 1)	(1, 3)

Table 5.2. Notation

Parameter	Definition
d	Defender
a	Attacker
$\vec{\sigma_d}$	Defender pure strategy
$\vec{\sigma_a}$	Attacker pure strategy
$\vec{m}$	Defender mixed strategy
r_a	Number of active attack receivers
s_a	Strategic location
N	Number of target regions
T	Number of time slots
$PD(j, \sigma_a)$	Probability of detection of the target region j by the attacker
$c_{j,t}(\vec{m})$	Probability that effective transmitters in target region j are turned off in time slot t
$Re_d(j,t), Pe_d(j,t)$	Reward and penalty for defender
$Re_a(j,t), Pe_a(j,t)$	Reward and penalty for attacker
$u_d(\vec{m}, \vec{\sigma_a})$	Utility of defender
$u_a(\vec{m}, \vec{\sigma_a})$	Utility of attacker
m	Number of target regions that defender is able to protect in each time slot
$p_{j,t}$	Probability of movement in target region j in time slot t
Δ	Increment in the probability of turning off effective transmitters of a target region in each iteration of Algorithm 4

Table 5.3. Average localization error in open environment when a person moves in target region 1

Approach	Average localization error(m)			
	Strategic area 1	Strategic area 2	Strategic area 3	Strategic area 4
All transmitters are turned on	1.7	1.9	5.8	0.6
Effective transmitters are turned off	3	5.3	6.4	1.6
Any pair of transmitters other than effective transmitters are turned off	2	1.7	4.8	0.61

Table 5.4. Average localization error when all transmitters are turned on and when using optimal scheduling in cluttered office

Approach	Average localization error(m)			
	Strategic area 1	Strategic area 2	Strategic area 3	Strategic area 4
All transmitters are turned on	3.4	4.1	2.8	2.5
Optimal scheduling	4.4	5.8	3.7	3.4

Table 5.5. Average localization error when all transmitters are turned on and when using optimal scheduling in open environment

Approach	Average localization error(m)			
	Strategic area 1	Strategic area 2	Strategic area 3	Strategic area 4
All transmitters are turned on	2.8	1.9	1.2	2.1
Optimal scheduling	4.3	4.7	4.1	5.7

Table 5.6. Average localization error using random strategy in open environment and cluttered office

	Average localization error(m)	
	Open	Office
Strategic area 1	3	3.6
Strategic area 2	3.5	4.5
Strategic area 3	2.6	2.9
Strategic area 4	4	2.6

Table 5.7. The minimum percentage of increase in the localization error in open environment and cluttered office using random and optimal strategies.

	The minimum percentage of increase in the localization error	
	Open	Office
Random	116%	4%
Optimal	240%	36%

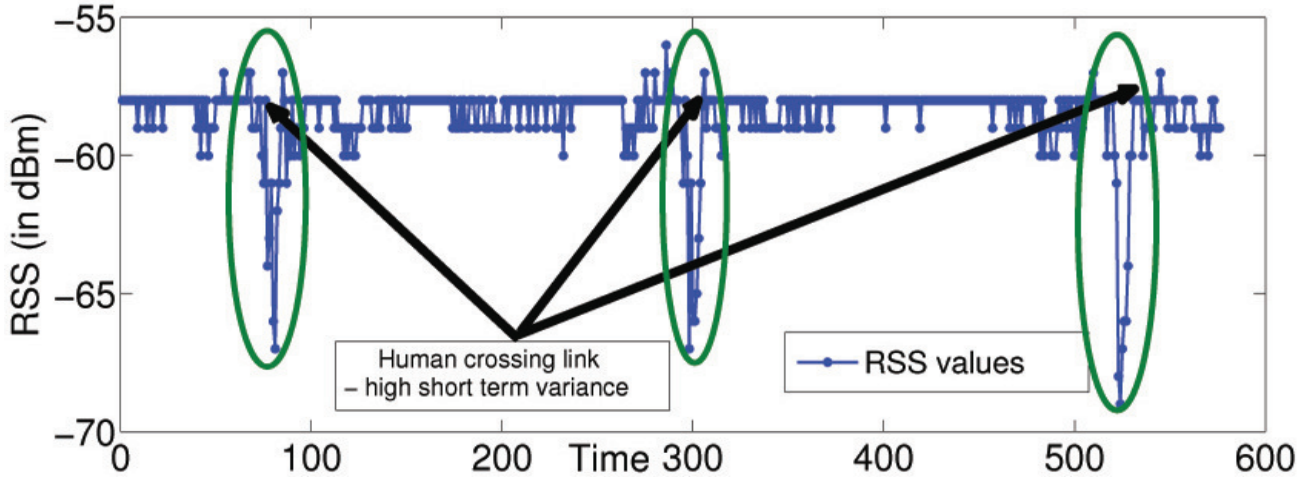

Figure 5.1. Detection of line of sight crossing.

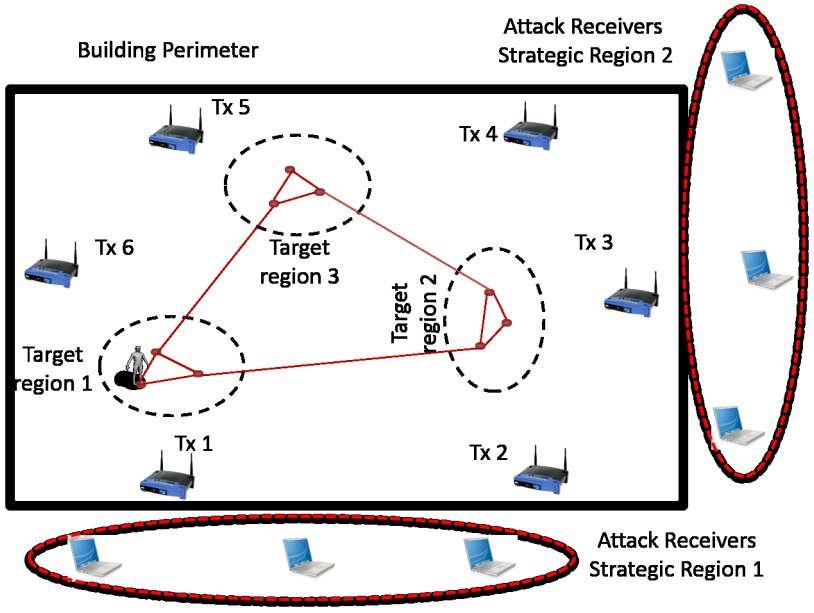

Figure 5.2. Radio network leakage attack scenario with multiple target regions.

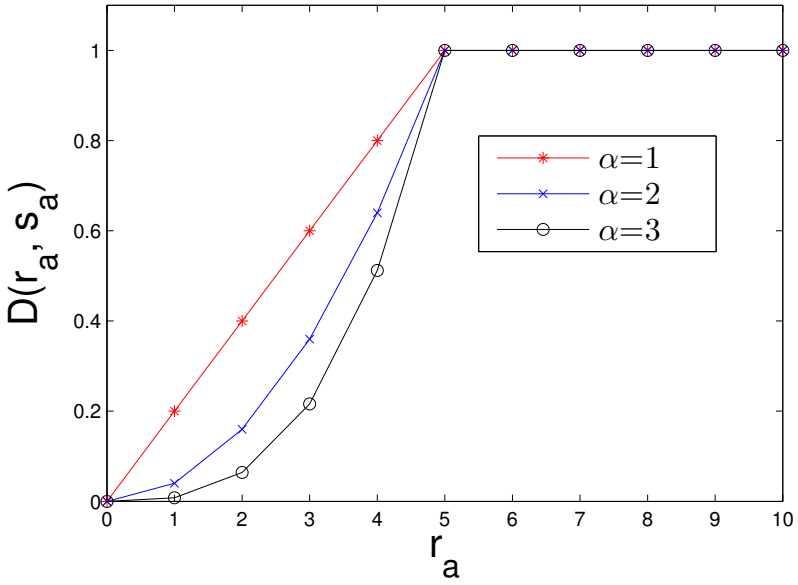

Figure 5.3. $D(r_a, s_a)$ with different values of α.

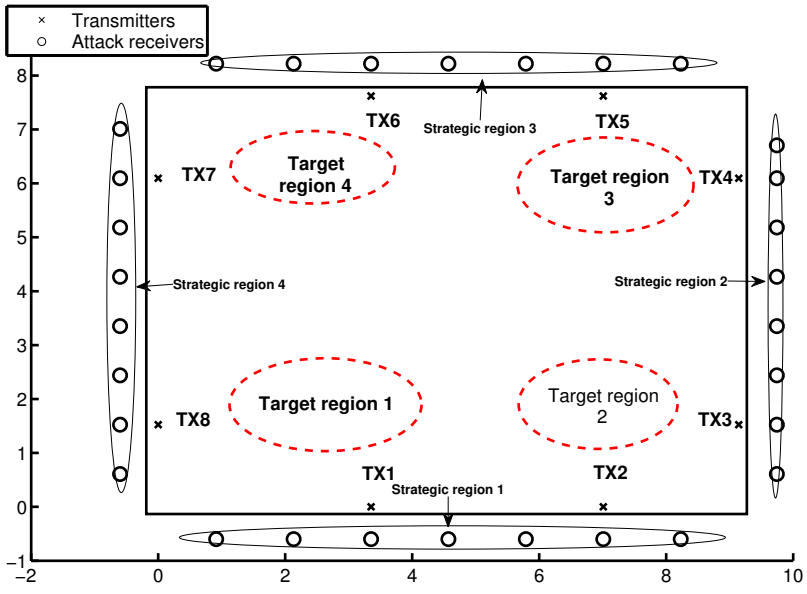

Figure 5.4. The experiment layout for open environment.

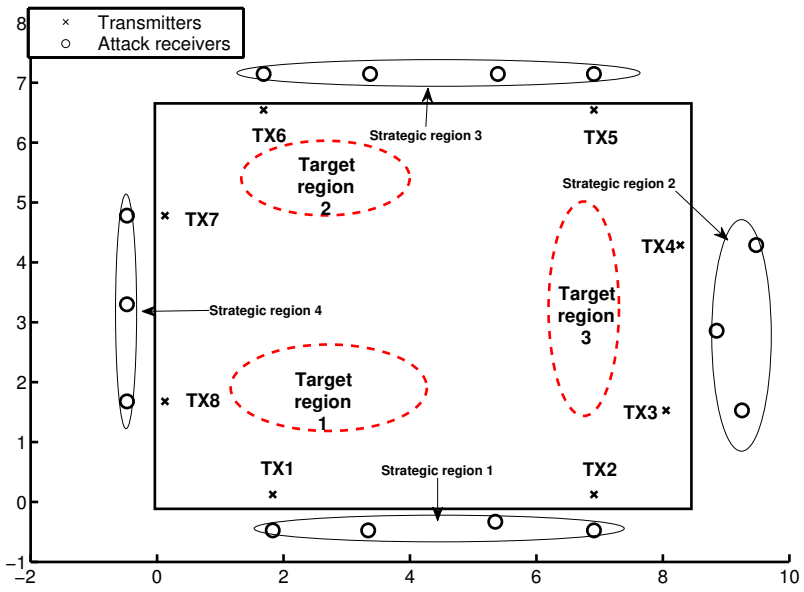

Figure 5.5. The experiment layout for cluttered office.

CHAPTER 6

COMPUTATION OFFLOADING IN MOBILE CLOUD COMPUTING

6.1 Introduction

Personal mobile devices such as smartphones and tablets are increasingly being used for in our daily lives. Many more smartphones are being sold worldwide than the total sales of PCs [93] and this growth in sales of smartphones is expected to continue in the future. Furthermore, various advances in technology are making these devices powerful tools capable of performing complex tasks, including speech to text conversion, audio identification, and image recognition. However, despite these advances, resources on mobile devices are constrained by weight and size requirements of the device that must be met for the devices to be easily carried around. Therefore, mobile devices still have limited battery, storage, heat dissipation ability, etc; which impede complex and resource intensive task execution. A possible remedy for tackling resource limitations of mobile devices is to offload the computational tasks to the cloud.

Mobile Cloud Computing (MCC) generally refers to a client-server communication model where a mobile device (client) offloads computing tasks to the remote cloud through a wireless network (mainly cellular or WiFi networks) [94, 95]. However, this model of mobile cloud computing is facing some important challenges. First, the performance of MCC is highly depend on wireless communication networks. With the tremendous growth of mobile data users, the wide-area mobile data access links (e.g., backhaul links in a cellular network) are becoming a bottleneck. This trend is expected to continue at even higher scales in the future because service providers, specifically cellular service providers, are unable to upgrade their backhaul networks due to shrinking profits. Thus, the MCC model can suffer from high latency and slow data transfer which may not be acceptable for the users of mobile applications. Second, although the cloud provides shared resources and

amortizes costs, its operation requires establishment and maintenance of highly expensive hardware to run the high computational tasks.

A possible remedy for tackling resource limitations of mobile devices when performing complex tasks, while not requiring the use of a server cloud and also minimizing latencies, is to offload computing tasks to nearby mobile devices. For example, in the speech to text conversion application, a mobile device can divide the audio file into smaller pieces, then assign each piece to a nearby device, and finally, combine the results obtained from nearby devices. Offloading compute tasks to nearby mobile devices rather than using a remote cloud through the mobile data cellular network lowers the latency and the burden on network backhaul. The nearby mobile devices, collectively and opportunistically, essentially provide the power of a cloud. Thus, we have another notion of mobile cloud computing that corresponds to offloading of compute tasks to a group of nearby mobile devices connected by various types of links including D2D, WiFi Direct, Bluetooth, etc. In this chapter, we use this second notion of mobile cloud computing.

There is a growing amount of work to utilize mobile device computing power for cloud computing. Hyrax [13] uses the computational power of a network of Android smartphones in MapReduce. Mobile Device Clouds [14, 15] and Serendipity [16] are platforms for opportunistic computing where a mobile device offloads computing tasks to nearby mobile devices. NativeBOINC for Android [17] is another example of utilizing mobile devices' computing power. Recently, Habak et al. [18] proposed FemtoCloud where a controller executes a variety of tasks arriving at controller by using the computational power of nearby mobile devices. SymbIoT [19] is another platform that uses the computational capability of all mobile devices within the same network to perform different tasks.

In this chapter, we examine a scenario where a mobile device or a central controller, which we call a *distributor* node, has a computational job or a set of different computational jobs and wants to utilize resources of nearby mobile nodes to reduce the job completion time. Due to mobility, the distributor (mobile device or a central controller) has frequent contacts with other mobile devices that can provide the required computational resources. The problem faced by the distributor is how to select the nearby nodes and divide the job among them in a manner that is beneficial to all the parties involved.

While the problem of using other nodes for executing the tasks of a job has been widely

used in distributed computing [96, 97], cyber foraging [98], and *crowdsourcing* [99], the existing work on task allocation cannot be simply adopted for mobile cloud computing. First, the task allocation method must take into account the selfish behavior of mobile nodes by providing incentives for them. This because in mobile cloud computing, a rational mobile node would not be willing to lend its resources (and thereby deplete its battery) unless it receives some payoff as compensation. Second, mobile devices can have different hardware/software and thus have different capabilities. For this reason, the execution time of a specific task can be different across mobile devices. Therefore, the task allocation mechanism must consider the heterogeneity of mobile devices to reduce the overall job completion time. Third, the task allocation needs to take into account the mobility of nodes. In a mobile environment, the distributor may observe disconnection of nodes with assigned tasks, and new arrivals that might provide high computational capabilities. Thus, decisions should be made according to the dynamics of the environment. Fourth, and very importantly, the distributor node should see a clear benefit in terms of job completion time.

We propose a game theoretic framework for task allocation that provides incentive for all mobile nodes. In our framework, the distributor node holds a multidimensional auction for allocating the tasks of a job among nearby mobile nodes based on their computational capabilities and also the cost of computation at these nodes, with the goal of reducing the overall job completion time. To the best of our knowledge, this is the first work that presents a multidimensional auction for task allocation in mobile cloud computing. Our proposed auction also has desired economic properties (that we formally prove later in the chapter) including *incentive compatibility* which ensures that players truthfully reveal their capabilities and costs, and that mobile nodes act cooperatively in the proposed auction for the benefit of all the parties involved. We also consider the mobility of mobile nodes in our game theoretic framework. In such a mobile environment, the topology of the network may change over time. Thus, some nearby nodes may get disconnected from the distributor before completing the assigned task, resulting in an increased job completion time. To deal with mobility, we perform multiple auctions over adaptive time intervals. We develop a heuristic approach to dynamically find the best time intervals between auctions to minimize unnecessary auctions and the accompanying overheads. We briefly explore the privacy of the distributor and the nearby mobile devices and show a tradeoff between

providing privacy, and the profits for the parties involved. Finally, we apply the incentive model for the application of mobile live video streaming [100].

We evaluate our framework and methods using both real world and synthetic mobility traces. We use two models of compute jobs: a simple single job model, and a multiple job model that uses a Directed Acyclic graph to represent causal dependencies in a set of jobs. Our evaluation results show that our game theoretic framework improves the job completion time by a factor of 2-5 in comparison to the local execution of the job, in both the job models, while minimizing the number of auctions. Thus, our approach is beneficial for the distributor in terms of enhancing its performance. We also show that the nearby nodes that execute the distributor's tasks receive a compensation higher than their actual costs.

The rest of this chapter is organized as follows. Section 6.2 contains the relevant related work. Section 6.3 explains the proposed method in detail. In Section 6.4, we extend the proposed game theoretic framework by considering the mobility of nodes and present the heuristic approach for finding the time interval between auctions. Section 6.5 explores the privacy of distributor and the nearby mobile devices. In Section 6.6, we describe experiments we conduct and our evaluation results. Section 6.7 explains the incentive model for the application of mobile live video upstreaming. Finally, we provide our concluding remarks in Section 6.8.

6.2 Related Work

Many existing works in distributed computing (e.g., SETI@ Home [96], BOINC [97], and cyber foraging [98]) have proposed using other nodes for executing the tasks of a job. However, all of these existing works primarily assume altruistic behavior in the distributed computing environment and do not carefully incentivize resource sharing. Like our work, Serendipity [16] enables remote computing among a set of intermittently connected mobile devices. However, our work differs from Serendipity in the following significant ways. First and foremost, Serendipity does not incentivize resource sharing. Second, it does not consider heterogeneity among mobile devices in task allocation. Moreover, all assigned tasks are assumed to have equal workload. Third, Serendipity does not use any adaptive methods for reassigning tasks.

Our game theoretic framework is inspired by the multidimensional mechanism proposed for the second score auction [101], where the authors use a linear function to map the multidimensional bid into a single dimension. However, we use a fractional function for the mapping which is more suitable to our setting (see equation 6.1). We also extend the existing multidimensional mechanism by allowing selection of k items while considering a budget limit instead of selection of only one item. The use of multidimensional auction allows us to consider the heterogeneity of nearby resources in task allocation. We also minimize the number of auctions and the accompanying overhead by developing a heuristic approach for finding the best time intervals between auctions.

Existing incentive mechanisms that have been used for task allocation [102, 103] do not consider multidimensional auction where both cost and the service quotient are important. The authors in [104] proposed a multidimensional optimal auction to provide incentive in mobile ad hoc network routing, considering both cost and the path duration in route selection. However, their proposed approach only works for time related bids such as path duration where the players cannot over-report the time related bid. In our task allocation problem, players can over-report and under-report both cost and the committed service quotient.

6.3 Problem Formulation and Solution

We consider mobile cloud computing in the presence of selfish smart phones. We assume that all smart phones act rationally and selfishly, and their main goal is to maximize their own profits, not to harm others. There is a distributor node that wants to offload a computational job, with a total work load of D units, to its nearby nodes with the goal of reducing the overall job completion time. Since a smart phone incurs a cost (in the form of resource and battery usage) while performing a task on behalf of the distributor, it may not be willing to participate unless the distributor provides the right incentives.

We propose a multidimensional auction to model the job/ task allocation in mobile cloud computing. In our auction, the distributor holds the auction among n nearby smart phones called *players*. Each player i has an individual private value t_i called its *type* which consists of the following two parameters:

- s_i: the committed service quotient that player i can provide. In distributed comput-

ing applications such as speech to text conversion, the service quotient is related to the amount of data that the smart phone can process in a given time (the smart phone finds this information by estimating its maximum execution time running synthetic benchmark and its energy consumption using techniques like PowerBooter [105]). In our system, s_i denotes the amount of data that player i can process per second. It should be noted that the distributor is not aware of the actual s_is until the tasks are completed and the results are received from nearby players who have been assigned the tasks.

- c_i: the cost of player i for performing the task with the committed service quotient. c_i is a function of the committed service quotient, s_i, and the player's private cost of performing the task, θ_i. We bound θ_i such that $0 < \theta_{min} \leq \theta_{max} < \infty$. θ_i is affected by various parameters such as processor speed, available storage, remaining battery level, communication cost, etc. c_i is an increasing function of both s_i and θ_i, and is private information of the player i. Hence, no one else can determine the exact value of c_i.

Our mechanism works as follows. First, the distributor sends a probe message to find the nearby nodes. Then, each nearby node i, interested in participating, replies by announcing its bid (c_i, s_i). Note that (c_i, s_i) announced by the node i need not be the actual private value of its type, t_i. Based on the received bids, the distributor selects a set of players and assigns the tasks to them in proportion to their committed service quotient. To provide incentive to the players for their resources, the distributor compensates them by paying the players. The distributor also has incentive to offload tasks to nearby mobile devices because the overall job completion time is less than the time taken for executing the job locally. Therefore, our game theoretical framework is beneficial to all parties involved.

Our mechanism implements truthfulness in *dominant equilibrium* implying that each player's best strategy, regardless of other players' strategies, is to bid truthfully (i.e. to report actual values of s_i and c_i). Table 6.1 summarizes the notation we use in this and the subsequent sections.

6.3.1 Allocation Mechanism

In this section, we specify how the distributor selects the nearby mobile devices by considering both the cost and the service quotient. The distributor wants players with minimum cost and maximum service quotient. For this reason, the distributor defines a weights function for player i with type (c_i, s_i) as follows:

$$w_i = \frac{s_i}{c_i} \tag{6.1}$$

The distributor must also determine how many players it should select. Intuitively, the number of players affects the job completion time and the sum of premiums paid by the distributor to the players over the players' actual costs (*overpayments*). As the number of selected players is increased, the job completion time decreases due to more tasks being executed in parallel. However, the increase in number of selected players also increases the overpayments.

To limit its cost, the distributor defines a budget limit B as the maximum amount that it can pay for the processing of one unit of data per second. B is a function of the distributor utility, u_d. An increase in the distributor utility, u_d, makes the distributor willing to increase its budget limit. Formally,

$$B(u_d) = \begin{cases} f^+(u_d^{max}) & u_d > u_d^{max} \\ f^+(u_d) & u_d^{min} \leq u_d \leq u_d^{max} \\ 0 & u_d < u_d^{min} \end{cases} \tag{6.2}$$

Here, u_d^{min} and u_d^{max} are the minimum and the maximum utilities that the distributor expects when using our task allocation framework. When the obtained utility, u_d, is less than u_d^{min}, the distributor prefers to execute the job locally. By increasing the value of u_d between u_d^{min} and u_d^{max}, the distributor is willing to use the task allocation framework. In this case, the budget limit is a nondecreasing function of u_d ($f^+(u_d)$ represents the nondecreasing function of u_d). u_d^{max} is the saturation point. Increasing the value of u_d beyond this saturation point does not increase the budget limit, B. The distributor determines u_d^{min}, u_d^{max}, and $f^+(u_d)$ depending on the application. For example, the distributor may prefer to choose a constant function for $f^+(u_d)$, if the distributor utility, u_d, be above u_d^{min} or the distributor may select an increasing exponential function for $f^+(u_d)$ in real time applications where time is critical.

Let t_O represent the job completion time when the distributor uses our task allocation framework, and let t_L represent the job completion time when the distributor executes the job locally. Also, let e_O and e_L be the energy consumption when the distributor using our task allocation framework and when the distributor executing the job locally. Then, the distributor utility, u_d, will be:

$$u_d = \alpha(t_L - t_O) + \beta(e_L - e_O) \tag{6.3}$$

The values of α and β determine the importance of job completion time and energy consumption in the distributor utility. In our evaluation, the distributor only wants to reduce the job completion time and ignores the energy consumption (we set $\alpha = 1$ and $\beta = 0$). We ignore the energy consumption for the following reasons. First, our task allocation is designed for high computational tasks such as audio identification and image recognition. In this case, the energy consumption for executing task, e_L, is much greater than the energy consumption for communication and transferring data among nearby mobile nodes, e_O. Second, in our task allocation framework, a computing cloud refers to a group of nearby mobile devices that connect by WiFi and Bluetooth. Thus, the energy consumption for communication and data transferring is less than the common client-server cloud.

Also, note that the distributor computes t_O by considering the preprocessing time, the overhead of dividing job into tasks, the postprocessing overhead of assembling the results, and the maximum job completion time among the selected nearby mobile nodes (the distributor can estimate the job completion time in each nearby mobile user by using their declared service quotients).

Now the goal is to select a subset of players that maximize the total weight under the budget limit constraint. We use a simple and efficient Greedy approach for allocation. First, the distributor orders the players based on their decreasing weights. Next, it selects the largest number of players $\{1, 2, 3, \ldots, k\}$ that satisfy the budget limit constraint (i.e., constraint 6.9). By considering the budget limit constraint, the distributor obtains at least the minimum utility, u_d^{min}. Therefore, participating in the task allocation is beneficial for the distributor. In the next section, we explain the payment function that the distributor pays to the nearby mobile users to provide incentives for them to participate in the task allocation.

6.3.2 Payment Mechanism

In this section, we determine the payment Mechanism. The payment Mechanism must provide the following desirable economic properties to ensure that players act cooperatively and bid truthfully.

- *Individual Rationality*: The utility of all players should always be nonnegative. Otherwise, players may choose to not participate.
- *Incentive Compatibility*: In an incentive compatible mechanism, no selfish node has incentive to lie (also called truthfulness).

Let $a_{c_i,s_i} \in \{0,1\}$ denote the allocation to player i with type (c_i, s_i) where if the distributor offloads a task to the player, $a_{c_i,s_i} = 1$, otherwise, $a_{c_i,s_i} = 0$. Also, let $p_i(a_{c_i,s_i})$ be the payment that the distributor pays to the player i under allocation rule a_{c_i,s_i}. Then, the utility of player i with type (c_i, s_i) is obtained from the following formula:

$$u_i(c_i, s_i) = p_i(a_{c_i,s_i}) - c_i \tag{6.4}$$

Having determined the utility of players, the payment Mechanism must satisfy the following constraints.

$$\forall i, c_i', p_i(a_{c_i,s_i}) - c_i \geq p_i(a_{c_i',s_i}) - c_i \tag{6.5}$$

$$\forall i, s_i', p_i(a_{c_i,s_i}) - c_i \geq p_i(a_{c_i,s_i'}) - c_i \tag{6.6}$$

$$\forall i, c_i', s_i', p_i(a_{c_i,s_i}) - c_i \geq p_i(a_{c_i',s_i'}) - c_i \tag{6.7}$$

$$\forall i, p_i(a_{c_i,s_i}) - c_i \geq 0 \tag{6.8}$$

$$\sum_{i=1}^{n} \frac{p_i(a_{c_i,s_i})}{s_i} \leq B(u_d) \tag{6.9}$$

Constraints 6.5, 6.6, and 6.7 provide incentive compatibility for both cost and the committed service quotient. Constraint 6.8 is for individual rationality, and constraint 6.9 captures the budget limit of the distributor.

The payment to player i consists of two parts:

$$p_i(a_{c_i,s_i}) = \begin{cases} p_i^1(a_{c_i,s_i}) + p_i^2(a_{c_i,s_i}) & i \leq k \\ 0 & i > k \end{cases} \tag{6.10}$$

where $p_i^1(a_{c_i,s_i})$ is paid to provide incentive compatibility for c_i and $p_i^2(a_{c_i,s_i})$ is paid to provide incentive compatibility for both c_i and s_i. p_i^1 is obtained from the following formula:

$$p_i^1(a_{c_i,s_i}) = \frac{s_i c_{k+1}}{s_{k+1}} \tag{6.11}$$

Here, $k+1$ is the index of the player with largest weight after the selected k players.

Let s_i be the actual service quotient (e.g., the amount of data that player i can process per second) that the distributor finds after the task completion by player i, and also let s'_i be the announced service quotient by the same player. Then, the second payment, p_i^2, is of the following form.

$$p_i^2(a_{c_i,s_i}) = d(s_i, s'_i)(s_i - s'_i) \tag{6.12}$$

Here, $d(s_i, s'_i)$ is a positive and nondecreasing function of $(s_i - s'_i) < 0$, that determines the impact of misreporting the service quotient. A suitable choice for the function $d(s_i, s'_i)$ must satisfy the following three properties. First, if the player i over-reports its committed service quotient ($s_i < s'_i$), then $p_i^2(a_{c_i,s_i})$ must be negative, i.e., the player must give back some money to the distributor. Second, a player must not pay a penalty for under-reporting its committed service quotient, i.e., when $s_i > s'_i$, $d(s_i, s'_i) = 0$. Third, the choice of $d(s_i, s'_i)$ should lend itself to satisfying the incentive compatibility property. In this chapter, we define $d(s_i, s'_i)$ as follows:

$$d(s_i, s'_i) = \begin{cases} 0 & s_i \geq s'_i \\ (s'_i - s_i)^2 + \frac{c_{k+1}}{s_{k+1}} & s_i < s'_i \end{cases} \tag{6.13}$$

This definition of $d(s_i, s'_i)$ satisfies the first two required properties described above. We show that this choice of $d(s_i, s'_i)$ helps satisfy the incentive compatibility property in Lemma 6.2. Figure 6.1 shows an instantiation of $d(s_i, s'_i)$. In this figure, the actual value of the service quotient is 0.9 Mbytes/sec. As shown in the figure, the value of $d(s_i, s'_i)$ increases if the player declares its committed service quotient (s'_i) to be greater than its actual service quotient. Also, the value of $d(s_i, s'_i)$ is zero if the player under-reports its committed service quotient.

Note that other choices of $d(s_i, s'_i)$ are also possible. Depending on the application, we can choose different functions for $d(s_i, s'_i)$. For example, in some applications the value of s_i might vary with changes in the environment that are not within the of control of the player. To provide incentive compatibility and to also prevent nonessential punishment, we can also define $d(s_i, s'_i)$ as follows:

$$d(s_i, s'_i) = \begin{cases} 0 & s_i \geq s'_i \\ \frac{c_{k+1}}{s_{k+1}} & s_i < s'_i \end{cases} \tag{6.14}$$

In this case, if the player over-reports its committed service quotient ($s_i < s'_i$), then the payment is:

$$p_i(a_{c_i,s'_i}) = \frac{s'_i c_{k+1}}{s_{k+1}} + (s_i - s'_i)\frac{c_{k+1}}{s_{k+1}} = \frac{c_{k+1}}{s_{k+1}} s_i = p_i(a_{c_i,s_i})$$

This payment is the same as the payment if the player wants to act truthfully. Therefore, the player has no incentive to lie. On the other hand, if the committed service quotient reduces after the player declares it, the player is paid only for the service quotient it accomplishes. We can also add a small reward if the player under-reports its committed service quotient ($s_i > s'_i$). The amount of reward should be less than the amount of payment when the player acts truthfully ($s_i = s'_i$). The player has no incentive to under-report its committed service quotient. However, if the committed service quotient increases after the player declares it, the player has an added incentive to provide a better service quotient.

Having determined the allocation rule and the payment policy, we must now prove the economic properties of our multidimensional auction, namely individual rationality and incentive compatibility using the definition of $d(s_i, s'_i)$ in equation 6.13. The same reasoning can be applied to other definitions of $d(s_i, s'_i)$, e.g., in equation 6.14.

6.3.3 Proofs

Players may cheat about their types, (c, s), to gain extra profit. Incentive compatibility (IC) ensures that players truthfully reveal their actual types. In a multidimensional mechanism, proving IC is challenging. This is because several cheating scenarios, formed from combinations of cheating in each dimension, must be considered. We must prove IC in the following conditions.

1. The truthful revelation of a player's cost (c) is a dominant equilibrium, given that the player reveals its committed service quotient (s) truthfully (constraint 6.5).
2. The truthful revelation of a player's committed service quotient (s) is a dominant equilibrium, given that the player reveals its cost (c) truthfully (constraint 6.6).
3. The truthful revelation of a player's cost and committed service quotient (c, s) is a dominant equilibrium (constraint 6.7).

Lemma 6.1. *Given the committed service quotient (s), truthfully revealing the cost (c) results in a dominant equilibrium.*

Proof. There are two possible cases:

- *Player is one of the winners.* If by over-reporting or under-reporting, the player still remains a winner, the utility of the player does not change. If the player becomes a loser by over-reporting, then the utility of the player becomes zero which is less than that it can obtain by acting truthfully. Therefore, in this case, the player has no incentive to lie.
- *Player is one of the losers.* If by over-reporting or under-reporting the player remains a loser, the utility of the player dose not change (it is still zero). On the other hand, if a player, (without loss of generality) $k+1$, under-reports the value of c_{k+1} to become a winner (i.e., $\frac{s_{k+1}}{c'_{k+1}} > \frac{s_k}{c_k}$), its utility becomes:

$$u_{k+1}(c'_{k+1}, s_{k+1}) = \frac{s_{k+1}c_k}{s_k} - c_{k+1} \tag{6.15}$$

  However, we know that $\frac{s_k}{c_k} > \frac{s_{k+1}}{c_{k+1}}$. This means that $c_{k+1} > \frac{s_{k+1}c_k}{s_k}$. Therefore, the utility is negative and the player has no incentive to lie.

Lemma 6.2. *Given the cost (c), truthfully revealing the committed service quotient (s) results in a dominant equilibrium.*

Proof. We show that the utility of the player has its maximum value at $s'_i = s_i$. Thus, the player has no interest to misreport s.

By substituting p^1 and p^2 in equation 6.4 and taking derivative with respect to s'_i,

$$\frac{\partial u_i}{\partial s'_i} = \frac{c_{k+1}}{s_{k+1}} + \frac{\partial d(s_i, s'_i)}{\partial s'_i}(s_i - s'_i) - d(s_i, s'_i)$$

For $s'_i = s_i$, $\frac{\partial u_i}{\partial s'_i} = 0$. For $s'_i \in (s_i, +\infty]$, given that $\frac{\partial d(s_i, s'_i)}{\partial s'_i}$ is positive, $\frac{\partial u_i}{\partial s'_i}$ is negative. Also, for $s'_i \in (0, s_i]$, $\frac{\partial u_i}{\partial s'_i}$ is positive. As a result, $s'_i = s_i$ is the maximum point. Figure 6.2 shows the utility of a player for different values of s'_i. This figure also shows that the utility is maximum at point $s'_i = s_i$.

Lemma 6.3. *Truthfully revealing both cost and the committed service quotient (c, s) results in a dominant equilibrium.*

Proof. Let c' and s' be the declared cost and the committed service quotient of a player. Also, let c, and s be the actual cost and performed service quotient. Then, to prove incentive compatibility, we should consider the following four cases.

1. $c' < c$ *and* $s' < s$. If the player is already a winner and also wins by misreporting, then the utility of the winner will be less than that when acting truthfully.
$$u_i(c'_i, s'_i) = \frac{s'_i c_{k+1}}{s_{k+1}} - c_i$$
Given that $s'_i < s_i$, $u_i(c_i, s_i) > u_i(c'_i, s'_i)$ and the player has no incentive to lie. If the player is not a winner, but wins the game by misreporting, then the utility of the player becomes negative (proof is similar to the second case of Lemma 6.1).
2. $c' > c$ *and* $s' < s$. If the player is already a winner and still wins the game by misreporting, the utility of the player will decrease. On the other hand, if the player is not a winner, then it cannot win by misreporting. Thus, its utility does not change.
3. $c' < c$ *and* $s' > s$. If the player is already a winner and remains a winner by misreporting, then the utility of the player is as follows.
$$u_i(c'_i, s'_i) = \frac{s'_i c_{k+1}}{s_{k+1}} - c_i + (s_i - s'_i)(\frac{c_{k+1}}{s_{k+1}} + (s'_i - s_i)^2) \tag{6.16}$$
By substituting s'_i with $s_i + (s'_i - s_i)$, we have
$$u_i(c'_i, s'_i) = \frac{s_i c_{k+1}}{s_{k+1}} - c_i + (s_i - s'_i)(s'_i - s_i)^2$$
which is less than the case where the player acts truthfully. If the player is not a winner, but wins the game by misreporting, its utility becomes
$$u_{k+1}(c'_{k+1}, s'_{k+1}) = \frac{s_{k+1} c_k}{s_k} - c_{k+1} + \big((s'_{k+1} - s_{k+1})^2\big)$$
$$(s_{k+1} - s'_{k+1})$$
The first term $\frac{s_{k+1} c_k}{s_k} - c_{k+1}$ is negative, because $\frac{s_k}{c_k} > \frac{s_{k+1}}{c_{k+1}}$. The second term is also negative, $s_{k+1} < s'_{k+1}$. Thus the utility is negative in this case.
4. $c' > c$ *and* $s' > s$. If the player is already a winner and still wins by misreporting, then the utility of player decreases. Also, if the player is not a winner, but wins the game by misreporting, then its utility becomes negative.

We must now prove that the proposed multidimensional auction is individually rational. This property ensures that players participate in the task allocation game.

Lemma 6.4. *The proposed multidimensional auction satisfies the individual rationality constraint (constraint 6.8).*

Proof. We need to show that when players act truthfully, their utilities are greater than or equal to zero. If the player is not a winner, then its utility is zero. If the player is a winner then its utility is determined as follows.

$$u_i(c_i, s_i) = \frac{s_i c_{k+1}}{s_{k+1}} - c_i$$

Given that $\frac{s_i}{c_i} > \frac{s_{k+1}}{c_{k+1}}$, the utility of the winner is positive.

6.3.4 Task Allocation

We now propose the task allocation policy among the k selected players to reduce the job completion time using different models of compute jobs.

Single job: In some applications such as audio to text conversion, we only have one job. In such a single job scenario, the workload assigned to each player is proportional to its committed service quotient (in the mobility aware approach in the next section, we also consider the auction time interval in the task allocation towards ensuring that the selected nodes are able to complete the assigned task before the distributor performs the next auction). For example, for player i the amount of workload is equal to $D\frac{s_i}{\sum_{i=1}^{k} s_i}$, where D is the total workload. In this case, all selected players execute the assigned tasks in parallel which reduces the job completion time. Any other task allocation reduces the number of tasks executing in parallel and consequently increases the job completion time.

DAG jobs: There are some applications that contain a set of jobs where the execution of jobs has causal ordering. We use a Directed Acyclic Graph (DAG) to represent causal dependencies in a set of jobs. In our DAG, nodes correspond to jobs and directed links represent the causal dependencies. Figure 6.3 shows an example of a DAG representing a set of jobs.

We can use the same task allocation as the single job for DAG jobs. For dependent jobs, we execute jobs sequentially and assign workloads to the selected k players in proportion to their committed service quotients. For independent jobs, we can run them in parallel. For example, job B and C in Figure 6.3 are independent and can be run in parallel. Let D_1, and D_2 be the total workloads of jobs B, and C, respectively. Then the assigned workload to player i is $D_1\frac{s_i}{\sum_{i=1}^{k} s_i} + D_2\frac{s_i}{\sum_{i=1}^{k} s_i}$.

Indivisible jobs: In some applications, jobs are indivisible. Even for indivisible jobs, the proposed game theoretic framework can be used to provide incentives, utilize nearby

resources, and reduce the job completion time.

For a single indivisible job, the distributor selects only one nearby device and assigns the whole job if the job execution time is less than the time taken in executing the job locally. For indivisible DAG jobs, the distributor acts in the same manner as it would for the single indivisible job scenario, for all dependent jobs and executes them sequentially. For all independent jobs, the distributor executes them in parallel. First, it selects k nearby smart phones where k is equal to the number of independent jobs, by considering the budget limit constraint. For task allocation, the distributor ranks the independent jobs in the following manner. For each independent job, the distributor computes the total remaining job load from the node corresponding to the independent job to the leaf of the DAG. Then, it assigns ranks to the independent jobs in decreasing order of the remaining load. Next, the highest rank job is assigned to the selected player with the highest value of committed service quotient. Let 50, 100, 100, 50 be the workload of jobs B, C, D, and E, respectively, in Figure 6.3. Then the total remaining job loads from B, and C are 200, and 150, respectively. Therefore, in allocating tasks, the distributor assigns C to selected player with the highest committed service quotient.

Although our game theoretic framework can be used for task allocation where jobs are indivisible, it is likely to have lower efficiency compared to local execution, in terms of job completion time. Specially, when we have computationally intensive indivisible jobs in an environment with high mobility (i.e., nearby mobile nodes are disconnected before completing the assigned task). Note that for divisible jobs, we deal with mobility by performing multiple auctions with dynamic time intervals (see section 6.4).

6.4 Mobility Aware Approach

So far, we have implicitly assumed that the mobile devices are available for the entire duration of the task computations. However, because of mobility of the distributor or the players, the topology of network may change and consequently some mobile smart phones may get disconnected before completing the assigned tasks. Furthermore, some other smart phones might arrive in the vicinity of the distributor with higher computational power and less cost. To deal with such device mobility, instead of holding the auction only once, we hold the auction multiple times. The key challenge in holding the auction

multiple times is the determination of the time interval between auctions.

When we perform the auction very frequently, we are able to find newly arriving smart phones. Note that the duration of the task allocated to the smart phones is limited by the time between auctions. Therefore, when the time between auctions is short, the chance that a smart phone gets disconnected before completing its assigned task will decrease. However, the number of auctions and the accompanying overheads will increase, since each time the distributor needs to probe to find the neighboring smart phones and their type values. On the other hand, if we hold the auction less frequently, then there is a higher chance that smart phones get disconnected before performing the assigned tasks. Also, we may lose potential computational power of newly arriving smart phones. As a result, we may see higher job completion times. Therefore, there is a trade-off between the overhead due to the number of times that we perform the auction and the job completion time.

Figure 6.4 shows the average percentage of disconnected nodes versus the time interval between two auctions (or distributor probes) for three different real world contact traces. The three real contact traces were collected as a part of the Haggle Project [106] at the Intel research lab, at the University of Cambridge Computer Laboratory, and during the *Infocom05* conference, respectively. This figure shows that in all three experiments, as we increase the time interval from 20 to 500 seconds, the average percentage of disconnected nodes increases, consequently increasing the job completion time in our task allocation mechanism.

Instead of using a fixed time interval between auctions, we propose an adaptive approach that modifies the time interval dynamically based on the mobility of the nodes in the environment. Our goal is to reduce the number of times we perform the auction without significantly increasing the job completion time.

We use AIMD (Additive Increase and Multiplicative Decrease) approach to adaptively set the time interval between two auctions. We use AIMD for two reasons. First, in real world scenarios, the change in the number of contacts has been shown to follow a power law distribution with a bursty traffic pattern [107], i.e., a large number of contacts arrive or leave a specific place over a short period of time. This implies that the time interval between two auctions, T, should decrease rapidly when a change is observed in the number of connections. Second, AIMD is a stable adaptive approach that is widely

used in networking protocols such as TCP.

We describe our adaptive approach as follows. We start with a fixed time interval, T. Then, we update the value of T by comparing the results from the previous auction. If no change in the number of connections is observed, we conservatively increase the value of T linearly (by a constant value of b seconds). Otherwise, we decrease the value of T multiplicatively by a factor of 2. Therefore, the value of T increases slowly when there is no change, and it decreases rapidly in the case of change in the number of connections from one auction to the next. More specifically, the value of T is obtained from the following formula.

$$T(i+1) = \begin{cases} \max(\frac{T(i)}{2}, T_{min}) & Nc_i \neq Nc_{i+1} \\ \min(T(i)+b, T_{max}) & \text{otherwise} \end{cases} \tag{6.17}$$

where, T_{min} and T_{max} are the minimum and the maximum values of T, respectively. Nc_i denotes the contact set at the beginning of time interval i.

6.5 Privacy

Another important issue in MCC is the privacy of data. In our game theoretic framework the distributor violates the privacy of data by offloading a computational job to the nearby mobile devices. Let $j = \{1, \ldots, m\}$ denote a set of computational jobs that the distributor wants to execute. We formulate the privacy risk for distributor, $PR(d)$, as the following:

$$PR(d) = \sum_{j=1}^{m} \rho_j v_j \tag{6.18}$$

$\rho_j \in [0, 1]$ in equation 6.18 determines the sensitivity of the computational job j. v_j represents the visibility of job j and is equal to the number of selected mobile devices by the distributor for executing tasks of job j.

As the above formulation shows, the privacy risk for the distributor, $PR(d)$, increases by increasing the number of selected mobile devices. However, the job completion time decreases as the number of selected mobile devices is increased. Figure 6.5 shows the tradeoff between $PR(d)$ and the job completion time. Note that the distributor can provide privacy by adding another constraint for privacy risk in both allocation and payment mechanism at the cost of increasing the job completion time.

Besides distributor, the nearby mobile devices also violate their privacy by revealing their computational capabilities and the cost of computation. To examine the impact of

privacy on the nearby mobile users' utilities, we define the privacy parameter $x \in [0,1]$ where the mobile device conceals the actual value of cost, c, and the actual value of service quotient, s, by a factor x. I.e, $c' = xc$ and $s' = \frac{1}{x}s$ (we only consider the case where the mobile device decreases the actual value of cost and increases the actual value of service quotient. For all the other cases, it is obvious that the utility of mobile device decreases by concealing the actual values of cost and service quotient). Figure 6.6 shows the CDF of nearby mobile devices' utilities for different values of x. This figure shows that the mobile devices can provide privacy at the cost of decreasing their utilities and the highest utility is obtained if the mobile devices act truthfully, $x = 1$, and reveal their actual costs and computational capabilities as we proved in Section 6.3.3.

6.6 Evaluation

6.6.1 Experimental Setup

To evaluate our task allocation method and its overhead, we use a custom simulator that we develop using Matlab to simulate the speech to text conversion application under different mobility conditions. Instead of performing the conversion at the distributor, the audio file of the speech is divided into smaller files and nearby smart phones are used to convert the small audio files to text files and send back the results to the distributor who finally combines them.

In our simulation, each player chooses its cost of processing the audio file (c) from a uniform distribution in interval $[1,5]$. The size of of audio file that a player can process per second (s) is selected randomly from a uniform distribution $[50KB, 125KB]$. The distributor's budget limit function ($B(u_d)$) is a constant function 50 per megabyte of data for the distributor utility u_d to be positive (note that other functions are also possible). Also, T_{min}, T_{max}, that represent the maximum and minimum values for the time interval between auctions, are set to $20s$, and $500s$ respectively. Finally, the linear increase factor in the heuristic approach (b) is set to $20s$ in all experiments. All the results are obtained by averaging over 1000 randomly selected samples.

We evaluate our task allocation method for two scenarios. In our first scenario, we assume that the neighboring nodes are available during the whole computation. In our second scenario, we consider the mobility of nodes and the possibility of disconnection

during computations. To evaluate the mobility aware approach in the second scenario, we use three real contact traces with different contact properties. These traces were collected as a part of the Haggle Project [106]. The first data set was collected at Intel research lab, the second set was collected at the University of Cambridge Computer Laboratory, and the third set was collected during the Infocom05 conference. We select 9 nodes from the first set and 12 nodes from the second and third sets. We also use two synthetic mobility models namely the Slaw mobility model [107] and the Random Walk mobility model(RW) [90]. The Slaw model captures many of the statistical properties of human mobility, while the Random Walk model provides high mobility and abrupt changes in movement patterns [90].

6.6.2 Results

We evaluate our task allocation method in terms of both the job completion time and the overhead. To quantify the overhead, we measure the number of times that the auction is performed before task completion. We also use the speedup ratio metric to evaluate the speedup in the job completion time as a result of offloading tasks to other smart phones. The speedup ratio is obtained by dividing the job completion time in locally executing the job by the job completion time in our distributed approach. The speedup ratio shows the performance benefits where we provide the required incentives for the nearby mobile devices to lend their resources with the case where there is no incentive model and selfish nearby mobile devices would not be willing to lend their resources. In other words, the speedup ratio demonstrates the performance benefits of the distributor from using the game theoretic framework.

Figure 6.7 shows the speedup ratio for the first case where selected neighboring nodes are available during the whole computation. We use three audio files with sizes of $100MB$, $300MB$, and $600MB$. The number of neighboring nodes are randomly selected from a power law distribution. Figure 6.7 shows how our method significantly improves the job completion time in comparison to the time taken in locally executing the job. This figure shows that for all of our chosen file sizes, our method can improve the job completion time by about a factor of 5.

Table 6.2, Table 6.3, and Table 6.4 show the job completion time, number of performed

auctions, and the percentage of disconnected nodes in case of mobility for the three real contact traces. In this table, we compare the results of fixed time intervals with our heuristic approach for adaptively selecting the time intervals between auctions. In this experiment, the file size is set to $600MB$. We make the following observations. First, as the time interval between two auctions (T) increases or the number of performed auctions decreases, the job completion time and the percentage of disconnected nodes increase. By increasing the time interval, the percentage of disconnected nodes increases and the distributor also misses some of the newly arriving nodes. This reduces the number of tasks that are being executed in parallel, and consequently the job completion time increases. Second, the fixed time interval approach performs well in terms of job completion time for small values of T. For large values of T, this approach performs well in terms of overhead, i.e., the number of performed auctions. Thus, finding the right value for T where it performs well in terms of both the job completion time and the overhead depends on the mobility patterns of mobile nodes. Third, the heuristic approach performs well in terms of both the job completion time and the overhead without requiring any prior knowledge of the mobility patterns of mobile nodes.

Figure 6.11 compares the performance of our heuristic approach for finding time intervals between auctions with the fixed time intervals approach, when nodes are mobile. In this figure, the x-axis represents the normalized job completion time, and the y-axis represents the normalized number of performed auctions. Our goal is to reach to the left bottom corner of the figure where minimum job completion time and minimum overhead exist. Figure 6.11 shows that the heuristic approach outperforms the fixed intervals method and gets close to the left bottom corner.

Next, we compare the job completion time of the mobility aware approach with that of local execution using the three real world contact traces for different file sizes. Figure 6.8, Figure 6.9, and Figure 6.10 show the speedup ratio of the mobility aware approach using our heuristic approach. These figures show that the speedup ratio is close to 3. However, the amount of improvement in the Infocom05 contact trace is a less than the other contact traces. This is because the average percentage of disconnected nodes in the Infocom05 trace is larger than the other traces (see Figure 6.4).

To further analyze the effect of contact traces, we use two synthetic mobility models to

generate the contact traces. Specifically, we use the Slaw [107] and Random Walk mobility models. In our setting, we assume that 20 mobile nodes move in an area of $1000m \times 1000m$ for one hour and each mobile node has a circular transmission range of $150m$. Also, the audio file size is set to $600MB$.

Figure 6.12 shows the speedup ratio of mobility aware approach with fixed and heuristic approaches for selecting the time interval between auctions in the Slaw and Random Walk models. This figure shows that our proposed model can improve the job completion time by a factor of 5 to 7 in the Slaw model (which mimics the human walk patterns). However, the speedup ratio of the mobility aware approach is at most 3 in the Random Walk model. This is because, in the Random Walk model, mobile nodes continuously move from one location to the another in a random fashion. According to our measurements in the Random Walk model, more than half of the selected neighboring nodes get disconnected before completing their tasks, consequently increasing the job completion time. However, we can still improve the job completion time by a factor of 3 for small fixed time interval ($T = 20s$) at the cost of increased auction overhead. Furthermore, we can improve the job completion time by factor 2 using our heuristic approach without significantly increasing the communication overhead. According to our experiments, the number of performed auctions using the heuristic method and using a small fixed time interval ($T = 20s$), are 59 and 110, respectively. It should be noted that for a high mobility contact trace (such as Random Walk), there is a higher change of contact among mobile nodes. Therefore, we can still improve the job completion time. Also, it is worth mentioning that by reducing the time interval between auctions or equivalently reducing the workloads of the assigned tasks, we can further improve the speedup ratio, but at the cost of increased overhead.

To understand the impact of the number of nearby nodes, we vary the number of mobile nodes between 10 and 50 with increments of 10, in the Slaw model. Figure 6.13 shows the job completion time of the proposed approach using the Slaw mobility model as a contact trace. This figure shows that as the number of nodes increases, the job completion time decreases by more than 40%, when fixed time intervals (T) are used. Job completion time reduces from 1486 to 833 seconds for $T = 20$ seconds, from 1888 to 1055 seconds for $T = 200$ seconds, and from 2238 to 1324 for $T = 500s$. In addition, this figure shows that our heuristic approach performs well in terms of job completion time.

Figure 6.14 shows both the job completion time and the overhead (number of performed auctions) using the Slaw mobility model with different number of nodes. This figure shows that for different number of nodes in the Slaw mobility model, the heuristic approach performs significantly better than the fixed time intervals method, in terms of both the job completion time and the overhead.

Finally, we evaluate our methods for DAG jobs. The structure of jobs is the same as Figure 6.3 and the workloads of jobs A, B, C, D, and E are 100, 50, 100, 100, 50, respectively. Figure 6.15 shows the speedup ratio of the mobility aware approach for DAG jobs, using the heuristic method to select the time intervals. This figure shows that our proposed approach improves the job completion time by factor 2.5 to 3 for DAG jobs. DAG jobs have the same behavior as the single job by varying the number of nodes or using different mobility models.

6.7 Incentive Model for Mobile Live Video Upstreaming

mobiLivUp is a mobile live video upstreaming system that utilizes nearby smartphones, forwarders, and cellular bandwidth to improve live wide-area video upstreaming [100]. In mobiLivUp there is a splitter that distributes the data between forwarding nodes and there is a gatherer that combines the data together.

We propose a pricing based method that provides incentives for the forwarding nodes to cooperate. We assume that all nodes are rational and selfish. A forwarding node's main goal is to maximize its profits but not to harm others. We use a simple auction to model cooperative bandwidth sharing. In our auction, the splitter holds the auction among n nearby forwarding nodes called *players*. Each player i has an individual private value c_i which is the cost of sending one unit of data to the gatherer using the cellular connection. c_i depends on various parameters including available cellular bandwidth, cellular data rate, and battery level on the phone. Our auction works as follows.

The splitter sends a request to the forwarding nodes and the forwarding nodes reply by sending their bids, c_i, to the splitter. The splitter determines the allocation rule and the payment mechanism based on the received bids and the received feedback from the gatherer about the actual data rates of the forwarders.

Due to the changes in the actual data rates, the splitter holds an auction every time

it receives feedback from the gatherer. The forwarding nodes can also change their bids in each auction and it is possible that the number of forwarding nodes varies in different auctions because of the mobility of mobile phones.

Our auction provides incentive for forwarding nodes to cooperate by implementing a *dominant equilibrium*. In this setting, each forwarding node's best strategy is to report its actual cost c_i, regardless of other players' strategies. Each player's utility is defined as its total received payment minus its cost of participation.

6.7.1 Problem Formulation and Solution

In our auction, the splitter must consider both the cost and the actual data rate in selecting forwarding nodes. Let $w_i \in [0, 1]$ be the weight that is assigned to the player i based on the gatherer feedback. We define a score s_i for forwarding node i, as $s_i(c_i, w_i) = \frac{w_i}{c_i}$. Note that in this equation, s_i depends on both the cost c_i and the actual data rate w_i. The splitter uses these scores to select a set of forwarding nodes. We obtain the utility of player i with score s_i from the following formula: $u_i(s_i) = p_i(a_{s_i}) - c_i$.

The number of players that are selected by the splitters depends on a budget limit B, the maximum amount that the splitter can pay per second. Depending on the type of the service, the splitter may choose different values for B.[1]

The splitter determines the allocation and the payment based on the scores of forwarding nodes under the following conditions:

- *Optimal*. The mechanism should maximize the total score, i.e., total actual data rates divided by total costs.
- *Incentive Compatibility*. There is no selfish forwarding node that has an incentive to lie about the cost, c_i.
- *Individual Rationality*. The utility of all forwarding nodes should be nonnegative to provide incentive for them to participate in the game.

Our problem description is as follows:

---

[1]The budget limit, B can be increasing function of the total data rate, $\sum_{i=1}^{k} w_i$, where k is the number of selected players.

$$\max_{a} \sum_{i=1}^{n} a_{s_i} s_i$$

$$s.t.$$

$$\forall i, c_i' \in C, p_i(a_{s_i}) - c_i \geq p_i(a_{s_i'}) - c_i \tag{6.19}$$

$$\forall i, p_i(a_{s_i}) - c_i \geq 0 \tag{6.20}$$

$$\sum_{i=1}^{n} p_i(a_{s_i}) \leq B \tag{6.21}$$

Here, $a_{s_i} \in \{0, 1\}$ represents the allocation to player i with score s_i, when the splitter assigns data to the player, $a_{s_i} = 1$, otherwise, $a_{s_i} = 0$. Also, $p_i(a_{s_i})$ represents the amount that the splitter pays to the player i under allocation rule a_{s_i}. Equation 6.19 provides incentive compatibility for cost.[2] Equation 6.20 is for individual rationality, and Equation 6.21 captures the budget limit of the splitter. The splitter orders the forwarding nodes based on their scores decreasing scores (s_i). Then, it selects the largest number of forwarding nodes $\{1, 2, \ldots, k\}$ such that $\forall i \in \{1, 2, \ldots, k\}$, $c_i < \frac{Bw_i}{\sum_{i=1}^{k} w_i}$.

The payment, $p_i(a_{s_i})$, is obtained from the following formula:

$$p_i(a_{s_i}) = min(\frac{w_i c_{k+1}}{w_{k+1}}, \frac{Bw_i}{\sum_{i=1}^{k} w_i}) \tag{6.22}$$

Here, $k + 1$ is the index of the player with the largest score after the selected k players. In equation 6.22, the payment increases linearly with the forwarding node's actual data rate, w_i.

6.7.2 Proofs

Lemma 6.5. *Truthfully revealing the cost c results in a dominant equilibrium.*

Proof. There are two possible cases: if the player is one of the selected forwarding nodes, then over-reporting or under-reporting does not increase the utility of the player. One the other hand, if a player is one of the losers, (without loss of generality) $k + 1$, and by under-reporting the value of c_{k+1} the player $k + 1$ becomes a winner, then the utility of this player is:

$$u_{k+1}(s_{k+1}') = min(\frac{w_{k+1} c_k}{w_k} - c_{k+1}, \frac{Bw_{k+1}}{\sum_{i=1, i \neq k}^{k+1} w_i} - c_{k+1}) \tag{6.23}$$

[2]Since the data rate of each forwarding node is obtained from the gatherer feedback, we do not need to provide incentive compatibility for the data rate.

However, we know that $\frac{w_k}{c_k} > \frac{w_{k+1}}{c_{k+1}}$. Therefore, the utility is negative which is less than the time if the player acts truthfully.

Lemma 6.6. *The proposed auction satisfies the individual rationality constraint (constraint 6.20).*

Proof. According to equation 6.4 the utility of player i is determined as follows.

$$u_i(s_i) = min(\frac{w_i c_{k+1}}{w_{k+1}} - c_i, \frac{B w_i}{\sum_{i=1}^{k} w_i} - c_i)$$

Given that $\frac{w_i}{c_i} > \frac{w_{k+1}}{c_{k+1}}$ and $\frac{B w_i}{\sum_{i=1}^{k} w_i} > c_i$, the utility is positive.

6.8 Conclusion

In this chapter, we presented and evaluated a game theoretic framework for task allocation in mobile cloud computing environments comprising of selfish mobile devices. Specifically, we proposed a multidimensional auction for allocating the tasks of a job among nearby mobile nodes based on their computational capabilities and also the cost of computation at these nodes with the goal of reducing the overall job completion time and be beneficial to all the parties involved. We considered node and task heterogeneity as well as node mobility in developing our methods. Our evaluations demonstrated the benefits of our methods. We also proposed an incentive model to provide cooperation among nearby mobile devices for the application of mobile live video upstreaming.

Table 6.1. Notation

Parameter	Definition
(c_i, s_i)	The cost and committed service quotient for player i
a_{c_i,s_i}	Allocation rule for player i with type (c_i, s_i)
$u_i(c_i, s_i)$	Utility of player i with type (c_i, s_i)
$p(a_{c_i,s_i})$	Payment to player i under allocation rule a_{c_i,s_i}
B	The distributor budget limit
D	The total workload of job
T	Time interval between auctions
b	Linear increase factor in heuristic approach

Table 6.2. Completion time (second) of the mobility aware approach with fixed and adaptive time intervals between auctions.

Approaches	Completion time (second)		
	Intel research lab	Computer lab	Infocom
$T = 20s$	2109	2100	2632
$T = 200s$	2646	3021	3776
$T = 500s$	3288	3718	4725
Heuristic approach	2337	2294	2997

Table 6.3. Number of performed auctions in the mobility aware approach with fixed and adaptive time intervals between auctions.

Approaches	Number of performed auctions		
	Intel research lab	Computer lab	Infocom
$T = 20s$	119	119	153
$T = 200s$	15	18	23
$T = 500s$	7	10	12
Heuristic approach	19	33	43

Table 6.4. Percentage of disconnected nodes in the mobility aware approach with fixed and adaptive time intervals between auctions.

Approaches	Percentage of disconnected nodes		
	Intel research lab	Computer lab	Infocom
$T = 20s$	0.04	0.054	0.066
$T = 200s$	0.247	0.4	0.43
$T = 500s$	0.267	0.5	0.57
Heuristic approach	0.16	0.16	0.2

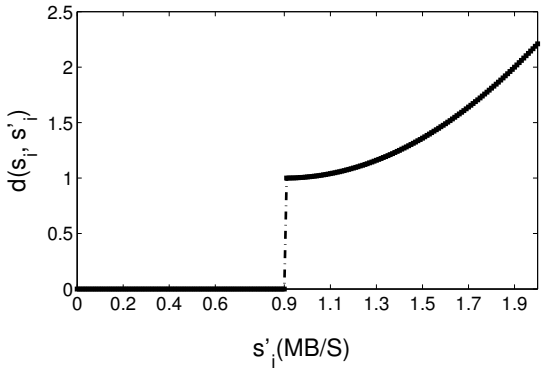

Figure 6.1. $d(s_i, s_i')$ for different values of s_i', with $s_i = 0.9$ Mbytes/sec.

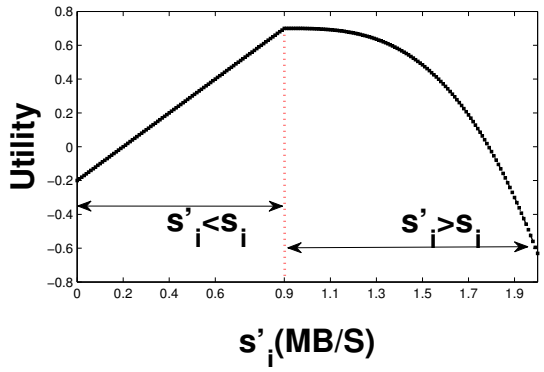

Figure 6.2. Utility for different values of s_i', with $s_i = 0.9$ Mbytes/sec.

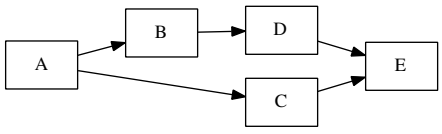

Figure 6.3. Example of DAG jobs.

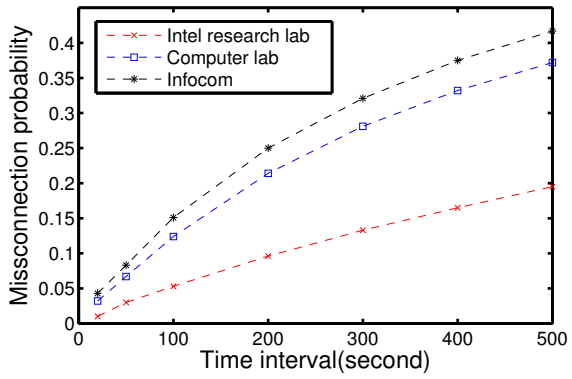

Figure 6.4. Average percentage of disconnected nodes versus the time interval between auctions.

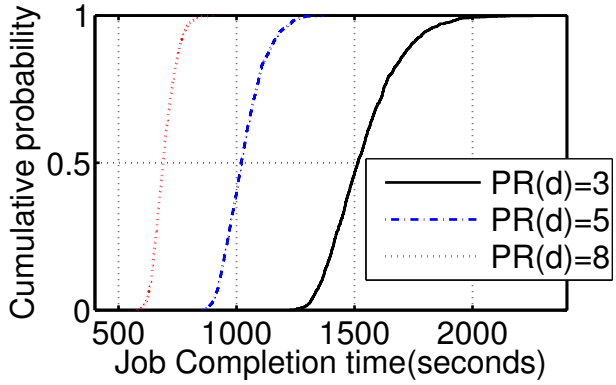

Figure 6.5. CDF of job completion time for different values $PR(d)$.

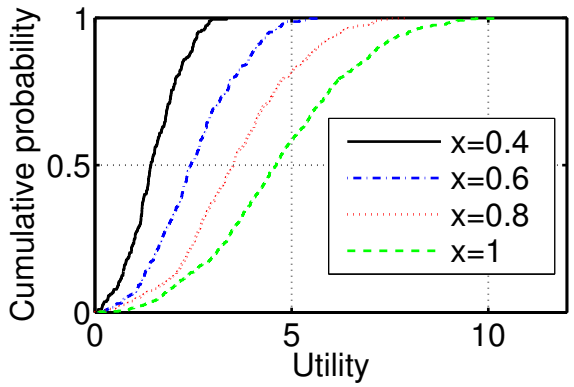

Figure 6.6. CDF of utility of mobile devices for different values of privacy parameter, x.

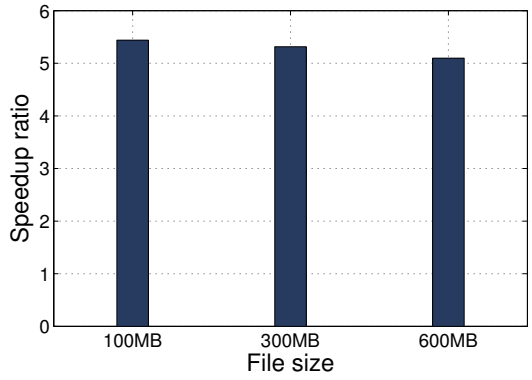

Figure 6.7. Speedup ratio without considering mobility for different file sizes.

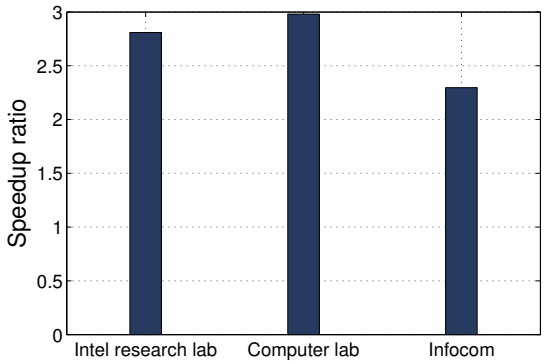

Figure 6.8. Speedup ratio of mobility aware approach using our heuristic to adaptively select the time interval between auctions for file size $100MB$.

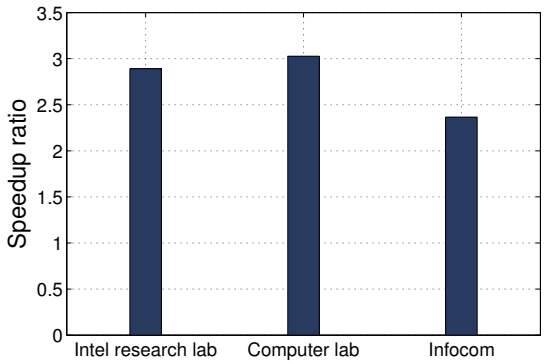

Figure 6.9. Speedup ratio of mobility aware approach using our heuristic to adaptively select the time interval between auctions for file size $300MB$.

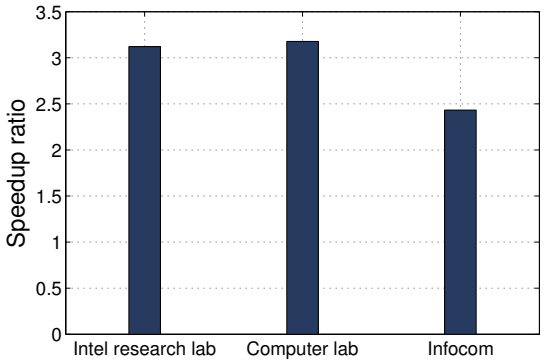

Figure 6.10. Speedup ratio of mobility aware approach using our heuristic to adaptively select the time interval between auctions for file size $600MB$.

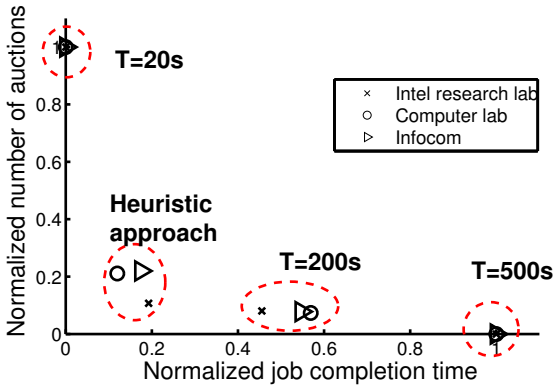

Figure 6.11. Comparison of approaches with fixed and heuristic-based time intervals between auctions, in the presence of mobility.

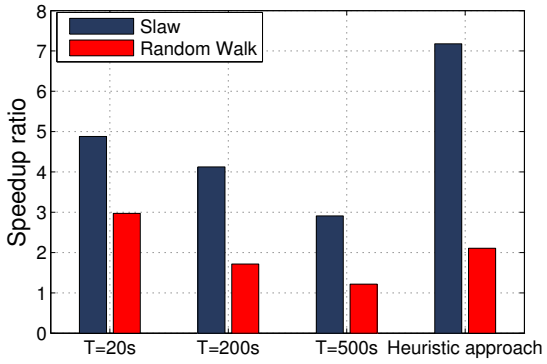

Figure 6.12. Speedup ratio of mobility aware approach using fixed and heuristic-based time intervals between auctions in Slaw and Random Walk models.

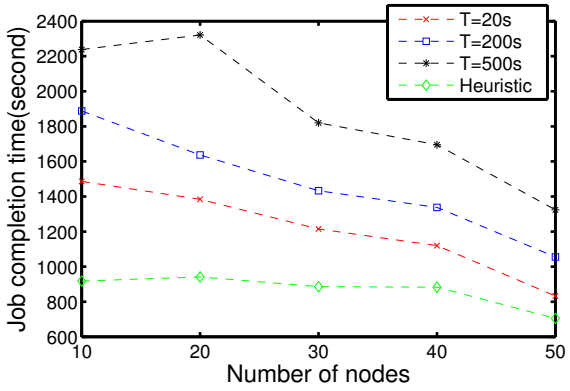

Figure 6.13. The job completion times for different values of T and the heuristic approach versus the number of nodes in the Slaw mobility model.

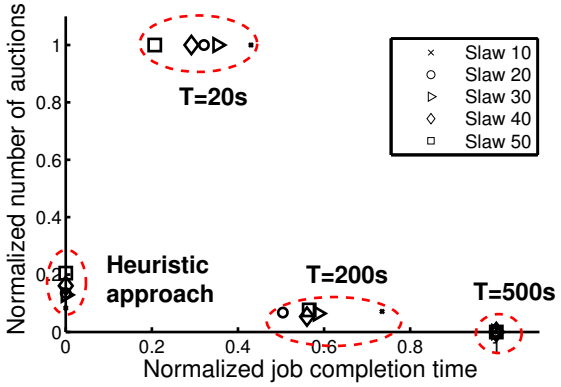

Figure 6.14. Comparison of approaches with fixed and heuristic-based time intervals between auctions in the presence of mobility based on the Slaw mobility traces.

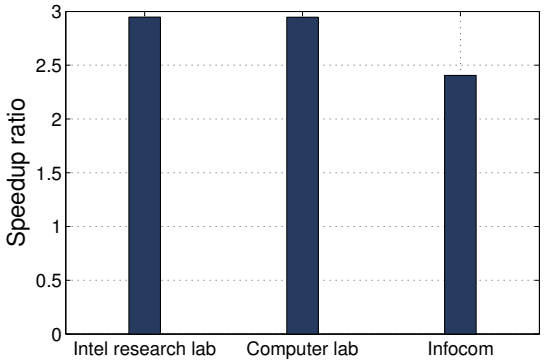

Figure 6.15. Speedup ratio of mobility aware approach using our heuristic for selecting the time interval between auctions in DAG jobs for three real contact traces.

CHAPTER 7

SUMMARY AND FUTURE WORK

7.1 Summary

We have exploited distributed sensing and computing in wireless networks to create novel, efficient approaches for localization and computation offloading. We have proposed efficient localization approaches for target tracking and spectrum monitoring that leverage the sensing power of a distributed set of wireless devices. Additionally, we have also shown how we can exploit the computing power of a set of nearby mobile devices to create a profitable task allocation. We also addressed the problems of incentives and privacy that arise in these applications and develop solutions for them. Our research work has demonstrated how we can exploit the sensing and the computing capabilities of the pervasive wireless devices to create efficient applications.

First, we proposed two effective energy efficient RTI approaches, ellipse-based and radius-based, for localization using RF sensor networks. In both energy efficient approaches, our aim was to save energy by reducing the number of links that we must measure to form an image of attenuation. In the ellipse-based approach, we only considered links in an ellipse around the velocity vector of the current position of the moving object and in the radius-based approach, we used links in a circle around the current position of the moving object. In addition, we proposed an algorithm to tune the radius of circle adaptively over time. We performed extensive evaluations using real experimental data from three different settings. Our experimental results showed that our energy efficient approaches can save 50% to 80% of energy without seriously degrading localization accuracy. Interestingly, our radius-based approach even increased the accuracy of localization in comparison to the basic RTI approach.

Second, we presented and evaluated a framework to locate multiple transmitters using crowdsourced measurements of received power. We addressed two main challenges in this framework. First, we presented a simple yet efficient and accurate method, SPLOT,

for simultaneous localization of multiple transmitters using the received power measured by the selected mobile sensing devices. SPLOT considered the temporal availability and mobility of both receivers and transmitters and made no assumptions about the number of transmitters. Second, we presented efficient sampling approaches that determined the number and locations of required mobile sensing devices for measurement. Our sampling approaches selected a set of wireless devices with maximum coverage considering mobility of both the sensing and the offending devices in a timely manner. Next, we enhanced our sampling to provide incentives for mobile sensing devices. We experimentally evaluated our framework and methods, and our results demonstrated that using SPLOT we are able to localize multiple transmitters with high accuracy and in a timely manner.

Third, we investigated an attack on location privacy where the location of people moving inside a private area can be inferred using the radio characteristics of wireless links that are leaked by legitimate transmitters deployed inside the private area. We modeled the radio network leakage attack using a Stackelberg game and used a Greedy method to obtain the optimal strategy for the defender. Our experimental results showed that our game theoretic solution significantly reduces the chance of an attacker finding the location of people inside a perimeter.

Finally, we presented and evaluated a game theoretic framework for task allocation in mobile cloud computing environments comprising of selfish mobile devices. Specifically, we proposed a multidimensional auction for allocating the tasks of a job among nearby mobile nodes based on their computational capabilities and also the cost of computation at these nodes with the goal of reducing the overall job completion time and be beneficial to all the parties involved. We considered node and task heterogeneity as well as node mobility in developing our methods. Our evaluations demonstrated showed that our game theoretic framework improves the job completion time by a factor of 2-5 in comparison to the time taken for executing the job locally, while minimizing the number of auctions and the accompanying overheads. Our approach is also profitable for the nearby nodes that execute the distributor's tasks with these nodes receiving a compensation higher than their actual costs. We also applied our game theoretic approach to mobile live video upstreaming application.

7.2 Future Research Directions

In this section, we describe several new directions that can progress the research in this dissertation.

7.2.1 Energy Efficient Target Tracking in RF Sensor Networks

Several future research directions exist for energy efficient target tracking using RF sensor networks. First, in our work, we proposed energy efficient approaches for tracking a single target only. However, in many real-world environments and settings, multiple people or objects are expected to move and be tracked in the monitored areas. While there is growing work on multiple target tracking [35, 38], energy efficiency in these scenarios has not received much attention. Future work must extend our energy efficient approaches for multiple target tracking. Second, it will be interesting to improve our energy efficient approach in terms of both the energy consumption and the error of localization. Currently, we reduce the energy consumption to half in the worst case. One way to reduce both the energy consumption and the localization error is by reducing the number of measured links even within the ellipse or the circle. Our idea is to assign weights to individual links and select a subset of them based on their weights. For example, we can weigh links based on their lengths. Short length links provide more information about the attenuation compared to the long ones. Figure 7.1 shows the difference in the localization error when we use all links that are crossing the monitored area versus the cases when we use k links in the monitored area. The k links are selected using shortest links first. In Figure 7.1, the total number of links is 870. This figure shows that increasing the number of links from 570 to 870 does not change the estimated position. However, not all short links are informative (e.g., the short links that cross metallic obstructions can actually negatively impact the accuracy). Thus, the location of the link or its fade level is very important. Therefore, we can weigh links based on multiple metrics, use the most effective links, and thereby save more in terms of energy without sacrificing accuracy. Compressed sensing (CS) which estimates tomographic images using fewer link measurements is another way of saving energy. Future work must also combine the benefits of both CS and our energy efficient approaches. Third, besides reducing the total energy consumption, future work must distribute this energy benefit uniformly across the sensor nodes. In our current

energy efficient approaches, the maximum energy consumption among the sensor nodes is half of that in the basic approach. However, this energy is not uniformly distributed among the sensor nodes especially in the cases where the person moves only in the small part of the experiment area and where sensor nodes are deployed nonuniformly around the perimeter, e.g., the bookstore. To tackle this problem, we can consider the energy consumption of nodes in selecting the links within the circle or the ellipse.

7.2.2 Fingerprinting-based Localization for Crowdsourced Spectrum Monitoring

In Chapter 3, we have presented a method for simultaneous power-based localization of transmitters (SPLOT) for crowdsourced spectrum monitoring. In SPLOT, we have reduced the problem of locating multiple, unknown numbers of transmitters to that of localizing a set of single transmitters by finding local maxima of RSSs observed by receivers. Another interesting approach to locate multiple unknown transmitters is to use RSS fingerprints. Fingerprinting-based methods have been widely explored for indoor localization of people and objects. The main idea is to create a fingerprint database of every location in the area of interest and then locate people or objects by mapping the online measured fingerprint against the fingerprint database. If we want to use fingerprinting-based methods for multiple transmitters localization, we need to create fingerprints considering all combinations of transmitters and receivers. For example for four transmitters, we need to create fingerprints for all 16 possible numbers of active transmitters for a given receivers. We also must create fingerprints for different locations of both transmitters and receivers. This increase the size of fingerprint database and the challenge is how to reduce the size of database by finding correlation. One way to reduce the size of database is to hold only single active transmitters and use the sum of linear received powers measured from each transmitter when transmitting sequentially to approximate the RSS for all possible multiple active transmitters. A future fingerprinting-based method must also explore different machine learning techniques to reduce the database size and locate multiple transmitters in a timely manner.

7.2.3 Scalable Localization for Crowdsourced Spectrum Monitoring

Our simultaneous multisource localization research for crowdsourced spectrum monitoring will not scale when covering large areas with hundreds of devices with only one central controller. Future work must extend our approach to hierarchical controllers. The large areas must be divided into a set of small areas based on the distribution and mobility patterns of mobile devices (there might be a small area with a large number of connected mobile devices or a large area with a small number of mobile devices). The best way to divide the area is to construct the connectivity graph and find communities in the graph. Then, the large area can be divided into a set of small areas covered by communities. Next, a controller can be placed in each small area. The controller in each area can receive the RSSs measurements from the local receivers, compress the received information, send it to an upper-level controller, and so on until the topmost layer of the hierarchy is reached. Alternatively, the central controller can perform localization based on received information and report the localization results to the topmost layer of the hierarchy.

7.2.4 Improving Sampling for Crowdsourced Spectrum Monitoring

Our proposed sampling approaches in Chapter 4 select a set of sensing devices that maximize the coverage in a timely manner considering the mobility and the selfish behavior of sensing devices. One interesting avenue for further exploration will be how to select a set of sensing devices to increase the accuracy of localization. In Chapter 4, we also proposed truthful sampling approaches to provide the required incentives for the selfish devices to participate in crowdsourced sensing. Beside the selfish behavior of sensing devices, it may be possible that some sensing devices act maliciously. For example, some sensing devices may change the received RSSs and report incorrect values. This makes both the sampling and the localization unreliable. Future work must explore different approaches including majority voting, and correlation in measurements performed by nearby nodes, to detect and handle malicious users/devices.

7.2.5 Improving Profitable Task Allocation

In Chapter 6, we have presented a profitable task allocation in mobile cloud computing that reduces the overall job completion time. Mobile devices such as smartphones have limited available energy and there are some scenarios when energy consumption is as

important as the overall job completion time. Therefore, future work must consider the energy consumption as well. In our profitable task allocation, we also developed a heuristic approach to dynamically find the best time intervals between auctions. Although the heuristic approach reduces the percentage of disconnected nodes, it is still possible that some mobile smartphones get disconnected before completing the assigned tasks or some other smartphones arrive with higher computational power and less cost. One way to tackle this problem is to improve the profitable task allocation to prevent the selfish mobile devices from lying about their availability for the computing task. In this setting, when the distributor probes nearby mobile devices for task allocation over the time interval T, the mobile devices reply by declaring their costs, computational capabilities and also their availability. Based on this information, the distributor selects a set of mobile devices with minimum costs that reduce the job completion time and also are available during the time interval T. By providing incentives for the mobile devices to declare the availability truthfully, the distributor only selects among those devices that are available over the time interval T. Another way to tackle this problem is to offload the computing task to the nearby mobile devices and whenever a new mobile device arrives with less cost and high computational capability, the distributor decides on preemption of the computing task. The distributor can efficiently preempt the computing task on one of the current selected mobile devices and reallocate the computing task to the newly arrived mobile device. Future work must explore the efficiency of these approaches in terms of the overall job completion time and the accompanying overheads.

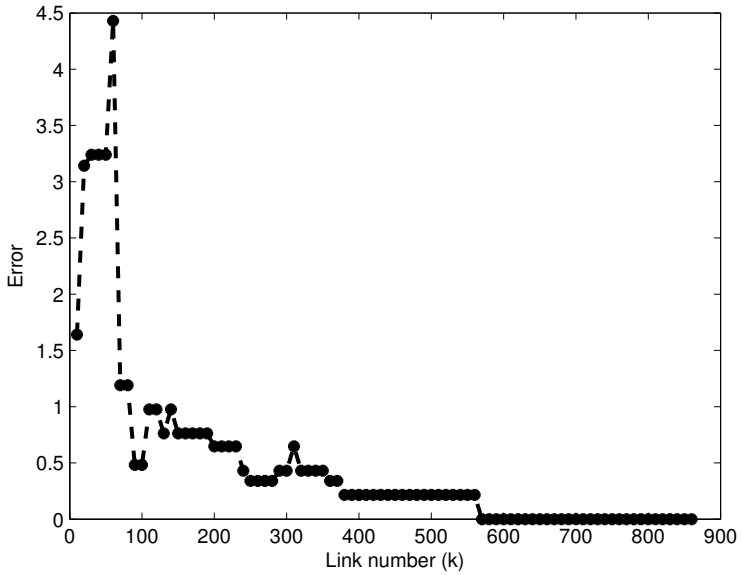

Figure 7.1. The difference between the localization results when we use all links versus when we use the k shortest links.

REFERENCES

[1] N. Patwari and J. Wilson, "Rf sensor networks for device-free localization and tracking," *Proc. of the IEEE*, vol. 98, no. 11, pp. 1961–1973, Nov. 2010.

[2] O. Kaltiokallio, M. Bocca, and N. Patwari, "Follow @grandma: Longterm device-free localization for residential monitoring," in *IEEE Int. Workshop on Practical Issues in Building Sensor Network Applications*, Oct. 2012.

[3] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, and H. Kremo et al., "Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *IEEE WCNC*, 2005.

[4] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, and R. J. ODea, "Relative location estimation in wireless sensor networks," *IEEE Trans. on Signal Processing*, vol. 51, no. 8, Aug. 2003.

[5] J. Braams, "A node scheduling scheme for energy conservation in large wireless sensor networks," *Wirel. Comm. Mobile Comput*, vol. 3, no. 2, Mar. 2003.

[6] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *ACM Trans. Sen. Netw.*, vol. 1, no. 1, Aug. 2005.

[7] B. Cărbunar, A. Grama, J. Vitek, and O. Crbunar, "Redundancy and coverage detection in sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 1, Feb. 2006.

[8] L. G. Jaimes, I. Vergara-Laurens, and M. A. Labrador, "A location-based incentive mechanism for participatory sensing systems with budget constraints," in *IEEE PerCom*, 2012.

[9] X. Ying, S. Roy, and R. Poovendran, "Incentivizing crowdsourcing for radio environment mapping with statistical interpolation," in *IEEE DySpan*, 2015.

[10] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "Trac:truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *IEEE INFOCOM*, 2014.

[11] L. Gao, F. Hou, and J. Huang, "Providing long-term participatory incentive in participatory sensing," in *INFOCOM*, 2015.

[12] A. Banerjee, D. Maas, M. Bocca, N. Patwari, and S. Kasera, "Violating privacy through walls by passive monitoring of radio windows," in *Wisec*, 2014.

[13] E. Marinelli, "Hyrax: Cloud computing on mobile devices using mapreduce," Tech. Rep., Masters thesis, CS. Dept., CMU, 2009.

[14] A. Mtibaa, A. Fahim, K. A. Harras, and M. H. Ammar, "Towards resource sharing in mobile device clouds: Power balancing across mobile devices," in *SIGCOMM MCC*, 2013.

[15] A. Mtibaa, K. Harras, and A. Fahim, "Towards computational offloading in mobile device clouds," in *IEEE CloudCom*, 2013.

[16] C. Shi, V. Lakafosis, M.H. Ammar, and E.W. Zegura, "Serendipity: Enabling remote computing among intermittently connected mobile devices," in *MobiHoc*, 2012.

[17] "Native boinc for android," http://nativeboinc.org/site/uncat/start/.

[18] K. Habak, M. Ammar, K. A. Harras, and E. Zegura, "Femtoclouds: Leveraging mobile devices to provide cloud service at the edge," in *IEEE CLOUD*, 2015.

[19] A. Saeed, M. Ammar, K. A. Harras, and E. Zegura, "Vision: The case for symbiosis in the internet of things," in *SIGMobile MCS*, 2015.

[20] M. Khaledi, S. Kasera, N. Patwari, and M. Bocca, "Energy efficient radio tomographic imaging," in *IEEE SECON*, 2014.

[21] M. Khaledi, M. Khaledi, S. Kasera, and N. Patwari, "Simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring," in *ACM MobiCom*, 2017.

[22] M. Khaledi, M. Khaledi, S. Kasera, and N. Patwari, "Preserving location privacy in radio networks using a stackelberg game framework," in *ACM Q2SWiNet*, 2016.

[23] M. Khaledi, M. Khaledi, and S. Kasera, "Profitable task allocation in mobile cloud computing," in *ACM Q2SWiNet*, 2016.

[24] N. Patwari and P. Agrawal, "Effects of correlated shadowing: Connectivity, localization, and RF tomography," in *IEEE/ACM Int. Conf. on Information Processing in Sensor Networks (IPSN'08)*, Apr. 2008, pp. 82–93.

[25] J. Wilson and N. Patwari, "Radio tomographic imaging with wireless networks," *IEEE Trans. Mobile Computing*, vol. 9, no. 5, pp. 621–632, May 2010.

[26] R. K. Martin, C. Anderson, R. W. Thomas, and A. S. King, "Modelling and analysis of radio tomography," in *2011 4th IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec. 2011.

[27] O. Kaltiokallio and M. Bocca, "Real-time intrusion detection and tracking in indoor environment through distributed rssi processing," in *IEEE 17th Int. Conf. Embedded and Real-Time Computing Systems and Applications (RTCSA)*, Aug. 2011, vol. 1, pp. 61 –70.

[28] O. Kaltiokallio, M. Bocca, and N. Patwari, "Enhancing the accuracy of radio tomographic imaging using channel diversity," in *IEEE Int. Conf. on Mobile Ad hoc and Sensor Systems*, 2012.

[29] J. Wilson and N. Patwari, "See through walls: Motion tracking using variance-based radio tomography networks," *IEEE Trans. Mobile Computing*, vol. 10, no. 5, pp. 612–621, May 2011.

[30] Y. Zhao and N. Patwari, "Noise reduction for variance-based device-free localization and tracking," in *IEEE SECON'11*, 2011.

[31] D. Zhang, J. Ma, Q. Chen, and L. M. Ni, "An RF-based system for tracking transceiver-free objects," in *IEEE PerCom'07*, 2007, pp. 135–144.

[32] X. Chen, A. Edelstein, Y. Li, M. Coates, M. Rabbat, and A. Men, "Sequential Monte Carlo for simultaneous passive device-free tracking and sensor localization using received signal strength measurements," in *ACM/IEEE Information Processing in Sensor Networks (IPSN)*, Apr. 2011.

[33] Y. Zheng and A. Men, "Through-wall tracking with radio tomography networks using foreground detection," in *Proc. Wireless Communications and Networking Conference (WCNC 2012)*, Apr. 2012, pp. 3278–3283.

[34] M. Moussa and M. Youssef, "Smart services for smart environments: Device-free passive detection in real environments," in *IEEE PerCom-09*, 2009, pp. 1–6.

[35] J. Wilson and N. Patwari, "A fade level skew-Laplace signal strength model for device-free localization with wireless networks," *IEEE Trans. Mobile Computing*, vol. 11, no. 6, June 2012.

[36] D. Zhang, J. Ma, Q. Chen, and L. M. Ni, "Dynamic clustering for tracking multiple transceiver-free objects," in *IEEE PerCom'09*, 2009, pp. 1–8.

[37] C. Xu, B. Firner, R. S. Moore, Y. Zhang, W. Trappe, R. Howard, F. Zhang, and N. An, "Scpl: Indoor device-free multi-subject counting and localization using radio signal strength," in *Proc. of the 12th Int. Conf. on Information Processing in Sensor Networks*, Apr. 2013, pp. 79–90.

[38] M. Bocca, O. Kaltiokallio, N. Patwari, and S. Venkatasubramanian, "Multiple target tracking with RF sensor networks," *IEEE Trans. on Mobile Computing*, vol. 13, no. 8, pp. 1536–1233, Aug. 2013.

[39] F. Adib and D. Katabi, "See through walls with Wi-Fi!," in *ACM SIGCOMM'13*, Aug. 2013.

[40] D. Maas, J. Wilson, and N. Patwari, "Toward a rapidly deployable rti system for tactical operations," in *8th IEEE Int. Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2013)*, Sydney, Oct. 2013.

[41] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *The 19th Ann. Int. Conf. on Mobile Computing and Networking (Mobicom'13)*, Sept. 2013.

[42] N. Patwari, J. Wilson, S. Ananthanarayanan P.R., Sneha K. Kasera, and D. Westenskow, "Monitoring breathing via signal strength in wireless networks," Tech. Rep. arXiv:1109.3898v1 [cs.NI], arXiv.org, Sept. 2011.

[43] B. Mager, N. Patwari, and M. Bocca, "Fall detection using RF sensor networks," in *IEEE Personal, Indoor and Mobile Radio Communications Conference (PIMRC 2013)*, London, Sept. 2013.

[44] S. Sigg, M. Scholz, S. Shi, Y. Ji, and M. Beigl, "RF-sensing of activities from non-cooperative subjects in device-free recognition systems using ambient and local signals," *IEEE Trans. Mobile Computing*, vol. 13, no. 4, pp. 1536–1233, Apr. 2013.

[45] O. Kaltiokallio and M. Bocca, "Real-time intrusion detection and tracking in indoor environment through distributed RSSI processing," in *2011 IEEE 17th Int. Conf. Embedded and Real-Time Computing Systems and Applications (RTCSA)*, Aug. 2011, vol. 1, pp. 61 –70.

[46] Y. Mostofi, "Cooperative wireless-based obstacle/object mapping and see-through capabilities in robotic networks," *IEEE Trans. on Mobile Computing*, vol. 12, no. 5, pp. 817–829, May 2013.

[47] A. Gonzalez-Ruiz and Y. Mostofi, "Cooperative robotic structure mapping using wireless measurements: A comparison of random and coordinated sampling patterns," *IEEE Sensors Journal*, vol. 13, no. 7, pp. 2571–2580, July 2013.

[48] M. A. Kanso and M. G. Rabbat, "Compressed RF tomography for wireless sensor networks: Centralized and decentralized approaches," in *5th IEEE Int. Conf. on Distributed Computing in Sensor Systems (DCOSS-09)*, Marina Del Rey, CA, June 2009.

[49] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *In Proc. of the Hawaii Conf. on System Sciences*, Jan. 2000.

[50] X. Wang, J. J. Ma, S. Wang, and D. W. Bi, "Cluster-based dynamic energy management for collaborative target tracking in wireless sensor networks," *Sensors*, vol. 7, July 2007.

[51] J. Fuemmeler and V. Veeravalli, "Smart sleeping policies for energy efficient tracking in sensor networks," *IEEE Trans. Signal Processing*, vol. 56, no. 5, May 2008.

[52] J. Wilson, "SPIN: TinyOS code for RSS collection," http://span.ece.utah.edu/spin.

[53] Texas Instruments, "A USB-enabled system-on-chip solution for 2.4 GHz IEEE 802.15.4 and ZigBee applications," http://www.ti.com/lit/ds/symlink/cc2531.pdf.

[54] CrossBow, "Telosb data sheet," http://www.willow.co.uk/TelosB Datasheet.pdf.

[55] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, "Locating the nodes: Cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, July 2005.

[56] A. Nasif and B. Mark, "Collaborative opportunistic spectrum sharing in the presence of multiple transmitters," in *IEEE Globecom'08*, 2008.

[57] R. Martin and R. Thomas, "Algorithms and bounds for estimating location, directionality, and environmental parameters of primary spectrum users," *IEEE Trans. Wireless Commun.*, vol. 8, no. 11, Nov. 2009.

[58] J. K. Nelson, M. R. Gupta, J. E. Almodovar, and W. H. Mortensen, "A quasi em method for estimating multiple transmitter locations," *IEEE Signal Processing Letters*, vol. 16, no. 5, May 2009.

[59] C. Kim, D. Song, Y. Xu, J. Yi, and X. Wu, "Cooperative search of multiple unknown transient radio sources using multiple paired mobile robots," *IEEE Trans. on Robotics (T-RO)*, vol. 30, no. 5, Oct. 2014.

[60] T. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall PTR, 2nd edition, 2001.

[61] J. Nonnenmacher, E. Biersack, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission," *SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 4, Aug. 1997.

[62] J. Hoffman and R. Mahler, "Multitarget miss distance via optimal assignment," *IEEE Trans. on Systems, Man and Cybernetics Part A: Systems and Humans*, vol. 34, no. 3, May 2004.

[63] M. Bocca, O. Kaltiokallio, N. Patwari, and S. Venkatasubramanian, "Multiple target tracking with rf sensor networks," *IEEE Trans. Mobile Computing*, vol. 13, no. 8, Aug. 2014.

[64] D. Schuhmacher, V. Ba-Tuong, and V. Ba-Ngu, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. on Signal Processing*, vol. 56, no. 8, Aug. 2008.

[65] "Rtl-sdr," http://sdr.osmocom.org/trac/wiki/rtl-sdr.

[66] "Bf-f8hp," https://baofengtech.com/bf-f8hp.

[67] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *IEEE INFOCOM*, 2000.

[68] U. Feige, "A threshold of ln n for approximating set cover," *Journal of the ACM*, vol. 45, no. 4, July. 1998.

[69] N. Metropolis, M. N. Rosenbluth, A. H. Teller, and E. Telle, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, Dec. 1953.

[70] C. Hubler, H. P. Kriegel, K. Borgwardt, and Z. Ghahramani, "Metropolis algorithms for representative subgraph sampling," in *ICDM'08*, 2008.

[71] L. Anderegg and S. Eidenbenz, "Ad hoc-vcg: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents," in *MobiCom*, 2003.

[72] C. Kim, D. Song, Y. Xu, J. Yi, and X. Wu, "Optimal auction design," *Mathematics of Operations Research*, vol. 6, no. 1, Feb. 1981.

[73] Y. Singer, "Budget feasible mechanisms," in *FOCS*, 2010.

[74] B. Wang, "Coverage problems in sensor networks: A survey," *ACM Computing Surveys*, vol. 43, no. 4, Oct. 2011.

[75] M. Youssef, M. Mah, and A. Agrawala, "Challenges: Device-free passive localization for wireless environments," in *MobiCom*, 2007.

[76] F. Adib and D. Katabi, "See through walls with wifi!," in *SIGCOMM*, 2013.

[77] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3d tracking via body radio reflections," in *NSDI*, 2014.

[78] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *MobiCom*, 2013.

[79] Y. S. Kim, P. Tague, H. Lee, and H. Kim, "Carving secure wi-fi zones with defensive jamming," in *ASIACCS*, 2012.

[80] M. Tambe, *Security and Game Theory*, Cambridge University Press, 2011.

[81] M. Manshaei, Q. Zhu, T. Alpcan, T. Basar, and J-P. Hubaux, "Game theory meets network security and privacy," *ACM Computing Surveys*, vol. 45, no. 3, June 2013.

[82] K. Rasmussen and C. Srdjan, "Location privacy of distance bounding protocols," in *Computer and Communications Security*, 2008.

[83] S. Brands and D. Chaum, "Distance-bounding protocols (extended abstract)," in *Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, 1993.

[84] X. Fu, N. Zhang, N. Debba, and A. Pingley, "The digital marauder's map: A new threat to location privacy," in *Distributed Computing Systems*, 2009.

[85] T. Jiang, J. Wang, and Y. Hu, "Preserving location privacy in wireless lans," in *MobiSys*, 2007.

[86] M. Gruteser and D. Grunwald, "Enhancing location privacy in wireless LAN through disposable interface identifiers: A quantitative analysis," *Mobile Networks and Applications*, vol. 10, no. 3, June 2005.

[87] Z. Han, N. Marina, M. Debba, and A. Hjrungnes, "Physical layer security game: How to date a girl with her boyfriend on the same table," in *IEEE GameNets*, 2009.

[88] G. Brown, M. Carlyle, J. Salmeron, and K. Wood, "Defending critical infrastructure," *Interfaces*, vol. 36, no. 6, Nov. 2006.

[89] P. Paruchuri, J. P. Pearce, M. Tambe, F. Ordez, and S. Kraus, "An efficient heuristic approach for security against multiple adversaries," in *AAMAS*, 2007.

[90] MJ. Khaledi, MH. Khaledi, and HR. Rabiee, "Fuzzy mobility analyzer: A framework for evaluating mobility models in mobile ad-hoc networks," in *IEEE WCNC*, 2010.

[91] C. Kiekintveld, J. Marecki, and M. Tambe, "Approximation methods for infinite Bayesian Stackelberg games: Modeling distributional payoff uncertainty," in *AAMAS*, 2011.

[92] "Cisco wireless controllers," http://www.cisco.com/en/US/products/hw /wireless/products_category_buyers_guide.html#c̃ontrollers.

[93] "Idc," https://www.idc.com/.

[94] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: Making smartphones last longer with code offload," in *MobiSys*, 2010.

[95] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Pa, "Clonecloud: Elastic execution between mobile device and cloud," in *EuroSys*, 2011.

[96] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, "Seti@home: An experiment in public-resource computing," *ACM Commun*, vol. 45, no. 11, Nov. 2002.

[97] D. P. Anderson, "Boinc: A system for public-resource computing and storage," in *IEEE/ACM GIRD*, 2004.

[98] R. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H.I. Yang, "Simplifying cyber foraging for mobile devices," in *MobiSys*, 2007.

[99] N. Eagle, "txteagle: Mobile crowdsourcing," in *HCII*, 2009.

[100] P. Lundrigan, M. Khaledi, M. Kano, N. Subramanyam, and S. Kasera, "Mobile live video upstreaming," in *ITC 28*, 2016.

[101] A. Papakonstantinou and P. Bogetoft, "Incentives in multi-dimensional auctions under information asymmetry for costs and qualities," in *AMEC/TADA*, June 2012.

[102] Y. Zhang and M. van der Schaar, "Incentive provision and job allocation in social cloud systems," *IEEE JSAC*, vol. 13, no. 9, Sept. 2013.

[103] A. Singla and A. Krause, "Truthful incentives in crowdsourcing tasks using regret minimization mechanisms," in *WWW*, 2013.

[104] M. Khaledi, M. Khaledi, and HR. Rabiee, "An optimal game theoretical framework for mobility aware routing in mobile ad hoc networks," Tech. Rep. arXiv:1407.7464[cs.NI], arXiv.org, 2014.

[105] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *CODES/ISSS*, 2010.

[106] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2006-01-31)," 2006.

[107] K. Lee, S. Hong, S.J. Kim, and I. Rhee, "Slaw: A new mobility model for human walks," in *INFOCOM*, 2009.