PERFORMANCE ANALYSIS OF

HYBRID MOBILE SENSOR NETWORKS

WANG WEI

(*B.Eng, M.Eng, NJU*)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2008

Acknowledgements

First, I would like to express my deepest gratitude to my supervisors, Dr. Vikram Srinivasan and Prof. Chua Kee-Chaing, for their excellent guidance through my Ph.D study. Their deep insights and advices beyond academic and research were and will always be well appreciated.

I would like to thank Dr. Mehul Motani for our fruitful collaborations and for his helpful suggestions in my research.

I would also like to thank my friends in Computer Networks and Distributed Systems Laboratory, Wang Bang, Zhao qun, Yap Kok-kiong, Luo Tie, Yeow Wai-leong, Ai Xin, Farshad Ahdi, etc., for their warm-hearted help and beneficial discussions which make my stay in the laboratory a very enjoyable and memorable one.

Last but not least, I am always grateful to my parents and my wife, for their patience and support during the long march of my Ph.D study.

Contents

Summary

Mobile sensors have become an important research area in wireless sensor networks, since mobility is a promising solution to the resource provisioning problem in sensor networks. Mobile sensors can redistribute their resources to resource bottlenecks and mobilize resources efficiently. Consequently, the network performance can be greatly improved by introducing mobile sensors into the network.

In this thesis, we focus on the performance of hybrid mobile sensor networks, which are formed by a small number of mobile sensors with a large number of static sensors. More specifically, we consider three performance metrics in hybrid sensor networks, which are (i) network lifetime, (ii) network coverage and (iii) localization accuracy.

Network lifetime optimization is vital to sensor networks, since sensors are normally required to work for months or even years with non-rechargeable batteries. In a static sensor network, sensors around the sink are bottlenecks for network lifetime since their energy can be quickly exhausted by the relaying tasks. In this thesis, we consider using mobile relays to help bottleneck sensors in relaying traffic. We analytically show that the network lifetime can be improved by 4 times with a single mobile relay in an ideal dense network. We also show by numerical results that the mobile relay solution can double the network lifetime in randomly deployed networks.

We then consider the coverage problem in randomly deployed networks. For a static network, the sensor density needs to increase as $O(\log L)$ to fully cover a network with size of L. This leads to an inefficient deployment in large networks. In this thesis, we propose a hybrid network structure which uses a small fraction of mobile sensors to cover the holes in the network. In particular, the mobile sensors used in this thesis can only move once for a short distance. We prove that the maximum moving distance for mobile sensors only increase with the network size as $O(\log^{3/4} L)$. We then use a distributed mobility algorithm to coordinate the movements of mobile sensors. The algorithm has also been implemented in real mobile sensors.

Finally, we investigate the localization problem for hybrid mobile sensor networks. We use static sensors as beacons to localize mobile sensors in the hybrid network. We study the problem of estimating the necessary beacon density for accurate localization over the network field. We show that the conventional disk coverage model is insufficient in localization problems since it overestimates the beacon density by a factor of two times. We then derive a new coverage model, called sector coverage, for localization coverage. Numerical results show that our sector coverage model provide better estimation for the beacon density.

List of Figures

List of Tables

List of Symbols

A	Area of a region.
$d_{i,j}$	Euclidean distance between node i and node j.
E	Initial energy stored in a static node.
$\mathbb{E}\{\cdot\}$	Mean of a random variable.
l	Network side length (for square networks).
L	Area of the whole network (for square networks).
λ	Static sensor density.
Λ	Mobile sensor density.
M	Total number of *mobile* sensors in the sensor network.
N	Total number of *static* sensors in the sensor network.
η	Over-provisioning factor of a network.
$O(\cdot)$	$g(n)$ is $O(f(n))$ if $\exists c > 0, n_0 > 0$ s.t. $\|g(n)\| \leq c\|f(n)\|, \forall n > n_0$.
$\Omega(\cdot)$	$g(n)$ is $\Omega(f(n))$ if $\exists c > 0, n_0 > 0$ s.t. $\|g(n)\| \geq c\|f(n)\|, \forall n > n_0$.
$\Theta(\cdot)$	$g(n)$ is $\Theta(f(n))$ if $g(n)$ is both $O(f(n))$ and $\Omega(f(n))$.
$\mathbb{P}\{\cdot\}$	Probability of a event.
P_k	The set of sensors which is k hops to the sink.
Q_k	The set of k hop neighbors of the sink.

ρ	Average neighborhood size.
r	Sensing range of a sensor.
R	Radius of the network (for circular networks).
s	A static sensor.
T	Lifetime of sensor or the network.
σ	Localization error of sensors.
u	Network resolution for localization.

Abbreviations

ARA	Aggregation Routing Algorithm
ARALN	Aggregation Routing Algorithm with Limited Nodes
a.s.	almost surely
BB	Bayesian Bound
CRB	Cramér-Rao Bound
GPS	Global Positioning System
MAC	Media Access Control
PDA	Personal Digital Assistant
RAM	Random Access Memory
ROM	Read-Only Memory
RSSI	Received Signal Strength Indicator
TDOA	Time Difference Of Arrival
w.h.p.	With High Probability
WSN	Wireless Sensor Networks

Chapter 1

Introduction

Wireless Sensor Networks (WSNs) are networks formed by a large number of sensors which collect data from the physical world. Sensors self-organize to perform certain tasks such as environmental data gathering [1, 2], target tracking [3] or infrastructure monitoring [4].

The unique network structure of wireless sensor networks brings new challenges to wireless network designers. First, sensors are extremely resource limited compared to existing network devices such as Personal Computers or PDAs. Due to the large scale of wireless sensor networks, individual sensors are often designed as simple and cheap devices which can only carry limited resources such as energy, computational power and data storage. For example, the Berkeley mote MICA2 is powered by two alkaline AA batteries and has only a 4MHz processor with 128kB of instruction memory and 4kB of RAM [5]. Therefore, one important research issue in WSN is how to use these limited resources judiciously so that the network can achieve certain performance requirements such as network lifetime, sensing coverage and packet delivery ratio.

The randomness in sensor deployment is another important research problem in wireless sensor networks. Sensors are often randomly scattered in the field due to the inaccessibility of the terrain or the large scale of the network. In random deployments, certain regions in the network may contain far fewer sensors than others. Consequently, there will be less resource deployed in these regions. The unevenness in resource distribution makes it even more difficult to efficiently utilize the limited resources. Random deployment also causes the coverage problem in sensor networks. As sensors often have limited sensing ranges, randomly deployed sensor networks may contain coverage holes, which are areas that cannot be monitored by any sensors.

Introducing mobility into wireless sensor networks is a promising way to mitigate these problems. Mobile sensors can carry more resources and dynamically relocate their own resources to other sensors. Therefore, the network resources can be redistributed and used in a more efficient way. Furthermore, mobile sensors can also compensate the randomness in deployment by relocating themselves to heal coverage holes in the network.

Although mobile sensor networks have many advantages over static networks, few studies have been focused on quantifying the benefits of introducing mobile sensors. To close this gap, this thesis tries to quantitatively investigate the performance of mobile sensor networks and answer the following problems:

- How much improvement in network performance can be achieved by introducing mobility into sensor networks?
- How many mobile sensors should be deployed in the network in order to achieve the desired network performance?
- How can we reduce the energy consumption used in movements?
- How can we provide real-time location information to mobile sensors?

In the following sections, we will discuss mobile sensor networks in detail and provide an overview of this thesis.

1.1 Mobile Sensor Networks

Many problems in wireless sensor networks can be reduced to resource provisioning problems. The network resources can be energy (for the network lifetime problem), communication ability (for the network capacity problem) or sensing ability (for the coverage problem). When sensors are static, the only way to provide enough resources for a certain task is through over-provisioning. In other words, we can only deploy sensors in a higher density so that the resource deployed per unit area can be increased. However, over-provisioning is not an efficient solution, especially when sensors are randomly deployed. In randomly deployed sensor networks, we may need extremely high sensor density to provide enough resources in the resource bottleneck area. In this case, the resource placed in areas other than the bottleneck area will be far more than necessary.

For example, consider sensors randomly and uniformly deployed in a square network with an area of L. Suppose that each sensor can only monitor events occurring within a distance of r from it. If we wish to guarantee that every point in the network is monitored by at least k sensors, the required sensor density should increase with the network size L as $\Theta(\log L + k \log\log L)$ [6, 7, 8]. In other words, the required sensor density increases to infinity when the network size goes to infinity. When we use such a high sensor density, there will be far more than k sensors monitoring the same region in most areas of the network, while we only need every region to be monitored by k sensors. The extra resources deployed in the network are just for covering a few bottleneck areas.

Mobility provides a new way to solve the resource provisioning problem in sensor networks. Mobile sensors can carry resources to the right place at the right time so that their resources can be used more efficiently. In this way, the limited resources in wireless sensor networks can be dynamically relocated by mobile sensors to improve the network performance.

Recent research shows that mobility can improve the network performance in many aspects. First, the lifetime of sensor networks can be increased by several times when we introduce mobile sinks [9] or mobile relays [10] into the sensor network. Second, mobile sensors can improve the network coverage by relocating themselves to heal coverage holes in the network [11, 12, 13] or patrolling around the network field [14]. Finally, mobile nodes can also increase the network capacity of wireless ad-hoc networks by carrying data packets while moving around [15].

Mobile sensors can be practically implemented using cheap Commercial Off-The-Shelf (COTS) hardware and software. Various mobile robot platforms have been developed to provide mobility for wireless sensor networks in recent years [16, 17, 18, 19, 20, 21, 22]. In this thesis, we also describes a hardware platform which is cheap (less than two hundred dollars) and easy to use, see Chapter 6.3.

Compared to static sensors, mobile sensors need additional mobility platforms for movement. These mobility platforms can be more expensive than the static sensor itself [20, 21]. Therefore, the hardware cost of an *all-mobile sensor network*, where all sensors in the network can move, will be far more than a network made of static sensors. The advantages of mobility may be offset by the higher hardware costs. Many sensor network designers prefer to use a *hybrid mobile sensor network* structure, where only a small number of mobile sensors are used with a large number of static sensors [9, 10, 23]. In this way, the hardware cost can be reduced as fewer mobile sensors are deployed in the network. An important observation in hybrid mobile sensor networks is that the network performance may be greatly improved by using just a few mobile sensors [9, 24]. Related research in hybrid mobile sensor networks will be further discussed in Chapter 2.

Although much research has been devoted to mobile sensor networks, many important issues in hybrid mobile sensor networks have not been addressed yet. First, there are no quantitative studies on the improvement of network lifetime or network coverage that can be achieved by adding mobile sensors into a static network. Also, it is still unclear as to how many mobile sensors should be used in a hybrid network in order to achieve certain network performance.

Moreover, most previous studies do not consider the movement distance of the mobile sensors. They assume that mobile sensors have enough energy to move for infinite distance. However, movement normally cost much more energy than communication or sensing. If a mobile sensor uses too much energy in movement, its battery may be quickly exhausted after the movement. So, it is important to consider mobility algorithms in which mobile sensors only move for a limited distance.

Another important research problem in mobile sensor networks is to support location information to mobile sensors. As mobile sensors are moving around in the field, it is vital for mobile sensors to obtain its own location information in order to perform certain tasks, such as routing data packets, navigating movements and tagging sensing data with location information. Most previous studies in mobile sensor networks assume that mobile sensors have certain localization devices such as GPS to provide accurate location information. However, GPS devices normally are expensive and consume considerable energy. It is preferable to use static sensors as beacons to localize mobile sensors. In this case, it is important to find the necessary static sensor density under which the mobile sensors can be precisely localized.

1.2 Scope

The aim of this study is to analyze the network performance of hybrid mobile sensor networks. Among various aspects of network performance, the present thesis mainly focuses on three topics in hybrid mobile sensor networks as listed below:

1. **Network lifetime optimization in hybrid mobile sensor networks**

Sensors are usually equipped with non-rechargeable batteries, and often required to work for months or even years without replacing batteries. In this case, we need to optimally allocate the limited energy to various tasks performed by sensors, such as sensing, idle listening, data processing and relaying packets for other sensors.

In a static sensor network with a single sink at the center of the network, sensors around the sink need to relay every packet heading towards the sink. These sensors will quickly exhaust their energy in relaying traffics which dominates their energy consumption. When the sensors around the sink die, the network will stop functioning as data packets cannot reach the sink. In this case, these sensors around the sink will be the network lifetime bottleneck. We consider using resource-rich mobile relays to relieve the relay task on these sensors. These mobile relays will form a “bridge” connecting static sensors to the sink therefore the bottlenecks around the sink can be removed.

In this research topic, our major objective is to find the upper bound on network lifetime improvement for a hybrid mobile sensor network. We also wish to find an algorithm which can achieve this upper bound in the ideal case. In addition, we will compare the mobile relay approach to other existing approaches such as the mobile sink approach and static network approaches. Most of the routing algorithms described in this thesis are for theoretical analysis. Therefore, we ignored some of the designing details, such as MAC layer contention or routing information gathering phases. The problem of designing practical routing and channel access protocols for mobile sensor networks is out of the scope of this thesis.

2. Network coverage problem in hybrid mobile sensor networks

In most sensor networks, each sensor only can monitor a small region around itself. If we wish to provide a good monitoring quality over the sensing field, the network should be fully covered by the sensors. As discussed in section 1.1, we need to deploy sensor in a high density so that every region in the network can be fully covered. Such over-provisioning solutions are not efficient for large static sensor networks as the sensor density grows to infinity when the network size is large.

In this thesis, we consider how to use mobile sensors to heal coverage holes in the randomly deployed static network. Our goal is to use a combination of mobile sensors and static sensors which can fully cover the network with a constant sensor density, irrespective to the network size. In particular, we use mobile sensors which only have limited mobility. Since mobiles in our network only need to move once for a short distance, the energy used for movement are limited and mobiles can be made simple and cheap.

In this topic, we focus on deriving the upper bound on the moving distance of mobile sensors and studying the relationship between the moving distance and network size. We will also design distributed mobility algorithm for mobile sensors with limited movement distance in this thesis.

3. Localization problem in hybrid mobile sensor networks

Location information is crucial for mobile sensor networks. One efficient way to localize mobile sensors is using static sensors as beacons. In this case, the location of mobile sensors can be calculated by measuring the distance between mobiles to these beacons.

Static sensors should be deployed at a sufficient density to provide accurate localization over the network. The beacon deployment problem can also be treated as a coverage problem as the localization process requires enough number of beacons to be in the neighborhood of the mobile. Traditional coverage model for sensor networks assumes that the sensing regions are disks with uniform radius centered at sensors.

This disk coverage model is insufficient for the localization problem as it does not consider the relative location of beacons.

In this research topic, we focus on deriving a new coverage estimation method which can accurately estimate the necessary beacon density for a given localization requirement. We will also derive distributed sleep-wake algorithms for beacons to achieve full localization coverage. However, the detailed localization algorithm is out of the scope of this thesis.

1.3 Thesis Organization

The rest of this thesis is organized as follows:

Chapter 2 reviews the related research in mobile sensor networks. The review focus on three subtopics in mobile sensor networks: the network lifetime optimization problem, the network coverage problem and the localization problem.

Chapter 3 considers the network lifetime optimization problem in an ideal high density network. In such network, we theoretically derived the lifetime upper bound of static sensor networks and networks with M mobile sensors. We also construct two joint mobility and routing algorithms, called ARA and ARALN, to achieve the lifetime upper bound of hybrid sensor network with a single mobile relay. In these algorithms, the mobile relay only needs to move within a two hop radius of the sink and only part of the network needs to know the position of the mobile relay in the ARALN algorithm.

Chapter 4 extends the results in Chapter 3 to low density networks. In this chapter, we formulate the mobility and routing algorithm as a linear programming problem. We study the network performance metrics, such as network lifetime, network dilation and traffic distribution, through intensive numerical experiments. We also compare the performance of mobile relay approach with static network and mobile sink solution in this chapter. Finally, this chapter also shows other extensions of this problem where power control is used.

Chapter 5 considers the coverage problem in hybrid mobile sensor networks. In this chapter, we derive the maximum movement distance required for all-mobile sensor networks and hybrid mobile sensor networks to achieve full coverage in the field. We also derive the number of mobiles required for a hybrid mobile network structure in this chapter. This chapter also discusses the real world deployment issues for hybrid mobile sensor networks.

Chapter 6 provides a distributed mobility algorithm for covering hybrid mobile sensor networks. We show that the mobility algorithm can be formulated as a network flow problem and solved through a push-relabel algorithm. We also study the performance of our algorithm through analysis and numerical experiments. An implementation of the algorithm on real mobile sensors is also described in this chapter.

Chapter 7 considers the localization problem in mobile sensor networks. We first convert the coverage problem for localization to a conventional disk coverage problem, where the sensing area is a disk centered at the sensor. Our results show that the disk coverage model requires 4 times more sensors for *localization* compared to *detection* applications. We then introduce the idea of sector coverage to tighten the lower bound. The lower bound derived through sector coverage is 2 times less than through disk coverage. A distributed sector coverage algorithm is also proposed in this chapter.

Finally, Chapter 8 concludes the whole thesis and discusses future research directions in hybrid mobile sensor networks.

Chapter 2

Literature Review

Mobile sensor networks become an important research topic in wireless sensor networks, as it can greatly improve the performance of wireless sensor networks [9, 10, 11, 13].

There are two paradigms for wireless mobile sensor network architecture design. The first is to deploy mobile sensor networks where all nodes can move [25, 11, 26, 14, 27], which is called *all-mobile sensor networks* in this thesis. This architecture is often expensive as all sensors should be equipped with mobile devices. The second is to use a hybrid mobile network structure [9, 10, 28]. In this case, only a small number of mobile sensors are used with a large number of static sensors. This architecture is cheaper compared to the all-mobile network and still it performs quite well.

In this thesis, we focus on the second approach. We consider using a small number of mobiles to improve the network lifetime and coverage. We also consider how to localize the mobile sensors through static sensors. Previous research work in these topics will be reviewed in the following sections.

2.1 Network Lifetime of Mobile Sensor Networks

Existing literature utilizes mobile nodes as mobile sinks to save energy. Shah *et al.* [24] propose to use randomly moving "Data Mules" for data gathering. Mobile sinks with predictable and controllable movement patterns are studied in [29, 20, 30]. In these approaches, the static sensors only send out their data when the sink is close enough to them. The disadvantage of such proposals is that there will be considerable delay in packet delivery, since a node needs to wait for the sink to approach it.

In order to minimize the delay, several methods of transmitting the sensed data through multi-hop communications to the mobile sink are proposed in [31, 32, 9]. The mobile sink can either "jump" between several predefined positions or patrol on a continuous route. In the first case, the problem can be posed as a linear programming problem where a mobile sink can find the optimal time schedule to stay at these predefined points [31, 32]. Another method is introduced in [9], where the optimal route is obtained through a geographic traffic load model. In this approach, as the mobile sink goes around the network, sensors will continuously track the position of the sink and send their packets to the sink via multi-hop communications. The disadvantages of the mobile sink paradigm are three folds, (i) the mobile sink needs to roam around the periphery of the network to maximize the network lifetime [9], (ii) all nodes in the network will have to be aware of the current location of the mobile sink in order to route information to it [33] and (iii) in most scenarios, the sink is a gateway to a backbone network, over which human operators can monitor the status of the sensor field. In such scenarios, it will be difficult to engineer a system whereby a mobile sink is connected at all times to the backbone network. In this paper, we use mobile nodes in a different way compared to existing mobile sink approaches. Instead of using mobile node as a data gathering proxy, we use energy rich mobile relays to help energy limited static sensors and extend the overall network lifetime.

Another large category of energy conserving methods is to use routing algorithms to find the optimal energy conserving routes [34, 35]. The energy conserving routing and the mobile sink approach share the same idea of distributing the traffic load evenly around the network so that the lifetime of the network is maximized. Energy provisioning in static sensor networks is studied in [36], where a total amount of energy is added in relay nodes deployed at selected positions. Such static relay nodes can heal the topology defects in randomly deployed networks, so the network lifetime can be improved greatly when the network is sparse. However, as the network density grows beyond a certain threshold, the improvement gets saturated since most of the topology defects have been mitigated. Compared to the static relay approach, the mobile relay approach can provide considerable improvements on dense networks as well as healing the topology defects.

Other solutions for energy saving have also been intensely studied, including data aggregation and topology control methods. Data aggregation and clustering methods such as [37, 38, 39] aggregate the sensed data to decrease traffic volume and thereby prolong network lifetime. Topology control methods such as [40, 41] use controllable transmission range to achieve the most energy efficient network topology. In our work we do not address the issues of data aggregation or topology control. However, these ideas can be useful complements to our proposal of using mobile relays. As we will describe later, depending on the position of the mobile relay, traffic is intentionally routed via a few specific network nodes. This could facilitate the data aggregation process.

2.2 Coverage Problem in Mobile Sensor Networks

Coverage problem in wireless sensor network tries to determine how well the target area is monitored or tracked given the distribution of sensors [42, 43, 44, 45]. Existing literature addresses the problem of determining whether the field is fully covered [43, 46] and deriving sensor sleeping schedules while keeping the field fully covered [47]. Theses works are mostly on static networks.

Coverage problems in all-mobile sensor networks have also been intensively studied. For a all-mobile sensor network in which all nodes can move, the basic way to ensure coverage is to move sensors in the dense area to the sparse area, using methods like virtual forces [25]. Wang et al. evaluated three kind of mobility algorithm: VEC (Vector Based), VOR (Voronoi Diagram) Based and Minimax [11]. The work in [27] uses a scan based way to distribute the sensors evenly over an area. However, an all-mobile sensor network is expensive in hardware costs. It is preferable to use a hybrid network structure to improve the network coverage.

Most research in hybrid networks is focused on using mobiles to fill holes in the static sensor network. Wang et al. use biding protocols to minimize the moving distance for mobiles when a hole is detected [28, 48]. The price is described as the area of hole the mobile covered in the new position minus the area it covers in its current position. A gird quorum based algorithm is used in [12] to find redundant mobile nodes and a cascaded movement is proposed to minimize the movement distance of mobiles.

Most of these approaches do not consider the limitations on the distance that individual mobile sensors can move. Mobiles may also use non-rechargeable batteries and they may exhaust all their energy in a long distance movement. Chellappan et al. introduce flip-based sensors for network coverage improvement in [26]. The flip-based sensors can only move once, over a limited distance, therefore the costs of such sensors are quite low. A network flow based algorithm is used in [26] to find the mobility schedule which maximizes network coverage. The problem is further formulated as an optimization problem which minimizes the variance of sensors in different regions in [49]. However, both [26] and [49] do not provide performance bounds for fraction of area covered or maximum moving distance. In this thesis, we show that the network can be completely k-covered by using only a small fraction of mobile sensors. We give bounds on the maximum moving distance of mobile sensors. Furthermore, our movement schedule formulation is simpler than in [26]. We also provide a distributed algorithm which achieves the optimal solution, while the algorithm in [49] is a distributed heuristic algorithm, with no guarantees of optimality.

The bound on maximum mobile moving distance in this thesis is based on the minimax grid matching result for uniformly distributed points [50]. The minimax grid matching result has been applied to solve load balancing problems on graphs in [51] and emulation problems for sensor networks in [52]. In this thesis, we extended the original minimax matching result to different distributions to bound the matching distance between mobiles and vacancies in hybrid sensor networks, where the vacancies are not uniformly distributed.

2.3 Coverage for Localization in Mobile Sensor Networks

Most of previous works on the coverage problem in WSN consider covering every point in the sensing field with sensing disks [43, 53, 11] or detecting a target when it passes through the sensing field [44, 45]. In this thesis, we connect the coverage problem to the localization problem of mobile sensors. The assumption that the sensing region are disks no longer holds in the context of localization. However, we show that the results and algorithms in the previous works on disk models can also apply to the coverage model derived in Chapter 7. The sector coverage concept proposed in this thesis is similar to the one studied in [54]. However, [54] focuses on network connectivity rather than localization.

Localization is an important component in sensor networks, since it provides coordinates both for the sensors [55] and for the targets in the sensor network [56]. In this thesis, we focus on the problem where the locations of beacons are already known and the objective is to localize mobile sensors in the sensing field. We use range based sensors to localize the mobile sensor. Several well known estimation methods, such as Cramér-Rao Bound (CRB) [57] or Bayesian Bound (BB) [58], are used in range based localization systems. In this thesis, we use a simplified localization model for the coverage problem. Our goal is to bound the location estimation error to be within a small circle of radius ε with high probability. This is different from the normal approach of minimizing the mean squared value of the localization error. Our model is more suitable for systems where the performance is determined by the maximal localization error.

Connecting the coverage problem with the localization problem will provide useful guidelines for sensor network deployment [59]. Beacon deployment for sensor localization is studied in [60, 61]. Nagpal et al. point out that there should be voids around areas where the localization error is large [62]. In [62], the localization error is bounded through intersecting concentric rings centered at different beacons with a width same as the communication range. Although we use a similar method to bound the localization error by studying the intersection area of rings, we provide deeper analysis for range based localization systems rather than hop-count based systems.

Chapter 3

Mobile Relay in High Density Networks

In this chapter, we focus on using mobile nodes which have more energy than static sensors to extend the network lifetime. Normally, static sensors only have limited energy stored in non-rechargeable batteries. Once the battery runs out, the sensor will die accordingly. Therefore, it is critical to use energy judiciously in order to maximize the benefit from the network before it dies. In a multi-hop sensor network, the sensors need to relay packets for others in addition to sensing. Due to the unevenness in traffic distribution, some sensors may spend most of its energy in relaying.

In this chapter, we introduce mobile relays which are rich in energy to help energy limited static sensors in packet relaying. The mobile relays have the same communication range and sensing ability as the static sensors. Fig. 3.1 shows an example of how a mobile relay may work. Suppose that the network is composed of two components which are connected via two sensor nodes A and B. Since all traffic flows between these two components pass through these two sensors, their battery will drain quickly. Suppose the lifetime of these two nodes is T. Even if other sensors have lifetime much longer than T, the network will get partitioned when these two sensors die at time T. If we have one mobile relay helping them, the network lifetime can be at least doubled. A simple algorithm for this would be for the mobile node to shuttle be-

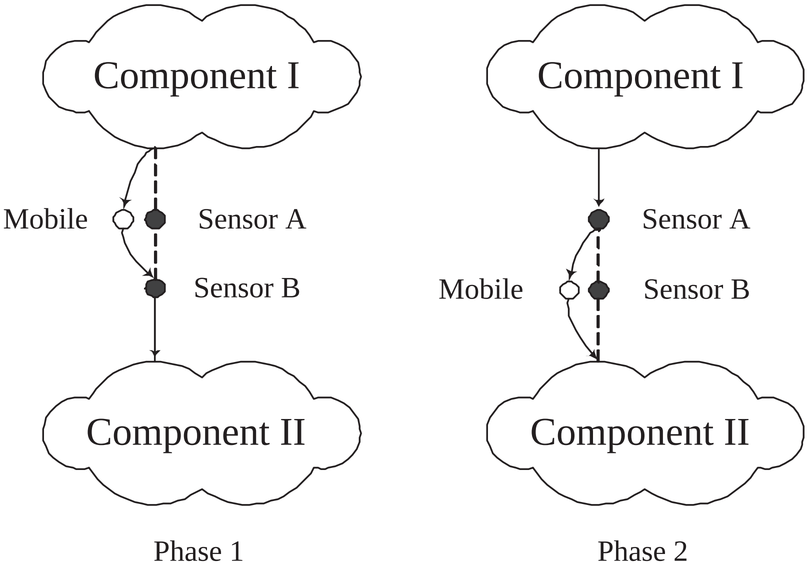


Figure 3.1: Using mobile relay to extend the lifetime of the bottleneck nodes

tween node A and node B and inherit the responsibilities of the node with which it is co-located (including sensing and relaying). It is clear that with an appropriate shuttling schedule, the network lifetime can be doubled to $2T$. We assume here that the energy resource at the mobile node is far greater than that of any of the static nodes.

3.1 Assumptions and Network Model

We assume that static sensors are densely deployed according to a Poisson point process with intensity λ in a circular area of radius $R \gg 1$. Since the network is dense, we assume that in each hop the packet can travel as far as the transmission range in any direction and the number of sensors in a region with area A is exactly

λA. We assume that there are N sensor nodes in the network with one sink s_0 at the center of the circular area.

We assume a data logging application, where the sensors are required to send their sensing data at a fixed rate. Furthermore, for the sake of simplicity, we assume that the data generation rates for all the sensors are the same. We normalize the data generation rate to one packet per time unit for each sensor. Our algorithms can also be applied to event driven applications. In this case, the performance could be potentially improved as the mobile relay can move towards the region where event occurs.

We also assume that the traffic load in the network is low and there are no congestions in the network. For highly loaded networks, packets may be dropped due to congestions around the sink. This violates the requirements that all data should be reliably transmitted to the sink. So, we don't consider networks with severe congestions. For network with moderate congestions, packet maybe retransmitted. This may cause extra overheads in energy consumption which will be further discussed in section 4.1.

We assume that the transmission range of all the sensors is equal to 1 and the sensors do not change their transmission powers [1]. We define ρ as the average number of neighbors for the sink. Since the transmission range is 1, the average number of nodes in the transmission range of the sink will be $\rho = \pi\lambda$.

We assume that all sensor nodes have the same initial energy E and the energy

[1] This assumption will be relaxed in Chapter 4.

of the sink is unlimited. For the energy consumption models, observing that there are many sleep scheduling methods to put idle sensors into sleep [63, 64], we suppose the network has adopted optimal sleep scheduling protocols. Thus, the energy consumption in idle state can be ignored and we only consider the energy used in sensing, receiving and transmission. We assume that sensors will consume e_s unit of energy for sensing data in each time unit, e_r and e_t for receiving and transmitting one packet, respectively. To further simplify our energy model, we assume that the transmission energy dominates the total energy consumption, so that the difference between e_s and e_r can be ignored. In our model, the total energy consumed by sensors in one outgoing packet is a constant e, which is the sum of the transmission energy and the receiving (or sensing) energy. Thus, if the average number of packets flowing out from the sensor i per time unit is f_i, the lifetime of this sensor would be:

$$T_i = \frac{E}{ef_i} \tag{3.1}$$

In this thesis, the lifetime of the whole network is defined as the time that the first node dies as in [34]. Since we assume that energy conserving routing is used, the network gets partitioned when the first node dies [34].

We assume mobile relays have the same sensing ability and transmission range as the static sensor nodes and they have rechargeable batteries thus there are no energy limits on mobile relays.

To facilitate our discussion, we divide static senors into different sets according to their distance to the sink. The set P_k contains all the nodes which can reach the sink with minimal hop count k. For example, the set of all the immediate neighbors

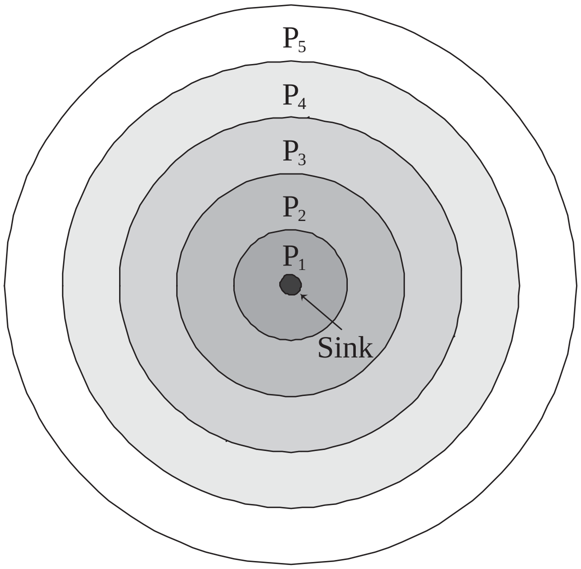


Figure 3.2: Dividing nodes to different subsets in the circular network

of the sink will be P_1. In a dense network, the sensor node s will be in the set P_k iff $d_{s,s_0} \in (k-1, k]$, where d_{s,s_0} is the Euclidean distance between node s and the sink s_0. Thus, the nodes in P_k will be in the k^{th} annulus around the sink as showed in Fig. 3.2. We denote the nodes outside the transmission range of the sink as $\overline{P_1}$. The set of all the nodes which can reach the sink within j hops is denoted as $Q_j = \bigcup_{k \le j} P_k$. We use s_k to represent a node in P_k and $s_{\le j}$ to represent a node in Q_j.

3.2 Upper Bounds on Network Lifetime

3.2.1 Static sensor networks

We first identify the bottleneck sensors in a static network with one sink at the center. It is clear that the nodes in P_1 will drain their energy much faster than others, since all traffic in the network will have to pass through nodes in P_1 at least once. Theorem 1 gives the upper bound for the lifetime of the static network.

Theorem 1 *The lifetime of a dense static network is upper bounded by* $\frac{E}{R^2 e}$ *time units when there is no congestion in the network.*

Proof: When the network density λ is large enough, there would be $\rho = \pi\lambda$ nodes in the transmission range of the sink *a.s.* by Law of Large Numbers [65]. Then the total initial energy stored in P_1 would be ρE.

Suppose the lifetime of the network is $\widetilde{T} > \frac{E}{R^2 e}$. In each time unit, there would be $N = \rho R^2$ packets generated by sensors in the network, and all of them should be delivered to the sink. Since the sink can only receive data from the nodes in P_1, both the packet generated by nodes in P_1 and in $\overline{P_1}$ must pass through nodes in P_1 at least once. Thus, the total packets passing through nodes in P_1 per time unit should satisfy:

$$\sum_{s_1 \in P_1} f_{s_1} \geq N \tag{3.2}$$

The total number of packets delivered by nodes in P_1 in time $\widetilde{T}$ will be

$$\begin{aligned} U &= \widetilde{T} \times \sum_{s_1 \in P_1} f_{s_1} \\ &\geq \widetilde{T} \times N \\ &> \frac{E}{R^2 e} N \\ &= \frac{\rho E}{e} \end{aligned} \tag{3.3}$$

The total energy used by nodes in P_1 would be $U \times e > \rho E$. This contradicts our assumption that the total initial energy stored in P_1 is ρE. So the lifetime of the static network must be less than or equal to $\frac{E}{R^2e}$. ∎

Although we assume a circular network here, Theorem 1 can be easily extended to the network with arbitrary shape. For a network with area L, the network lifetime is bounded by $\frac{\pi E}{Le}$. Note that the result in Theorem 1 is the best possible lifetime over all routing algorithms.

3.2.2 Networks with single mobile relay

Theorem 1 shows that for a static network with only one sink, the network lifetime decreases as the network size increases. However, if we introduce one mobile relay into the network, the bottleneck will no longer be the neighbors of the sink, since the mobile relay can act as a "bridge" connecting nodes in P_2 to the sink in this case.

Theorem 2 *With one mobile relay, the lifetime of a dense network is upper bounded by $4\frac{E}{R^2e}$ time units when there is no congestion in the network.*

Proof: The amount of traffic passing through nodes in Q_i is at least the sum of the traffic generated in $\overline{Q_i}$, which is $N - \rho i^2$ per time unit. Since the mobile relay has a transmission range of one and it can only be at one place at a time, $\overline{Q_i}$'s traffic should be relayed for at least $i-1$ hops by static nodes in Q_i. Since the number of nodes in Q_i is ρi^2, we can bound the lifetime of the network by

$$\begin{aligned} T^1 &\leq \frac{\rho i^2 E}{(N-\rho i^2)\times(i-1)e} \\ &= \frac{i^2 E}{(R^2-i^2)(i-1)e} \end{aligned} \tag{3.4}$$

When i ($i \geq 2$) increases, the right hand side of inequality (3.4) increases monotonically and when $i = 1$, the right hand side is infinity. Therefore the least upper bound on lifetime is when $i = 2$, i.e., $T^1 \leq \frac{4E}{(R^2-4)e}$. By taking into account the traffic generated by Q_2, which also needs to pass through nodes in Q_2 at least once, we can further tighten the bound to $4\frac{E}{R^2 e}$. ∎

What Theorem 2 shows is that the mobile relay needs to stay only within a *two hop radius* of the sink in order to maximize the lifetime.

3.3 Achievable Network Lifetime

3.3.1 ARA algorithm

In this section, we construct joint mobility and routing algorithm to achieve lifetime close to the upper bound derived in Theorem 2 under the assumption that the network is dense and large.

A broad outline of the algorithm is as follows. From Theorem 2, we know that the mobile relay needs to only stay within a two hop radius in order to maximize the lifetime. Therefore the mobility pattern of the mobile relay is as follows: Starting from the sink, the mobile relay traverses a path which forms a set of concentric circles, centered around the sink with increasing radii, until it reaches the periphery of Q_2. It stays on each point on this path for a certain duration and relays traffic to the sink.

The outline of the routing algorithm is as follows. Assume that the mobile is located at position O, then all traffic in $\overline{Q_2}$ is first aggregated to points on the line

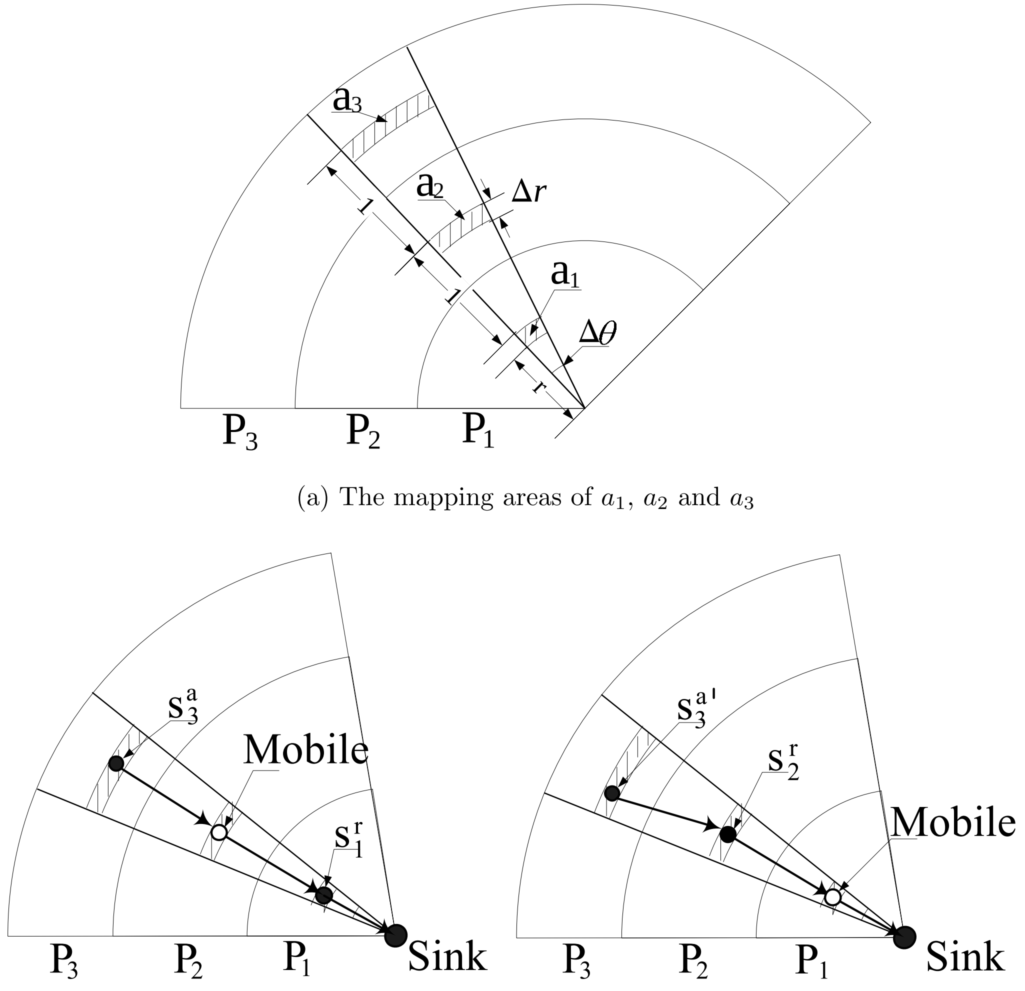
OS, where S is the position of the sink. This traffic is then directed hop by hop along the line OS until it reaches the sink. We call this routing algorithm ARA (Aggregation Routing Algorithm) for the rest of this thesis.

Theorem 3 *There exists a routing scheme which can extend the network lifetime to at least* $4\frac{E}{R^2e} - \frac{16E}{R^4e}$ *with one mobile node, when the network radius* $R > 16\pi + 4$ *and the traffic is uniform.* [2]

Proof: First, we will use the mobile relay to build 4ρ relay paths in Q_2. Each path will only contain one static node $s_{\leq 2}$ in Q_2 and each static node in Q_2 will only be used once.

Consider three arbitrary small areas a_1, a_2 and a_3 illustrated in Fig. 3.3 (a), where $0 < r \leq 1$ is the distance from the area a_1 to the sink. The distance between the three areas is 1. By our definition, the nodes in area a_i will be in the set P_i. As $\Delta\theta$ and Δr are small, any node in area a_1 can directly talk to the sink and the nodes in a_2. Also, nodes in a_2 and a_3 can communicate with each other. If we put the mobile in area a_2, we can connect a static node s_3 in a_3 with a static node s_1 in a_1, thus the packet received by s_3 can flow to the sink with the help of the mobile and s_1. Also, when the mobile is in the area of a_1, it can connect the sink with a node s_2 in area a_2 which can draw data from another node s'_3 in area a_3, as Fig. 3.3 (b) shows. Since the network density is $\lambda = \frac{\rho}{\pi}$, the number of nodes in area a_1 would be $n_1 = \frac{\rho}{\pi} r \Delta r \Delta\theta$. The number of nodes in area a_2, a_3 would be $n_2 = \frac{\rho}{\pi}(r+1)\Delta r \Delta\theta$ and

[2] Here the bound for R is loose but strictly holds under all conditions. We conjecture that bound could be tighten to $R \leq 20$, since less strict methods yields $R = 20$.



(a) The mapping areas of a_1, a_2 and a_3

(b) Connecting nodes by the mobile relay

Figure 3.3: Aggregation node selection and the relay path construction

$n_3 = \frac{\rho}{\pi}(r+2)\Delta r \Delta\theta$. For $0 < r \leq 1$, we have:

$$\begin{aligned} n_1 + n_2 &= \frac{\rho}{\pi}(2r+1)\Delta r \Delta\theta \\ &\leq \frac{\rho}{\pi}(r+2)\Delta r \Delta\theta \quad = n_3 \end{aligned} \tag{3.5}$$

which means that the number of nodes in a_3 is always bigger than the sum of that in area a_1 and a_2. Therefore, for any node in a_1 or a_2, we can associate a unique node in a_3 to it. Varying r and θ, we can cover all the nodes in Q_2, and build an injective mapping $f : Q_2 \rightarrow P_3$. This implies that for each node $s^r_{\leq 2}$ in Q_2, we can associate a unique node, $s^a_3 = f(s^r_{\leq 2}) \in P_3$ such that s^a_3 can communicate to the sink by relaying via the mobile node and $s^r_{\leq 2}$. We call the node $s^r_{\leq 2}$ the *relay node* and the node s^a_3 the *aggregation node* (in the latter discussion we will drop the subscript and just denote them as s^r and s^a). We define the range of mapping f as the aggregation set, denoted by G. What this means is that if the mobile is at a distance r_m, $0 < r_m \leq 1$ from the sink, then s^r is in Q_2 one hop from the mobile and at a distance $1 + r_m$ from the sink. A unique aggregation point s^a which is at a distance 1 from s^r and $2 + r_m$ from the sink is chosen. Similarly, s^r and s^a can be defined appropriately when $1 < r_m \leq 2$. Therefore, depending on the position of the mobile, s^r and s^a can be defined. If the mobile covers all the positions in Q_2, then every node in Q_2 will be chosen as s^r once[3]. If, depending on the position of the mobile, all traffic in $\overline{Q_2}$ is routed via the associated aggregation node s^a, then we will have 4ρ unique paths from P_3 to the sink.

Suppose that we can aggregate the packets generated by nodes in $\overline{Q_2}$ to the

[3] Since we assume a Poisson point process, the probability that two nodes fall at the same location can be ignored.

aggregation node s^a (later we will show this is possible), then there will be at most N packets passing through any of the 4ρ paths in each time unit. When a node s^a is aggregating all traffic from $\overline{Q_2}$, the rate of energy consumption for s^r and s^a will be very high. Suppose, we reserve $E' = e \times 4\frac{E}{R^2 e} = 4\frac{E}{R^2}$ in these nodes to send out the data that they sense, then they will have $E - E'$ units of energy for relaying traffic from other nodes in the network. Therefore, the mobile relay will stand in one location for a duration $\frac{E-E'}{Ne}$ time units before moving to the next location.

Since there are 4ρ routes in total, the total lifetime for nodes in Q_2 will at least be:

$$4\rho \times (\frac{E - E'}{Ne}) = 4\frac{E}{R^2 e} - \frac{16E}{R^4 e} \tag{3.6}$$

We will now construct a mobility algorithm and a routing algorithm for the nodes in the network and show that none of the nodes in the network will deplete their energy prior to $4\frac{E}{R^2 e} - \frac{16E}{R^4 e}$.

To get the lifetime of $4\frac{E}{R^2 e} - \frac{16E}{R^4 e}$, we need to aggregate the traffic generated outside Q_2 to the node s^a. Fig. 3.4 shows the mobility algorithm for the mobile relay. Note that the mobile relay remains within Q_2. Fig. 3.5 describes the Aggregation Routing Algorithm (ARA) in detail. We outline below the salient points of this algorithm.

1. **Nodes in Q_3:**

   The data generated in Q_3 will be delivered as follows: Nodes in P_1 directly send their data to the sink in one hop. Nodes in P_2 will send their data to nodes in $P_3 \setminus G$, which are the set of spare nodes which is not used as aggregation

Mobility algorithm for the mobile relay

Parameters:

(r_m, θ): the coordinate for the mobile's position in a polar coordinate system where the sink is at the origin.

Method – Mobility management:

```
Set $r_m = 0$, $\theta = 2\pi$;
while ($r_m < 2$)
  if $\theta < 2\pi$
    Set $\theta = \theta + \Delta\theta$;
    Move to the new position $(r_m, \theta)$;
  else
    Set $r_m = r_m + \Delta r$,$\theta = 0$;
    Move to the new position $(r_m, \theta)$;
  endif
  if $r_m < 1$
    while There exists an unselected node whose coordinate is $(r_m + 1, \theta)$
      Pick up one unselected node $s$ whose coordinate is $(r_m + 1, \theta)$ and $s'$ with
       coordinate $(r_m + 2, \theta)$;
      Set $s^r = s$, $s^a = s'$;
      Broadcast the information about $s^r$ and $s^a$ to the nodes in the aggregation area;
      Relay packets from $s^r$ to the sink for $\frac{E}{\rho R^2 e} - 4\frac{E}{\rho R^4 e}$ time units;
      Mark $s$ and $s'$ as selected;
    endwhile
  else
    while There exists an unselected node $s$ whose coordinate is $(r_m - 1, \theta)$
      Pick up one unselected node $s$ whose coordinate is $(r_m - 1, \theta)$ and $s'$ with
       coordinate $(r_m + 1, \theta)$;
      Set $s^r = s$, $s^a = s'$;
      Broadcast the information about $s^r$ and $s^a$ to the nodes in the aggregation area;
      Relay packets from $s^a$ to $s^r$ for $\frac{E}{\rho R^2 e} - 4\frac{E}{\rho R^4 e}$ time units;
      Mark $s$ and $s'$ as selected;
    endwhile
  endif
endwhile
```

Figure 3.4: Mobility algorithm for the mobile relay

points in P_3. Nodes in P_3 will also send their data to nodes in $P_3 \setminus G$. The task for nodes in $P_3 \setminus G$ is to redirect all the data they receive to the current aggregation node s^a using nodes in $P_3 \setminus G$ as relays.

Aggregation Routing Algorithm running on a static node $s \in P_k$
Parameters:

s^a: the current aggregation node

s^r: the current relay node

r: the distance between s^a and the sink is $r + 2$

Method – ARA:

```
switch ($k$: the index of $P_k$ where $s \in P_k$)
  case 1:
    if $s = s^r$
      Relay the received packet to the sink;
    else
      Send the sensed data to the sink;
    endif
  case 2:
    if $s = s^r$
      Relay the received packet to the mobile;
    else
      Find a neighbor in $P_3 \setminus G$ and send sensed data to it;
    endif
  case 3:
    if $s = s^a$
      Relay the received packet to the mobile or $s^r$;
    else if $s \in G$
      Find a neighbor in $P_3 \setminus G$ and send sensed data to it;
    else
      Relay the received packet towards $s^a$ using a neighbor in $P_3 \setminus G$;
    endif
  case 4, ..., $R$:
    if $d_{s,s_0} = k - 1 + r$ and $s$ is on the line $OS$
      Find a neighbor in $P_{k-1}$ whose distance to the sink is $k - 2 + r$ and send the packet to it;
    else if $s$ is on the line $OS$
      Find a neighbor on $OS$ whose distance to the the sink is $k - 1 + r$ and send the packet
       to it;
    else
      Find a neighbor which is closest to line $OS$ and has the same distance to the sink, send
       the packet to it;
    endif
```

Figure 3.5: The Aggregation Routing Algorithm

2. **Nodes in $\overline{Q_3}$:**

   (a) The nodes in $\overline{Q_3}$ will first relay the packets they generate to the line OS.

A node in annulus P_k, $k > 3$ which is at a distance $k - 1 \leq d < k$ sends the packets it sensed to a point on the line OS which is also at a distance d from the sink. It does so by relaying its traffic only via nodes which lie on the circle of radius d as shown in Fig. 3.6.

(b) Then we use the points on OS to deliver the traffic to the aggregation point s^a. For packets generated in P_k, $k > 3$, after they reach the line OS, they will first be sent to the node on OS which is at a distance $k - 1 + r$ from the sink. Then the packets are sent hop by hop, passing through nodes on line OS with distance $i - 1 + r, 4 \leq i < k$ from the sink, until they reach the aggregation point.

(c) After aggregating the traffic at s^a, we will use one mobile and one node s^r in Q_2 to build a path from s^a to the sink.

As there are 4ρ candidates for s^a, the routing table will change as the aggregation point changes. From the symmetry of the mobility and routing algorithms described in Fig. 3.4 and Fig. 3.5, it is clear that the traffic load for the nodes which lie on a circle with center at the the sink will be equal. Although some nodes may be heavily burdened for a short duration, the total traffic load for nodes on a circle centered at the sink will be equal over the lifetime of the network. In the rest of this chapter, we will frequently refer to nodes within a ring with width of Δr. We denote a ring of $[i - 1 + r, i - 1 + r + \Delta r]$ as $Ring_{i,r}$ in the following discussion.

We will investigate the energy consumption of all nodes in the network under this joint mobility and routing scheme, and show that the network lifetime will be at least

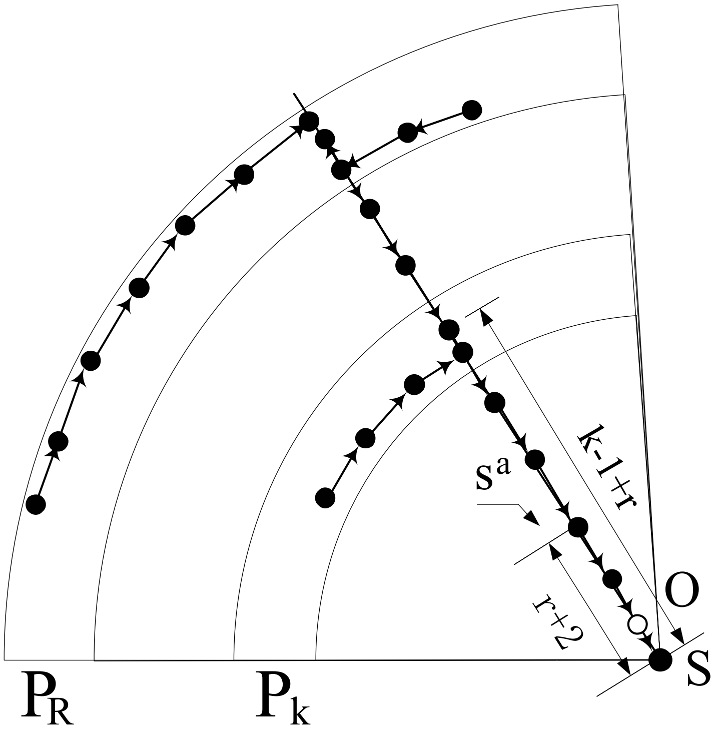


Figure 3.6: Packet aggregating routes with ARA

$4\frac{E}{R^2 e} - \frac{16E}{R^4 e}$.

1) Lifetime for nodes in Q_2:

A node in P_1 either relays traffic for the entire network or relays only its own traffic directly to the sink. For each node in P_1, we have reserved $E' = \frac{4E}{R^2}$ units of energy for transmitting its own traffic. As we discussed in the mobility algorithm, the lifetime of the nodes in P_1 is at least $\frac{4E}{R^2} - \frac{16E}{R^4 e}$. Similarly, for nodes in P_2, since they transmit the information they sense to a node in $P_3 \setminus G$ and we have reserved $\frac{4E}{R^2}$ units of energy for this, they can also live for at least $\frac{4E}{R^2} - \frac{16E}{R^4 e}$ units of time.

2) Lifetime for nodes in P_3:

Like the nodes in Q_2, the nodes in G will be only left with enough energy to send out their own data after they have acted as aggregation points.

Therefore nodes in $P_3 \setminus G$ need to carry the task of relaying the data generated by nodes in P_3 and P_2. There will be ρ nodes in $P_3 \setminus G$, and they must relay 8ρ packets to the aggregation node in each time unit, 3ρ for nodes in P_2 and 5ρ for nodes in P_3, since there are $(2i-1)\rho$ nodes in P_i. They will first deliver the packets for at most $3\pi + 1$ hops to get them to reach the line OS. Then they need one more hop to send them to the aggregation node. So, a packet from P_2 or P_3 will be relayed for no more than $3\pi + 2$ hops by nodes in $P_3 \setminus G$.

There will be $2\rho(1-r)\Delta r$ nodes in $P_3 \setminus G$ in $Ring_{3,r}$ and they will be symmetrically distributed around the ring. From the symmetry of the mobility algorithm and consequently the symmetry of the location of aggregation points, the traffic load in this ring can be evenly distributed. On average each node in $P_3 \backslash G$ will deliver packets for 8 nodes in P_2 and P_3 including itself. Here the energy consumption for delivering one packet to the aggregation point is counted in the energy budget of the first node relaying it in $P_3 \setminus G$, since the energy consumption will be evenly distributed among nodes in $Ring_{3,r}$. In order to distribute the load evenly among different rings, we need to map $16\rho(1-r)\Delta r$ nodes in P_2 and P_3 to nodes in $P_3 \setminus G$ in $Ring_{3,r}$ and deliver the packets generated by them only through nodes in this ring. Such a mapping can be built as follows: First, map P_2 and P_3 nodes to the $P_3 \setminus G$ nodes in the outermost ring of $Ring_{3,r=1}$, then decrease r and map inner rings in sequence, until we reach the innermost ring of $Ring_{3,r=0}$. When mapping nodes to a particular ring $Ring_{3,r}$, all nodes in $\{P_3 \setminus G\} \cap \{[r+2+\Delta r, 3]\}$ would have already been mapped to 8 nodes in P_2 and P_3. Therefore, when we are at $Ring_{3,r}$, total number of nodes in P_2 and P_3 which have already been mapped will be $8\int_{x=r}^{1} 2\rho(1-x)dx = \rho(8-16r+8r^2)$. The nodes in P_2 and P_3 which can communicate with a node in $Ring_{3,r}$ are in the area of $[r+1, 3]$, which has $\rho(8-2r-r^2)$ nodes in total. Since $\rho(8-2r-r^2)$ is always bigger than $\rho(8-16r+8r^2)$ for $0 < r \leq 1$, this implies that there are unmapped nodes in P_2 and P_3 which can be mapped to $Ring_{3,r}$. This holds true for every r as we go form the outmost ring to the innermost ring. Therefore, we can build a mapping from the 8ρ nodes in P_2 and P_3 to the ρ nodes in $P_3 \setminus G$, with each node in in $P_3 \setminus G$ exactly being mapped to 8 unique nodes in P_2 and P_3. Since each node in $P_3 \backslash G$ will have to relay for 8 nodes and each packet is routed for at most $(3\pi+2)$ hops. The lifetime for any node in $P_3 \setminus G$ is at least $\frac{E}{8(3\pi+2)e}$. When $R > 20$, we can guarantee the lifetime of them will be larger than $\frac{4E}{R^2 e}$.

3) Lifetime for nodes in P_k with $k \geq 4$

The nodes in $P_k, \quad k \geq 4$ will have to relay traffic for information generated in P_k and for information generated in $\overline{Q_k}$. First consider the packets generated in P_k: For nodes in $Ring_{k,r}$, the packets generated in this ring will be relayed to the line OS by nodes in this ring. Each packet will travel at most π in angle before it can reach the line OS. It also needs to be relayed for one more hop to reach some node on line OS with exactly $k-3$ distance to the aggregation point. Deducting the first hop reserved for nodes to send out its own data, the maximal hops a packet will travel in $Ring_{k,r}$ will be $\pi(k-1+r)+1$. Since the mobility and routing algorithm is symmetric, the traffic will be evenly distributed among nodes in this ring. As there are $2\rho(k-1+r)\Delta r$ nodes in $Ring_{k,r}$, in each time unit the ring will generate $2\rho(k-1+r)\Delta r$ packets. During the lifetime of the network, which is $\frac{4(E-E')}{R^2 e}$, the total energy used in delivering this part of traffic will be upper bounded by:

$$\begin{aligned} E_1(k,r) &\leq 2\rho(k-1+r)\Delta r \times \frac{4(E-E')}{R^2 e} \times e \times (\pi(k-1+r)+1) \\ &= \frac{8\rho(k-1+r)(\pi(k-1+r)+1)}{R^2} \times (E-E')\Delta r \qquad (3.7) \end{aligned}$$

The next part is the packet generated by nodes in $\overline{Q_{k-1}}$. They will be relayed for one hop to the nodes in P_{k-1}. Observe that the nodes in $Ring_{k,r}$ only will be involved in relaying this part of traffic when the distance between the current aggregation point s^a and the sink is $2+r$. As we have mentioned, there will be $2\rho(2r+1)\Delta r$ aggregation points whose distance to the sink is $r+2$. Each of them will be used for at most $\frac{E-E'}{\rho R^2 e}$ time units. Then, the nodes in $Ring_{k,r}$ will need to route traffic from $\overline{Q_{k-1}}$ for at most $2(2r+1)\frac{E-E'}{R^2 e}\Delta r$ time units. Since the aggregation points in circle $2+r$ are sequentially chosen in a clockwise direction, the traffic load on every node in the ring $Ring_{k,r}$ as a relay to ring $Ring_{k-1,r}$ will be equal. Therefore, we need to calculate the total energy used in this ring to ensure that no node in the ring will use up its energy prematurely. There will be at most N packets from $\overline{Q_{k-1}}$ passing through $Ring_{k,r}$ per time unit. So the total energy consumption during the lifetime will be upper bounded by:

$$\begin{aligned} E_2(k,r) &\leq 2(2r+1)\frac{E-E'}{R^2e}\Delta r \times N \times e \\ &= 2\rho(2r+1)(E-E')\Delta r \end{aligned} \tag{3.8}$$

Since there are $2\rho(k-1+r)\Delta r$ nodes in $Ring_{k,r}$, the total energy that can be used for relaying will be $2\rho(k-1+r)(E-E')\Delta r$. So the total residual energy for nodes in $Ring_{k,r}$ will be lower bounded by:

$$\begin{aligned} E_{re}(k,r) &\geq 2\rho(k-1+r)(E-E')\Delta r - [E_1(k,r)+E_2(k,r)] \\ &= 2\rho(k-r-2)(E-E')\Delta r - E_1(k,r) \\ &\geq 2\rho(k-3)(E-E')\Delta r - \frac{8\rho k(\pi k+1)(E-E')\Delta r}{R^2} \\ &\geq 2\rho(k-3)(E-E')\Delta r - \frac{8\rho(\pi k+1)(E-E')\Delta r}{R} \\ &= \frac{2\rho(\,(R-4\pi)k-3R-4)(E-E')\Delta r}{R} \\ &\geq \frac{2\rho((R-4\pi)\times 4-3R-4)(E-E')\Delta r}{R} \\ &= \frac{2\rho(R-16\pi-4)(E-E')\Delta r}{R} \end{aligned} \tag{3.9}$$

When R is bigger than $16\pi+4$, the total residual energy will be greater than 0. Since the traffic will be distributed evenly among nodes in $Ring_{k,r}$, the residual energy will also be evenly distributed among them and none of them will die before $4\frac{E}{R^2e} - \frac{16E}{R^4e}$.

The bound holds for all k and r, so no node in $\bigcup_{k\geq 4} P_k$ will die prematurely when $R > 16\pi + 4$.

We have shown that every node in the network will have lifetime more than $4\frac{E}{R^2 e} - \frac{16E}{R^4 e}$, thus the joint mobility and routing algorithm that we have constructed can achieve a lifetime of at least $4\frac{E}{R^2 e} - \frac{16E}{R^4 e}$ with one mobile. ∎

Theorem 3 shows that we can construct a routing algorithm which can achieve the lifetime of $4\frac{E}{R^2 e} - \frac{16E}{R^4 e}$ with one mobile. Since R^4 decays much faster than R^2, as the network radius R becomes large, the lifetime will approach 4 times that of the static network with one mobile relay.

3.3.2 ARALN algorithm

In the previous discussion, we constructed the ARA which can achieve the lifetime of $4\frac{E}{R^2 e} - \frac{16E}{R^4 e}$ when R is large enough. In ARA, every node in the network needs to know the position of the mobile and appropriately route traffic. However this implies large overheads in disseminating knowledge of the location of the mobile node to all the nodes in the network. We could of course argue that since the mobility algorithm is deterministic, it will only involve a one time dissemination of information to the nodes in the network. However, in a distributed sensor network, synchronization is extremely difficult and over time nodes in the network may make incorrect assumptions about the location of the mobile node. In this section, we show that we can construct a routing algorithm, whereby only a limited number of nodes in the network need to know the location of the mobile relay. More interestingly, we show that with this routing algorithm, we can still achieve a lifetime bound of $4\frac{E}{R^2e} - \frac{16E}{R^4e}$. We call this routing algorithm as ARALN (Aggregation Routing Algorithm with Limited Nodes).

ARALN is described in detail in Fig. 3.7. We outline the ideas of the algorithm below.

1. Nodes which are outside the circle with radius w do not need to know the position of the mobile and they can use shortest path routing algorithm to send their packets towards the sink.
2. Once the information from $\overline{Q_w}$ reaches P_w, it is relayed in one hop to the aggregation ring – $Ring_{w,r}$ in P_w, where the distance from the aggregation point s^a to the sink is $2 + r$. Once it reaches a node in this aggregation ring, it will be delivered by a series of aggregation rings – $Ring_{i,r}, 4 \leq i \leq w - 1$ until it reaches s^a. In each aggregation ring, it will be relayed around an angle ϕ_i within $Ring_{i,r}$ before it is relayed to the next aggregation ring – $Ring_{i-1,r}$. When this traffic reaches the line OS, it is then routed hop by hop along OS as before. This is shown in Fig. 3.8.
3. Packets generated by nodes in Q_w are routed as in the ARA described in Fig. 3.5.

Theorem 4 *With Aggregation Routing Algorithm with Limited Nodes, the network lifetime is lower bounded by* $4\frac{E}{R^2e} - \frac{16E}{R^4e}$*, for* $w = 22$*,* $R > 84$.[4]

[4] Here the bounds for both R and w are quite loose, we suspect we can achieve the lifetime with w near 10 and R near 40.

ARALN running on a static node $s \in P_k$

Parameters:

s^a: the current aggregation node

s^r: the current static relay node

r: the distance between s^a and the sink is $r + 2$

OS: the straight line connecting the sink and the mobile

Method – ARALN:

```
switch ($k$:the index of $P_k$ where $s \in P_k$)
  case 1,. . . ,4:
    Call ARA
  case 4, . . . , $w - 1$:
    if $d_{s,s_0} = k - 1 + r$
      if the packet is generated in $\overline{Q_{w-1}}$ and it has traveled $\phi_k$ in $P_k$
        Find a neighbor in $P_{k-1}$ whose distance to the sink is $k - 2 + r$ and send the packet
         to it;
      elseif the packet has reached line $OS$
        Find a neighbor in $P_{k-1}$ whose distance to the sink is $k - 2 + r$ and send the packet
         to it;
      else
        Find a neighbor who is closest to line $OS$ and whose distance to the sink is $k - 1 + r$,
         send the packet to it;
    else if $s$ is on the line $OS$
      Find a neighbor on $OS$ whose distance to the the sink is $k - 1 + r$ and send the packet
       to it;
    else
      Find a neighbor whose is closest to line $OS$ and has the same distance to the sink, send
       the packet to it;
  case $w$:
    if $d_{s,s_0} = k - 1 + r$
      Find a neighbor in $P_{k-1}$ whose distance to the sink is $k - 2 + r$ and send the packet to it;
    else
      Find a neighbor whose distance to the sink is $k - 1 + r$, send the packet to it;
  case $w + 1$, . . . , $R$:
    Find a neighbor who is closest to the sink, send the packet to it;
```

Figure 3.7: The Aggregation Routing Algorithm with Limited Nodes

Proof: Here we will show that all the nodes can stay alive for at least $4\frac{E}{R^2 e} - \frac{16E}{R^4 e}$ time units under ARALN described in section 3.3.1. We will also show that w is a constant and only nodes in a limited area have to know the position of mobile relay.

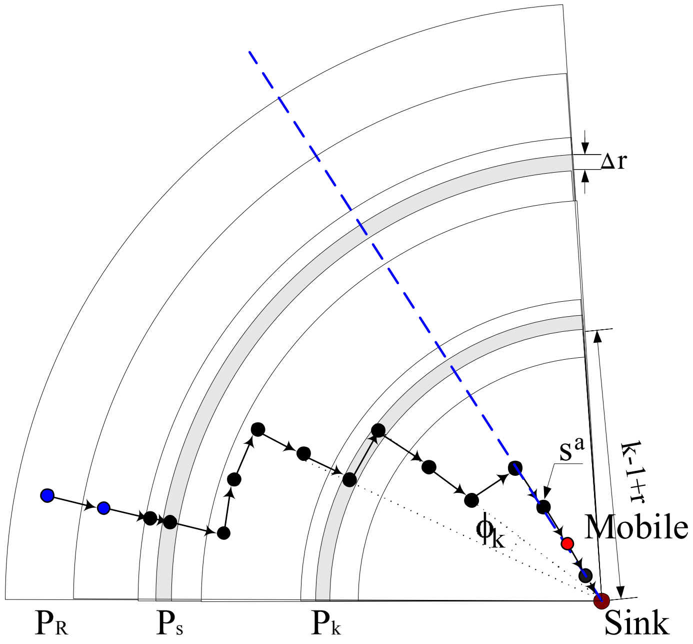


Figure 3.8: Packet aggregating routes with ARALN

1). Lifetime for nodes in Q_3

Since the ARALN in Q_3 is same as the ARA, the lifetime for nodes in Q_3 can be derived in the same way as in ARA.

2). Lifetime for nodes in P_k with $4 \leq k < w$

As in the proof of Theorem 3, the total energy consumption for relaying packets generated in the aggregation ring will be $\frac{8\rho(\pi(k-1+r)+1)(k-1+r)(E-E')}{R^2}\Delta r$. For $k \leq \frac{R}{4}$, this part is further bounded by $2\rho(E-E')\Delta r$. After the packets get to the line OS, these packets will be treated as the packets generated in $\overline{Q_w}$ and we will have to take into account the energy consumption for relaying these packets.

For the data generated in $\overline{Q_k}$, the nodes in the aggregation ring of $Ring_{k,r}$ will use their residual energy for two tasks: First, they need to forward the packets one hop in the normal direction to make them reach the nodes in the next ring $Ring_{k-1,r}$; Second, they should relay the packet in the tangential direction for a certain angle ϕ_k to deliver the traffic to the line OS.

As shown in the proof of Theorem 3, the nodes in a particular aggregation ring will only relay packets for at most $2(2r+1)\frac{E-E'}{R^2e}\Delta r$ time units. Since nodes in $\overline{Q_k}$ can not generate more than N packets per time unit, the total number of packets to be relayed by the aggregation ring in the network lifetime will not exceed $2\rho(2r+1)\frac{E-E'}{e}\Delta r$ packets.

There will be $2\rho(k-1+r)\Delta r$ nodes in the aggregation ring in P_k, so the total energy can be used in relaying for others would be $2\rho(k-1+r)(E-E')\Delta r$. Delivering $2\rho(2r+1)\frac{E-E'}{e}\Delta r$ packets to the next ring will consume $2\rho(2r+1)(E-E')\Delta r$ energy. For $k \leq \frac{R}{4}$, the nodes in the aggregation ring can spend at least

$$\begin{aligned} & 2\rho(k-1+r)(E-E')\Delta r - 2\rho(2r+1)(E-E')\Delta r - 2\rho(E-E')\Delta r \\ = \; & 2\rho(k-r-3)(E-E')\Delta r \qquad (3.10) \end{aligned}$$

energy to relay the packets in the tangential direction towards line OS. The angle ϕ_k, by which every packet generated by $\overline{Q_k}$ can travel in P_k, would be:

$$\begin{aligned} \phi_k &= \min_r \frac{2\rho(k-r-3)(E-E')\Delta r}{(k-1+r)\times 2\rho(2r+1)\frac{E-E'}{e}\Delta r \times e} \\ &= \min_r \frac{k-r-3}{(k-1+r)(2r+1)} \\ &= \frac{k-4}{3k} \qquad (3.11) \end{aligned}$$

Each packet will at most need to be relayed for an angle of π to reach the line OS. Considering that we relay the packets in discrete steps, some packets may over shoot.

For example, when the packets are routed around the circle with radius 3 (the inner circle for P_4), one hop will carry the packet for an angle of 0.335. Since the over shoot will never exceed the largest step, which is 0.335, the total angle a packet need to travel will not exceed $\pi + 0.335$. In ARALN, each packet will be relayed by ϕ_k in the aggregation ring in P_k before it reaches the line OS. For $w = 22$, the total angle packets can travel is $\sum_{k=4}^{21} \phi_k = 3.584$, which is larger than $\pi + 0.335$. So we only need to use nodes in $\bigcup_{k=4}^{22} P_k$ to aggregate the traffic. As we assumed that $k \leq \frac{R}{4}$ for $4 \leq k \leq 21$, then R should be bigger than 84 in this case.

3). Lifetime for nodes in P_k with $k = w$

For the nodes in P_w, first, they need to send their data to nodes in $Ring_{w,r}$, then the nodes in this aggregation ring will relay the packet to nodes in $Ring_{w-1,r}$. Since the nodes in $P_k, k > w$ will not know the position of the mobile, they will deliver their packets via the shortest path. The traffic passing through $Ring_{k,r}, k \geq w$ will be $\sum_{l=k}^{R} 2\rho(l-1+r)\Delta r = \rho(R-k+1)(R+k+2r-2)\Delta r$. So the nodes in $Ring_{w,r}$ should send out $\rho(R-w+1)(R+w+2r-2)\Delta r$ packets per time unit. In addition, they need to deliver $2\rho(2r+1)\frac{E}{e}\Delta r$ packets to the next ring in total. The traffic load for this task is $\frac{2\rho(2r+1)\frac{E}{e}}{4\frac{E}{R^2 e}}\Delta r$ per time unit. Then the average load per time unit for one node in $Ring_{w,r}$ will be:

$$\begin{aligned}
Load(w,r) &= \frac{\rho(R-w+1)(R+w+2r-2)\Delta r + \frac{2\rho(2r+1)\frac{E}{e}}{4\frac{E}{R^2 e}}\Delta r}{2\rho(w-1+r)\Delta r} \\
&= \frac{2(R-w+1)(R+w+2r-2) + R^2(2r+1)}{4(w-1+r)} \\
&\leq \frac{2(R-w+1)(R+w) + 3R^2}{4(w-1)} \\
&< \frac{7R^2}{4(w-1)}. \qquad (3.12)
\end{aligned}$$

As we have shown that w is much bigger than 8, the load for one node in the aggregation ring is well below $\frac{R^2}{4}$ and the lifetime of these nodes will exceed $4\frac{E-E'}{R^2 e}$.

4). Lifetime for nodes in P_k with $k > w$

For the nodes in P_k with $k > w$, they only need to relay traffic passing through them for one hop towards the sink. It is easy to see that the load for these nodes is smaller than nodes in P_w. So the lifetime of nodes in P_k with $k > w$ will also exceed $4\frac{E}{R^2 e}$. ∎

3.4 Network with Multiple Mobile Relays

In this section, we will extend our discussion to a network with M mobile relays. When we have M mobile nodes in the network, they will stay within Q_{2M} and get nearly $4M$ times lifetime when R is large.

Theorem 5 *The lifetime of a uniform dense network with M mobile relays is upper bounded by $4M\frac{E}{(R^2-4M^2)e}$ time units.*

Proof: Consider the traffic load in Q_i with $i \geq M$. The traffic generated in $\overline{Q_i}$ would be $N - \rho i^2$ per time unit. This traffic will be relayed at least $i - M$ times by static nodes in Q_i. The number of nodes in Q_i is ρi^2. Constraining the M mobile nodes to remain within Q_i and using similar arguments in the proof of Theorem 2, we can bound the network lifetime by:

$$\begin{aligned} T^M &\leq \frac{\rho i^2 E}{(N-\rho i^2) \times (i-M)e} \\ &= \frac{i^2 E}{(R^2 - i^2)(i-M)e} \end{aligned} \tag{3.13}$$

For $i < M$, such bound will be infinity. For $i \geq M$, the function $\frac{i^2}{i-M}$ takes the smallest value at $i = 2M$. So, set $i = 2M$ we get the least upper bound, which is given by $4M\frac{E}{(R^2-4M^2)e}$ time units. ∎

Notice that, here the bound is looser than the one we derived in Theorem 2.

Theorem 6 *There exists a routing scheme which can extend the network lifetime to* $4M\frac{E}{R^2e} - \frac{32\pi M^3E}{R^4e}$ *with* M *mobile nodes, when* R *is large enough.*

Proof: For the network with M mobiles, we need to form $4M\rho$ paths with nodes in Q_{2M} to reach the desired network lifetime.

Consider the areas of $a_i, i = 1, ..2M+1$ as defined the the proof of theorem 3, there would be $\frac{\rho}{\pi}(r+i-1)\Delta r\Delta\theta$ nodes in the area a_i. We need to construct $\frac{\rho}{\pi}(2r+2M-1)\Delta r\Delta\theta$ paths using M mobile with all the static nodes in $a_i, i = 1, ..2M+1$. It is obvious that there are more than $\frac{\rho}{\pi}(2r+2M-1)\Delta r\Delta\theta$ nodes in area a_{2M+1}. So, we only need to consider the nodes in $a_i, i = 1, ..2M$. For convenience, we let $\alpha = \frac{\rho}{\pi}\Delta r\Delta\theta$ in this section.

First, we randomly pickup M areas in $a_i, i = 1, ..2M$ and from a path with one static node in each and M mobiles. We form $r\alpha$ paths in this way. Then we will use the nodes in the other M areas to form another $r\alpha$ paths with the M mobile. After this, we have formed $2r\alpha$ paths, and there will be $(i-1)\alpha$ nodes left in a_i.

We use induction to show that we can form $(2M-1)\alpha$ paths using M mobiles and static nodes in $a_i, i = 1, .., 2M$, where each area have $(i-1)\alpha$ nodes:

It is easy to see that when $M = 1$ we can form α paths using static nodes from area a_1, a_2 when they have 0 and α static nodes respectively.

Table 3.1: Construction of routes with two mobiles

Area	Static nodes used (normalized by $\frac{\rho}{\pi}\Delta r\Delta\theta$)				Sum
	State 1	State 2	State 3	State 4	
a_1	r	Mobile	Mobile	Mobile	r
a_2	Mobile	r	1	Mobile	$r+1$
a_3	Mobile	r	Mobile	2	$r+2$
a_4	r	Mobile	1	2	$r+3$
a_5	r	r	1	2	$2r+3$

Suppose our claim is true for the case $M = 1, 2, \ldots, m-1$. Now we have m mobile relays and need to form $(2m-1)\alpha$ paths with static nodes in area $a_i, i = 1, ..2m$, where each area have $(i-1)\alpha$ nodes. We first use α static nodes in a_{2m} and α nodes in each of the areas form $a_2, ..a_m$ together with m mobiles to form α paths. Then we use α nodes in each of the areas form $a_{m+1}, ..a_{2m}$ together with m mobiles to form another α paths. After that we have $(2m-3)\alpha$ nodes in a_{2m}, 0 nodes in a_1 and $i-2$ nodes in $a_i, i = 2, .., 2M-1$. Changing the i by $i' = i-1$, we can apply our assumption in the area of $a_i, i = 2, .., 2m-1$: With $m-1$ mobiles we can form $(2m-3)\alpha$ paths with nodes in area $a_{i'}, i' = 1, ..2m-2$, each area has $(i'-1)\alpha$ nodes. Together with one mobile in a_1 and $(2m-3)\alpha$ nodes in a_2m, we can construct $(2m-3)\alpha$ paths connecting $a_1, .., a_{2m}$. Add this up with the first 2α paths we formed, we have constructed $(2m-1)\alpha$ paths from m mobile nodes and all the static nodes in area $a_i, i = 1, ..2m$, where each area has $(i-1)\alpha$ static nodes.

By induction, for any M we can always form $(2M-1)\alpha$ paths using M mobiles and static nodes in $a_i, i = 1, .., 2M$ with each area have $(i-1)\alpha$ nodes.

Then the paths formed in $a_i, i = 1, .., 2M + 1$ areas will be $\frac{\rho}{\pi}(2r + 2M - 1)\Delta r \Delta\theta$. Integrating in the area will give the total number of paths as $4M\rho$.

Table 3.1 gives a example of how to form $\frac{\rho}{\pi}(2r + 3)\Delta r \Delta\theta$ from nodes in $a_i, i = 1, .., 5$ with two mobile relays. Here the mobility pattern is more complex than the single mobile case. To fully exploit the energy stored in static nodes in Q_4, the two mobile relays need to cooperate with each other when moving. In general, they should stay on the same line connecting the sink and the the aggregation point n_a. Furthermore, the mobile relays should move along this line to form enough paths along this line with static nodes. When all the static nodes on this line have been used as relay nodes, they will shift around a small angle to another line. The mobile relays will sweep around the area of Q_4 in this pattern until all the static nodes in Q_4 have been used as relay node.

The routing algorithm for the M mobile relays is similar to the single mobile one. The only difference is that in the M mobile case, the packets generated in Q_{2M} will be delivered in a different manner. We need to use the node in the ring $[k - 1 + r, k - 1 + r + \Delta r], 1 < k \leq 2M$ to relay the packet generated in this ring to the line OS. A packet will travel at most $\pi \times 2M$ hops to that line.Thus, we need to reserve energy of $E'' = 2\pi M \times e \times 4M\frac{E}{R^2 e} = 8\pi M^2 \frac{E}{R^2}$ for nodes in Q_{2M}. Since there are $4M\rho$ paths, each can sustain $\frac{E-E''}{Ne}$ time units, the total network lifetime would be $4M\frac{E}{R^2 e} - \frac{32\pi M^3 E}{R^4 e}$. ∎

3.5 Chapter Summary

In this chapter, we considered the lifetime upper bound of dense wireless sensor networks. We first introduced a circular network model with one sink at the center. Using this model, we derived the upper bound for static networks and networks with one mobile relay. We showed that a single mobile relay can at most improve the network lifetime by 4 times. Then, we constructed two jointly mobility and routing algorithms ARA and ARALN which can asymptotically achieve the upper bound of lifetime for networks with one mobile relay. We also showed that the network lifetime can be asymptotically improved by $4M$ times when there are M mobile sensors in the network.

The analysis and algorithms in this chapter is based on an ideal assumption that the network is very dense. However, sensors are often deployed in low density in real networks. In the next chapter, we will further study the network lifetime in network with finite density. We will use numerical results to show the performance of our mobile relay scheme in randomly deployed networks and compare it to other approaches.

Chapter 4

Mobile Relay in Low Density Networks

4.1 Problem Formulation

In this chapter we will investigate the mobile relay approach in random networks with moderate size and relax the assumption of dense and large network in Chapter 3.

For a random network with moderate size, such as 100 sensors randomly deployed in a 5×5 hops area, the randomness of sensor distribution may generate topology defects, i.e., voids or low density areas in the network. Such topology defects prevent the construction of perfect symmetric routing as described in Chapter 3. However, the experiments on random networks show that the routing schemes in random finite networks still follows the basic routing concepts as in the ideal networks. The experiment results also show that the mobile relay can improve the network lifetime by more than two times for networks with such topology defects.

We first construct an optimization problem to solve the routing and mobility problem in the random finite network with only one mobile relay. The network topology is abstracted as a *Random Geometric Graph* with edge (i, j) between any pair of vertex i and j of distance smaller than 1 from each other. We maximize the overall network lifetime under the energy constraints of static sensors. Similar to the assumptions used in [32] for mobile sinks, we assume that the mobile relay will stay at positions where there is a static sensor. When the mobile relay is at the position of sensor k, it will take over the task of sensor k and sensor k will sleep for that time period. The mobile relay will shift between static sensors and try to help as many sensors as possible during the network lifetime. We assume that the mobile relay will always stay at the position of some static sensor during the network lifetime, since this will always give a longer lifetime than removing the mobile node. Thus, maximizing the sum of periods for which the mobile relay stays at each location will give the optimal network lifetime. Here, we ignore the time used for moving the mobile nodes, which will be further discussed in section 4.3.3.

We formulate the linear optimization problem for maximizing the network lifetime as follows:

$$\begin{aligned} & Maximize & & \sum t_k & (4.1) \\ s.t. \quad & \sum_j x_{ij}^k - \sum_j x_{ji}^k & = \; & g_i \times t_k \quad \forall i, k & (4.2) \\ & x_{ij}^k & \geq \; & 0 \qquad \forall (i,j), \forall k & (4.3) \\ & \sum_{\forall k \neq i} \quad \sum_j x_{ij}^k \times e & \leq \; & E \qquad \forall i \neq 0 & (4.4) \end{aligned}$$

where t_k is the length of time that the mobile relay will stay at sensor k and x_{ij}^k is the total traffic flow from sensor i to sensor j during that time period. E and e represent the initial energy of the sensor nodes and the energy for relaying one packet, respectively. g_i is the packet generation rate for node i. We assume the data generation rate is uniform, i.e., $g_i = 1$, in all experiments. Then, the data generation rate for the sink, denoted as g_0, is set to $-N$, where N is the number of sensors in the network. Constraints (4.2) and (4.3) give the network flow constraint for node i during the period t_k, which restricts the difference between total out-flow and the total in-flow during the period must be equal to the packets node i generated in this period. Constraint (4.4) is the energy constraint, namely, the sum of energy over all the periods should not exceed the initial energy E. The energy used in period t_i is not counted, since during t_i the mobile is at the position of sensor i and sensor i is sleeping.

In this linear programming formulation, we assume a uniform energy depletion model where each packet consumes the same amount of energy. The channel access overheads, such as backoffs and retransmissions, are ignored in this model. However, Eq. (4.4) can be used in general nonuniform energy depletion models. Once the detailed energy consumption model under a specific channel access protocol is given, the energy consumption rates on different edges in Eq. (4.4) can be changed accordingly. However, the study of energy consumption model under different channel access protocols are out of the scope of this thesis. In section 4.4, we will give a nonuniform energy consumption rate sample where the energy consumption rates depend on the distance between nodes.

On average there are $O(\rho N)$ edges in the static network and the mobile can appear in N different places. So, there are $O(\rho N^2)$ variables and $O(N^2)$ constraints in the linear programming problem, which is solvable in polynomial time.

In practice, the sink can be the entity who collects traffic information and calculates the optimal schedule. Once the optimal schedule is know, the sink will broadcast the schedule to all sensors in the network. A static node will know when the mobile will take over its task and sleep during that time period. The static node also knows when the mobile will leave and will wake up slightly before that time to take over the relay tasks.

The energy used for collecting traffic information and broadcasting the movement schedules are not considered in the linear programming. As the algorithm only executes once after deployment, the energy used in these operations will be negligible when compared to the energy used for relay packets during the network lifetime, which can be serval months to years.

Our numerical results is based on the simplified energy model stated in Chapter 3 without considering the MAC or physical layer. The sensors are randomly deployed on fields with different size and shape. For each network instance, we calculate the lifetime for the static network through the linear optimization algorithm as described in [34], which gives the optimal lifetime for the static network. The lifetime of the mobile relay solution on the same network instance will also be calculated and we averaged over at least 100 network instances to get the improvement.

4.2 Performance of Mobile Relay

4.2.1 Traffic distribution

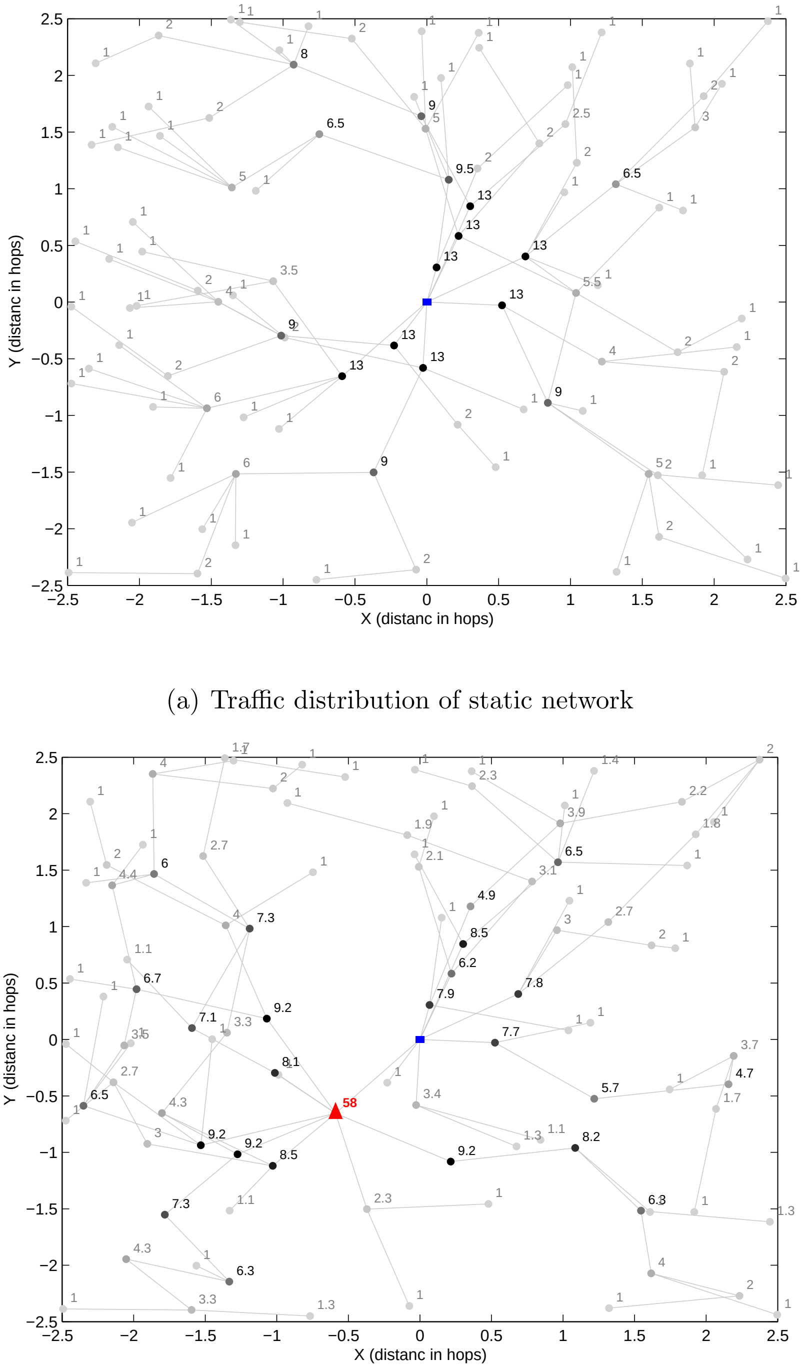
Fig. 4.1 shows the traffic distribution for a randomly deployed network. The network contains 100 randomly deployed sensor on the 5×5 hops square area with one sink at the center. We can see there are topology defects in this network, like the hole up-left to the sink. Fig. 4.1(a) describes the optimized traffic distribution of the static network, where the numbers on the sensors indicate the amount of traffic passing through it per time unit. Since the traffic flow is optimized by energy conserving routing, the neighbors of the sink shares the same load of 13 packets per time unit. However, such load is much higher than other sensors. Thus, those immediate neighbors of the sink will die first and the sink will be disconnected from other parts of the network after that.

Fig. 4.1(b) and Fig. 4.1(c) show the traffic routes with one mobile relay in the network[1]. The mobile relay will stay at different places (represented by triangles) during different time period. The traffic flow in each time period will be optimized to maximize the overall network lifetime. Although we do not use the routing schemes developed in Chapter 3, the routes obtained through optimization still follows the ideas used in Chapter 3:

1) The mobile relay stays within two hops range of the sink even though we do not explicitly oblige the mobile to stay around the sink. The experiments with and without such moving range constraints provide similar network lifetime [10]. This shows the mobile relay actually only needs to stay in two hop distance around the sink to maximize the network lifetime.

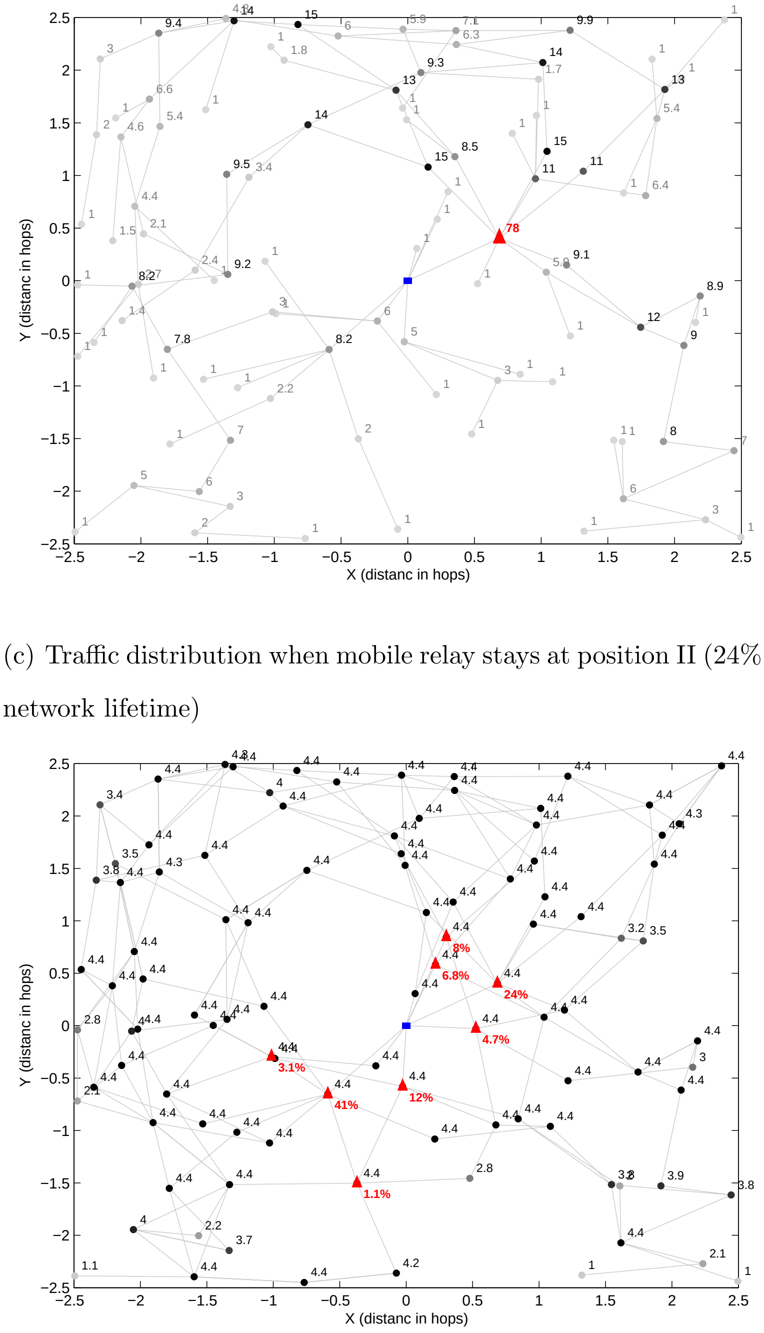
2) Sensors should redirect their packets towards the mobile relay so that the mobile relay can carry more traffic. Also, the aggregation process will mostly use the sensors

[1] In this example we only show two positions and the correspond traffic flow, while there are the 8 positions the mobile relay will stay during the network lifetime.



(a) Traffic distribution of static network

(b) Traffic distribution when mobile relay stays at position I (41% network lifetime)



(c) Traffic distribution when mobile relay stays at position II (24% network lifetime)

(d) The traffic flow averaged over the network lifetime with one mobile relay. Triangles are positions where the mobile relay will stay and the numbers below it are the length of stay

Figure 4.1: Traffic flow in network with 100 sensors in 5×5 hops square area. The sink is marked as the square in the center of the network. Mobile relay is marked as triangle.

on the edge of the network. We can see from Fig. 4.1(b) and Fig. 4.1(c), that nearly 70% packets are flowing through the mobile relay.

3) Sensors will work as the static relay node alternately. There exist some time periods that a particular sensor will be highly loaded, for example, some sensors need to carry 15 packets per unit time in Fig. 4.1(c). However, after they perform their relay task, they seldom need to relay for others in the rest of their lifetime. Such alternation reduces the overall traffic load for sensors. As shown in Fig. 4.1(d), most sensors share the average traffic load of 4.4 packets over the total network lifetime after we put in the mobile relay. This maximal traffic load is much smaller than the maximal load in the static network case, so the network can sustain for a much longer time. Also, only a few sensors have traffic loads smaller than 4.4, which means all the sensors will use up their energy almost at the same time, thus there is little waste in residual energy.

4.2.2 Network lifetime

Fig 4.2 shows the lifetime for static circular networks and the lifetime with one mobile relay under different network sizes. The lifetime is normalized by the maximal node lifetime of $\frac{E}{e}$. The experiment result of the static network fits the theoretical result in Theorem 1 quite well.

Fig. 4.3 shows the lifetime improvement of the mobile relay solution under different network settings. In general, the improvement is more than two times for networks with more than 100 nodes, irrespective of whether the network shape is circular or

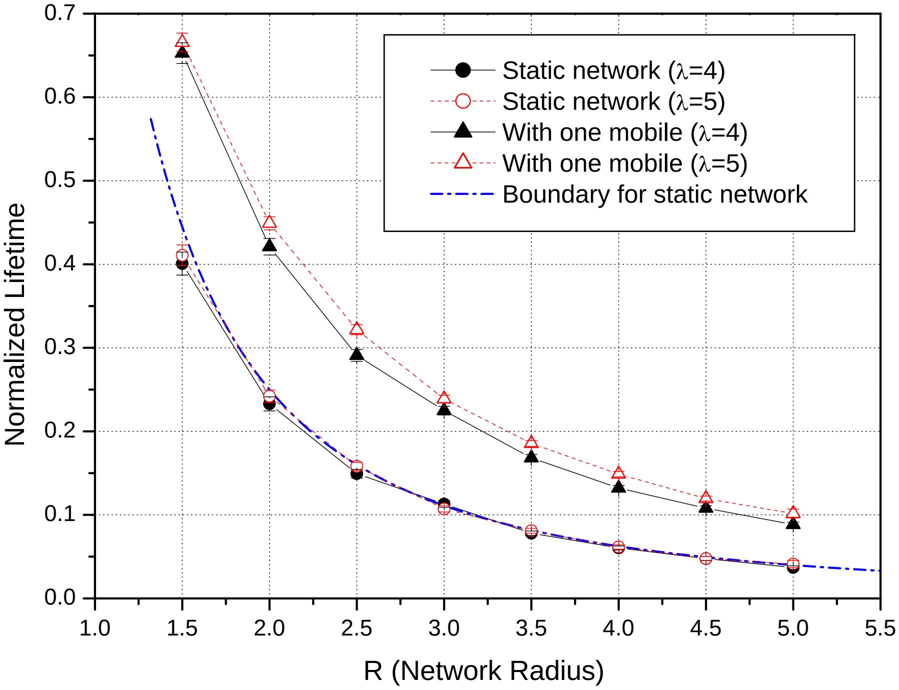


Figure 4.2: Network lifetime for nodes randomly deployed on a circular region, for $\lambda = 4$ and 5 (confidence interval 95%)

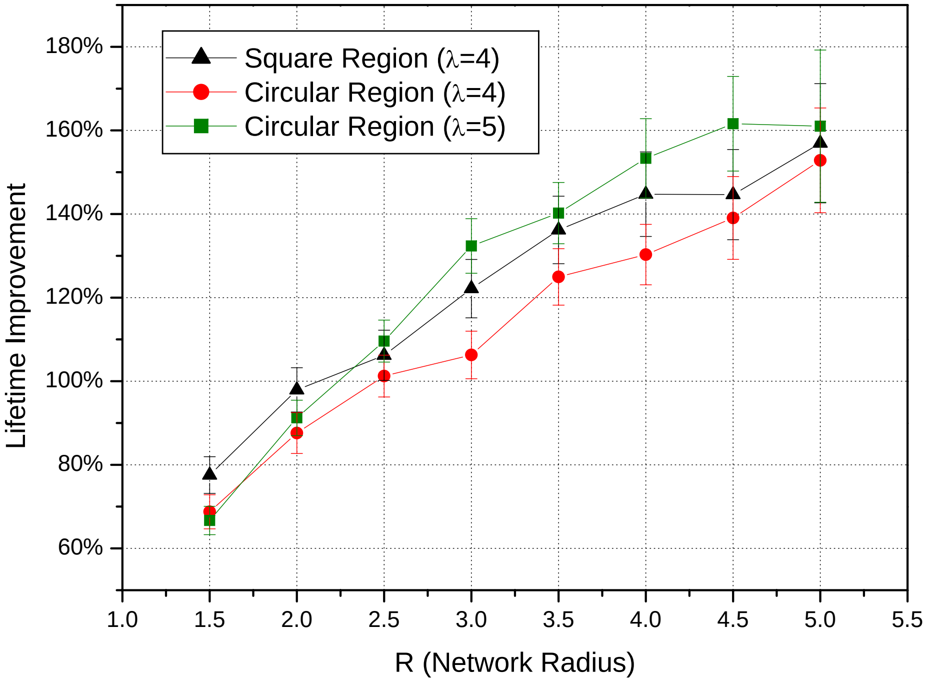


Figure 4.3: Average lifetime improvement for networks with one mobile over the static network (confidence interval 95%). For the square network, the sink is put at the center of the square and the network radius R is defined as half the side length of the square.

square.

We also observed that the improvement increases as the network size and density increases. This coincides with the analysis in Chapter 3. The density and size of the network are the two factors which limits the performance of mobile relay in the random network. With low network density, the chance that there are topology defects around the sink is very high. Such defects will limit the number of routes we can build across the two hop area around the sink. Another factor is the network size. With the limited network size, only part of the traffic can be redirected and pass through the mobile, which limits the efficiency of the mobile relay. As Fig. 4.1 shows, all nodes in the network participate in the traffic aggregation, still there is not enough space for all packets to be aggregated to the mobile relay. As the network density and size grows, the effect of topology defects and limited space will be mitigated so that the lifetime improvement will be increased.

4.2.3 Network dilation

Another important performance metric is the average path length, which affects many metrics including the average delay. As our scheme requires the traffic to be redirected towards the mobile relay, packets will take a longer route compared to shortest path routing. However, the path dilation is bounded by $\pi+1$ in our scheme, since the packet will at most go around an anger of π and then directly go towards the sink. Fig. 4.4 compares the average hop count of mobile relay solution with other schemes. The results show that energy conserving routing does not incur much route dilation and

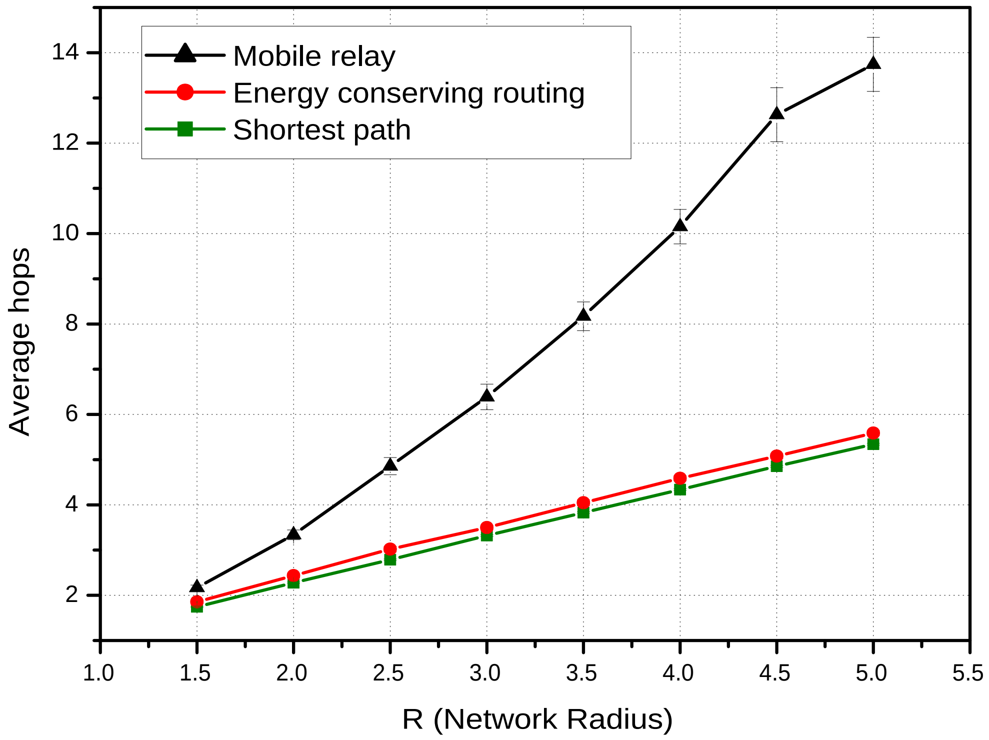


Figure 4.4: Comparing the average hops for different approaches. (Network on square area with $\lambda = 4$,confidence interval 95%)

its average hop count is quite close to the shortest path case. The mobile relay scheme use almost two times more hops to deliver the packet. As the network size grows, the path dilation converge to near 2.6 in our experiment. So, the path dilation is bounded for the mobile relay scheme. Consequently, the packet delay of our algorithm will be comparable to the delay of shortest path routing when there is no congestion. Note that the delay of mobile relay scheme is much smaller compared to data ferry solutions where the delay depends on the moving speed of the mobile [24].

4.3 Comparing Mobile Relay with Other Approaches

4.3.1 Network lifetime comparison

Fig. 4.5 compares the network lifetime for different approaches. We average over 100 realizations of a random network with 100 nodes and one sink in a 5×5 square area. In this figure, the lifetime of non-energy-aware shortest hop routing networks is normalized to 1. As we can see, the energy conserving routing scheme can improve the network lifetime nearly 4 times over the minimal hop routing. This scheme gives the upper bound for homogenous static network. By adding only one mobile relay, we can get double lifetime than energy conserving routing. However, when we use static energy provisioning schemes, e.g., adding 4 times more energy to 25% random selected static sensors, the network lifetime can be only extended by 40%. The discussion below will show that mobile relay is better than most static energy

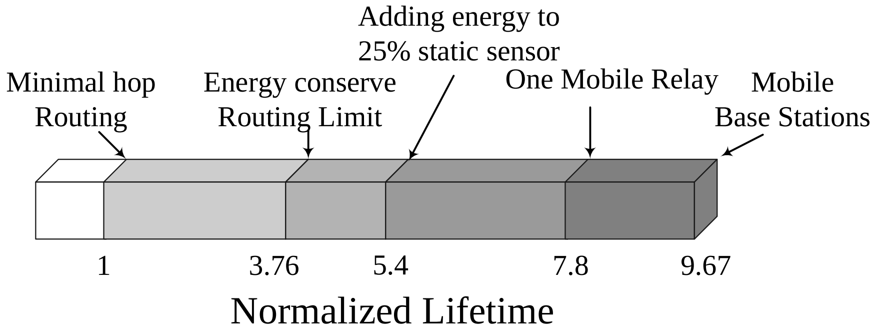


Figure 4.5: Comparing network lifetime for different approaches

provisioning approaches. Another method to extend lifetime is to use the mobile node as the sink. It gives lifetime improvement even better than the mobile relay. However, the mobile sink approach has certain constraints. We will further discuss the tradeoffs in choosing between mobile relay and mobile sink in the following section.

4.3.2 Static networks

i) Increasing the density of static nodes

One way to increase the network lifetime is to redeploy more static nodes in the area near the sink. These additional static nodes serve as reservoirs of energy. Normally these nodes are in the sleep state. If the sensors nearby dies, these additional nodes will wake up to take over the relay tasks. To achieve a lifetime improvement of 4 by this approach, we need to increase the density of nodes within two hop radius of the sink. It is easy to see that we need to deploy at least 4ρ additional static nodes around the sink to achieve the same performance as one mobile relay.

ii) Increase the energy carried by static nodes

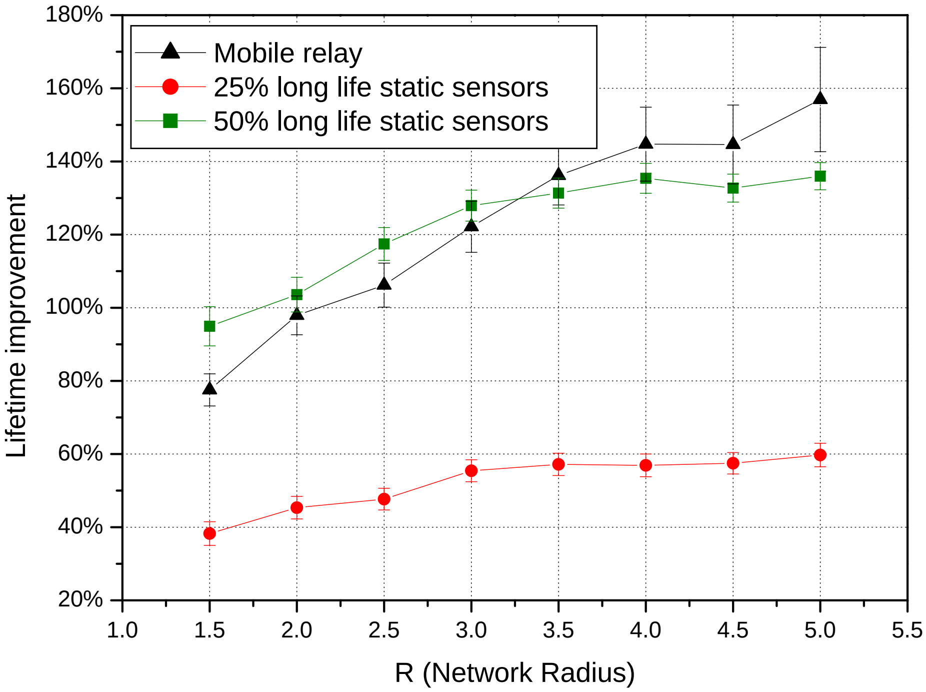


Figure 4.6: Comparing the lifetime improvement of adding one mobile relay with solutions that add more energy to static sensors. (Network on square area with $\lambda = 4$,confidence interval 95%)

We also can upgrade the battery of static sensors to extend their lifetime. Assume that we have some static sensors equipped with better batteries, which provide four times more energy than normal sensors. In the deployment phase, those more powerful static sensors are mixed up with normal sensors and randomly deployed on the field. Therefore, the network lifetime will be extended since some of the static sensors gain more initial energy. Fig. 4.6 compares the lifetime for such hybrid static network with networks have one mobile relay. The lifetime improvement of the mobile relay solution is close to the static network where 50% sensors has four times more energy. However, in the mobile relay solution we only need to add one mobile relay in the network, which may be much cheaper than the static solutions where we must upgrade the battery of 50% sensors to gain the similar performance.

iii) Resource Rich Static Nodes

Instead of mobile nodes, we can also add resource rich static nodes as relays or sinks. For the static relay case, we add static sensors with more energy but has the same communication ranges as normal sensors. For the static sink case, we assume their are multiple static sinks in the network.

As shown in the mobile relay case, given a limited number of resource rich static nodes as relays, the best location for these relays will be near the sink such that they can take over the relay tasks of the critical sensors. Therefore, the amount of traffic flowing through the critical nodes can be reduced. By adding the static relays, the new set of bottleneck nodes will be the neighbors of the sink and neighbors of the resource rich static nodes. For example, consider Fig. 4.7, assume we add two static

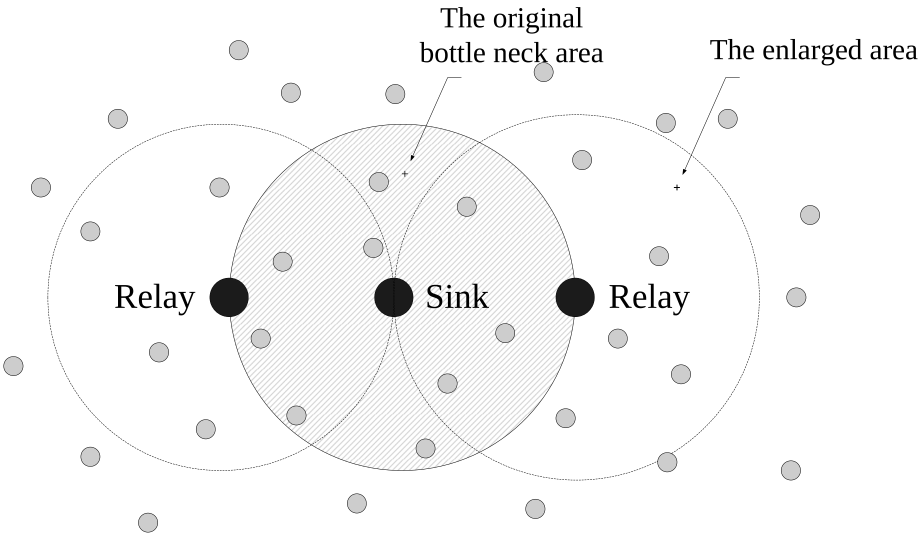


Figure 4.7: Adding static relays to improve the network lifetime

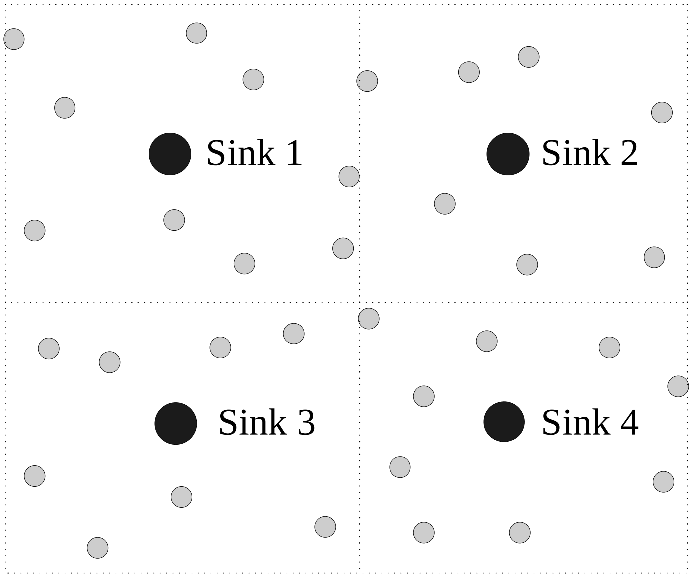


Figure 4.8: Adding static sinks to improve the network lifetime

relays which can directly communicate with the sink. Some of the relaying burden of the bottleneck nodes around the sink is mitigated. In other words, in this new network, the set of bottleneck nodes which determine the lifetime of the network will be the nodes which are one hop neighbors of the sink and the two relays. Each of these new bottleneck nodes will carry less traffic than in the original network, thereby improving the lifetime of the network. Therefore, one can view, the addition of these static relays as increasing the set of bottleneck nodes in the network, but reducing the network load on each of them. Since the static relay must be connected to the sink or other relays, adding one static relay can wield its influence over an area of at most 0.6090π, as can be seen in Fig. 4.7. Thus, one static resource rich relay can increase the network lifetime by at most 60% over the original network, while one mobile relay can increase the lifetime by 300% as we have shown previously.

Another way to use static resource rich sensors is to use them as sinks. By adding more static sinks, we can scale down the size of the sensor network. As shown in Fig. 4.8, by adding three more static sensors, we can split the original large network into four smaller networks. Sensors only need to send their data to the closest sink, thus the traffic is largely reduced. As we can see, the sensors around the static sinks will still be the bottleneck. Since each small network only contains 1/4 sensors than the original network, the traffic flow through the bottleneck sensors will be reduced to 1/4. The newly added sinks still have ρ neighbors as the original network, so the lifetime of the new network will be 4 times of the original due to the reduce in traffic. As we shown in Chapter 3, adding one mobile relay can achieve 4 times improvement in the ideal case, which is similar to adding 3 more static sinks.

By investigating different static approaches, we see that mobile relay is better than most static approaches in the sense of lifetime improvement. However, static solutions also have certain advantages since their routing schemes can be very simple and the routing overhead is low.

4.3.3 Networks with mobile sinks

The mobile sink approach use the mobile node as the sink. As the mobile node goes around the network, the bottleneck around the sink is distributed over the whole network. As shown in Fig. 4.5, the mobile sink approach in [9] can provide longer lifetime than the mobile relay approach. When the sink is moving around the network peripheral, the mobile sink approach can give an improvement factor of $O(R)$ on network lifetime, while we need $O(R)$ mobile relays to achieve the same improvement [10].

However, there are certain tradeoffs between the mobile relay and mobile sink solution. First, the mobile relay solution is more flexible than the mobile sink. The mobile sink is the gateway to the outside network, certain applications may not permit the sink to be mobile. Also, mobile sink may create more routing overheads. For a realtime data logging application, the mobile sink should always be connected to the sensors. So, each single movement of the sink must be traced and the data gathering path should be changed accordingly. For the mobile relay approach, the network is fully functional even without the mobile relay. So, the mobile relay can detach form the network, move, then reattach at the new position. Such flexibility in movement will not incur as much overhead as the mobile sink. Second, the mobile relay can heal the topology defects in the network. Due to the randomness in deployment, there may be topology defects in the network, i.e., the network is partitioned in the beginning or the communication to certain part of the network relies on few sensors, as shown in Fig. 3.1. The mobile relay can improve the network lifetime greatly in this case, while the mobile sink can not. Finally, the mobile relay requires only a small moving range, also only sensors in limited area need to know where the mobile relay is. For the mobile sink approach, the lifetime improvement depends on the moving range of the mobile sink. If the sink only moves in a small area, the lifetime improvement will be similar to mobile relay approach.

From the discussions, we see that mobile relay only need to move in a small region around the sink. Also, the mobile relay do not need to move continuously as in the mobile sink case. The mobile relay will only move between several positions in the network as shown in Fig. 4.1(d). Therefore, the overall movement distance is small and the moving time can be ignored compared to the network lifetime of serval months to years.

4.4 Power Controlled Networks

4.4.1 Problem formulation

Although we assume that sensors do not use power control in transmission in the above analysis and numerical results, the optimization framework in section 4.1 can be easily extended to scenarios where sensors can reduce the transmission power when they are close to each other.

Assume that sensors select the transmission power according to the distance to the packet destination. In other words, when a relay node is closer to the next hop node, it can reduce the transmission power level to be just “enough” to reach the next hop node. In this case, the energy consumption for relaying one packet from sensor i to sensor j will depend on the distance between i and j. As the static sensors do not move, the relay energy for a particular link will not change during the network lifetime. So, we can denote the energy for relaying one packet from sensor i to sensor j as a constant e_{ij}.

Assume that mobile sensors will still stay at the positions of static sensors. The linear programming problem of Eq. (4.1) – (4.4) will be changed to:

$$\begin{aligned} Maximize \qquad & \sum t_k && (4.5)\\ s.t. \quad \sum_j x_{ij}^k - \sum_j x_{ji}^k &= g_i \times t_k \quad \forall i,k && (4.6)\\ x_{ij}^k &\geq 0 \qquad \forall (i,j), \forall k && (4.7)\\ \sum_{\forall k \neq i} \quad \sum_j x_{ij}^k \times e_{ij} &\leq E \qquad \forall i \neq 0 && (4.8) \end{aligned}$$

As e_{ij} depends only on the distance between sensor i and j, this linear programming problem has a similar complexity as the linear programming problem of Eq. (4.1) – (4.4). Note that when nodes can do power control, they may choose to transmit over a long edge to reduce the number of hops or choose to use several short hops to save the transmission energy used per hop. Both of these choices are captured in the linear programming described above. The optimal solution will automatically use the most energy efficient route.

In this work, we do not consider physical layer interference between neighboring nodes. When considering physical layer interference, the energy used for relaying one packet, e_{ij}, depends on the detailed channel access model, which is out of the scope of this thesis. A detailed discussion on the tradeoffs between transmission power control and interference to neighboring nodes can be found in [66].

4.4.2 Numerical results

We use numerical results to study the performance of power controlled networks. In the numerical experiments, we use a simplified model for power consumption. Assume that the distance between sensor i and j is $d_{i,j}$. Then the energy for relaying a packet from node i to node j is set to $e_{ij} = e * d_{i,j}^{\alpha}$. We set $\alpha = 2$ in the experiments. Note that we assume that the maximal transmission range is 1 for all sensors in previous discussions. So, the maximal relay energy is upper bounded by e which is the power consumption when sensors are not using power control. In this experiment, we randomly deploy static sensors in circular networks as in section 4.2. The network lifetime is also averaged over 100 networks.

Fig. 4.9 shows the lifetime for static networks and the lifetime with one mobile relay under different network sizes. The lifetime is normalized by $\frac{E}{e}$ as in section 4.2. Comparing Fig. 4.9 and Fig 4.2, we have following observations:

- The network lifetime is increased when sensors are using power control, for both the static network and network with one mobile relay. This is because

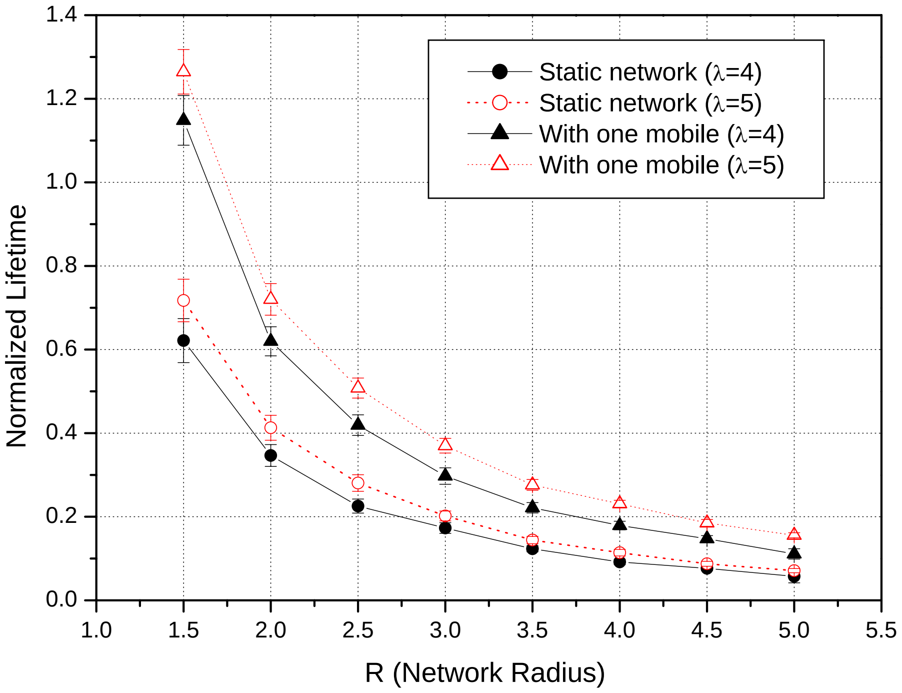


Figure 4.9: Network lifetime for nodes randomly deployed on a circular region when sensors are using power control, $\lambda = 4$ and 5 (confidence interval 95%)

the relay energy used in the power controlled networks is always smaller than or equal to that in the networks without power control.

- In power controlled static networks, the network lifetime increases as the network density gets higher, which is not observed in networks without power control. This is due to the fact that a higher sensor density will decrease the average distance between sensors. In power controlled networks, reducing the distance between sensors normally leads to a reduction in transmission energy. So, the network lifetime increases with the network density in power controlled networks.
- In the power controlled networks, the mobile relay can still provide about 2 times increase in network lifetime, which is similar as in the network without power control. This shows that our scheme can also work well when power control is used.

4.4.3 Other extensions

Our solution can also be extended to cases where mobile sensors have longer transmission ranges, possibly via directional antennae. When the mobile relay uses directional antenna with transmission and receiving gain of G. The transmission and receiving range of mobile relay can be improved to $G^{1/\alpha}$, where α is the path loss exponent. We can show that the network lifetime increase can be improved by $2(G^{1/\alpha} + 2G^{2/\alpha} - 1)$ in this case when there is a single mobile relay in the network.

4.5 Chapter Summary

In this chapter, we further investigated the performance of the mobile relay scheme in networks with finite sensor density. We first proposed a linear programming algorithm to find the best traffic route and sojourn time for mobile relays. In the numerical results, we showed the network traffic distribution, network lifetime and network dilation of the mobile relay approach. Compared to the lifetime improvement upper bound of 4 times derived in Chapter 3, the lifetime improvement in networks with finite density is around 2 times. This performance gap is possibly due to the randomness in deployment.

We also compared the network performance of the mobile relay scheme to other approaches such as mobile sink, static sink and high energy static nodes. We showed that the mobile relay scheme is better than other static network approaches and has certain advantages over the mobile sink approach. Finally, we considered the network where sensors can use power control and showed that our mobile relay scheme can also work well in networks with power control.

Chapter 5

Trade-off Between Coverage and Mobility

Coverage problem is an important problem in wireless sensor network since it determines how well the sensor network can monitor the physical world. A critical aspect which determines the quality of coverage is network deployment. Due to a variety of factors, such as the scale of the network, inaccessibility of the terrain etc., optimal deterministic deployment of the network is often infeasible. A common scenario envisioned for deployment is that of randomly scattering sensor devices over the field of interest. Although this eases the task of network deployment, it makes the task of guaranteeing coverage much harder.

In this thesis, we define a metric, *over-provisioning factor*, which indicates the efficiency of a network deployment strategy. For a given network deployment strategy, if a sensor density of λ is required to guarantee k-coverage, then we say that the deployment strategy has an *over-provisioning factor* $\eta = \frac{\lambda}{k}$. Consider a random deployment strategy with static sensors of sensing range $r = \frac{1}{\sqrt{\pi}}$ over a square region of area L. Then to guarantee k-coverage, we need sensor density $\lambda = \log L + (k+2)\log\log L + c(L)$ with $c(L) \to +\infty$ when $L \to +\infty$ [6, 7, 8]. Since $c(L)$ can grow slower than $O(\log\log L)$, the over-provisioning factor is $\eta = \Theta(\frac{\log L}{k} + \log\log L)$. Compared to a deterministic deployment, which has

$\eta = \Theta(1)$, the random static deployment has an unbounded over-provisioning factor as network size L grows. Loosely speaking, for the random deployment, many areas in the field will have far more than k sensors covering them, while a few critical regions will have around k sensors covering them. Consequently, the random deployment strategy has a high over-provisioning factor and low efficiency for large networks.

As an alternative, mobility can be used to improve network coverage efficiency [11, 25]. Mobile sensors can relocate themselves to heal coverage holes in the network so that the randomness in sensor deployment can be compensated. Clearly, the over-provisioning factor for a network with all mobile sensors can be $\Theta(1)$ since the sensors have the flexibility of relocating themselves to the optimal locations. Unfortunately, this extra degree of freedom does not come cheap. First, mobile sensors are far more expensive than static sensors. Second, mobility consumes more energy than communication or sensing. However, most research in mobile sensor networks do not consider the cost of movement for mobile sensors. If a mobile sensor is required to move over long distances, then its entire energy supply may be depleted in locomotion. Moreover, the redeployment process may take considerable time in large networks since the speed of mobiles is limited.

In this chapter, we will introduce a new way to improve the network coverage with mobile sensors. Unlike previous mobile sensor networks, the mobile sensors used in our scheme only have limited mobility. We will derive the upper bound on movement distance for mobiles to fully cover the field. We will also study the performance of a hybrid network where only a small portion of sensors are mobile in this chapter.

5.1 Coverage Efficiency in Sensor Networks

5.1.1 System Model

Consider a square sensing field with side length l and area $L = l \times l$. We assume that there are $N = \lambda L$ static sensors uniformly and independently scattered in the network. When N is large , the number of static sensors in a region with area of A, which is denoted as n_A, will be Poisson distributed with mean of λA [67]:

$$\mathbb{P}\{n_A = i\} = \frac{(\lambda A)^i e^{-\lambda A}}{i!} \tag{5.1}$$

Also, the number of sensors in disjoint areas will be asymptotically independent to each other ([67], page 39). Thus, our point process can be approximated by a stationary Poisson point process when the network is large enough. In later derivations, we directly use the properties of Poisson point processes, since we study large networks where these assumptions are valid.

We assume that each static sensor can cover a disk with radius $r = \frac{1}{\sqrt{\pi}}$ centered at it. In other words, every sensor can cover a disk with *unit area.* The field is said to be k-covered when every point in it can be covered by at least k sensors. The communication range for sensors is assumed to be larger than $2r$ so that the network will be connected when it is completely covered [68].

We also assume that mobile sensors are uniformly and independently scattered in the network and the total number of mobiles is $M = \Lambda L$. The mobile sensors have the same coverage range as static sensors. Due to energy and cost considerations, we assume that each mobile sensor only moves once over a limited distance, to heal coverage holes in the network. We assume that the mobiles are provisioned with sufficient energy, so that after relocation, they can sense and communicate for at least the same duration as the static sensors. Finally, our goal is to guarantee that the entire field is k-covered, where k is determined by the network operator prior to deployment.

5.1.2 Over-Provisioning Factor

We define a new metric, which we call the *over-provisioning factor* $\eta = \frac{\lambda+\Lambda}{k}$, i.e., the ratio of sensor (static and mobile) density to the coverage requirement of the network. Clearly, the smaller the value of η, the more efficient is the network deployment in providing k-coverage.

For deterministically deployed networks, the optimal over-provisioning factor is $\Theta(1)$. The upper bound for η can be found by placing sensors on regular grids. For example, placing sensors on square grids with side length of $d_a = \sqrt{2}r$ can provide 1-coverage over the network. If k-coverage is required, k sensors can be placed at every grid point. Thus, the over-provisioning factor for this deterministic deployment is $\eta_s = \frac{k}{k(\sqrt{2}r)^2} = \frac{\pi}{2}$. The reason that η_s is larger than 1 is that there are still some overlapping areas between adjacent sensors in deterministic deployments. For higher efficiency, we can place sensors on equilateral triangular lattices to achieve $\eta_t = \frac{2\sqrt{3}\pi}{9}$, which is the most efficient regular lattice for 1-coverage [69]. It is easy to see that the over-provisioning factor is lower bounded by 1 for any deployment, since the sum of areas of sensing regions of all sensors should be k times larger than the sensing field size. Therefore, the optimal over-provisioning factor for deterministically deployed sensor networks is $\Theta(1)$.

Let us now investigate the over-provisioning factor for randomly deployed static sensor networks with density λ. By the theory of random coverage processes ([67] Theorem 3.6), the total expected area which is uncovered is $e^{-\lambda}L$. By choosing a large enough λ, the percentage of uncovered area, which is $e^{-\lambda}$, can be made arbitrarily small. However, the probability that there exists a connected coverage hole larger than unit area approaches one for a network with constant sensor density λ when the network size $L \to \infty$. The reason for this is as follows: Consider the case that a point in the network has no sensors within a distance of $2r$ from it. If such a point exists, the disk with radius $r = \frac{1}{\sqrt{\pi}}$ around it will be uncovered, which is a coverage hole with an area of at least 1. Note that such a point always exists when the network is *not* completely covered with an increased sensing range of $2r$. As shown by the theory of random coverage processes ([67] Theorem 3.1), with probability approaching one, a network cannot be completely covered by a constant density of sensors with range of $2r$ when the network sizes goes to infinity. Therefore, we see that a constant sensor density of λ can not guarantee that there are no big holes in the network as the network size grows, even though most areas of the field will be covered.

To achieve k-coverage in a large network, the static sensor density needs to grow with the network size as $\lambda = \log L + (k+2)\log\log L + c(L)$ where $c(L) \to +\infty$ as $L \to +\infty$ [6]. The over-provisioning factor for a randomly deployed static sensor network is:

$$\eta_s = \frac{\log L + (k+2)\log\log L + c(L)}{k} \tag{5.2}$$

which is $O(\log L)$ for fixed values of k. This shows that the coverage efficiency for random static sensor networks become worse as the network size increases.

5.2 Coverage in All-mobile Sensor Networks

5.2.1 Maximum movement distance

We now consider coverage in networks when all sensors are mobiles and are randomly deployed. These mobile sensors then reposition themselves so as to provide k-coverage. Clearly, in this case we should be able to achieve $\eta_m = \Theta(1)$. However, the key question is what is the maximum distance that each sensor has to move in order to place itself at the optimum location, since movement consumes a significant amount of energy [48]. Most prior research tries to minimize the total distance moved or total number of movements made by all the sensors, e.g., [27]. This is inadequate since energy is not transferable between mobile sensors. Therefore, it is better to limit the maximum moving distance for each mobile by moving several mobiles over a short distance, such as the cascaded movement in [12].

We bound the maximum moving distance for all-mobile networks as follows:

Theorem 7 *Consider an all-mobile sensor network uniformly and independently distributed over a square field with area L. The network can provide k-coverage with an*

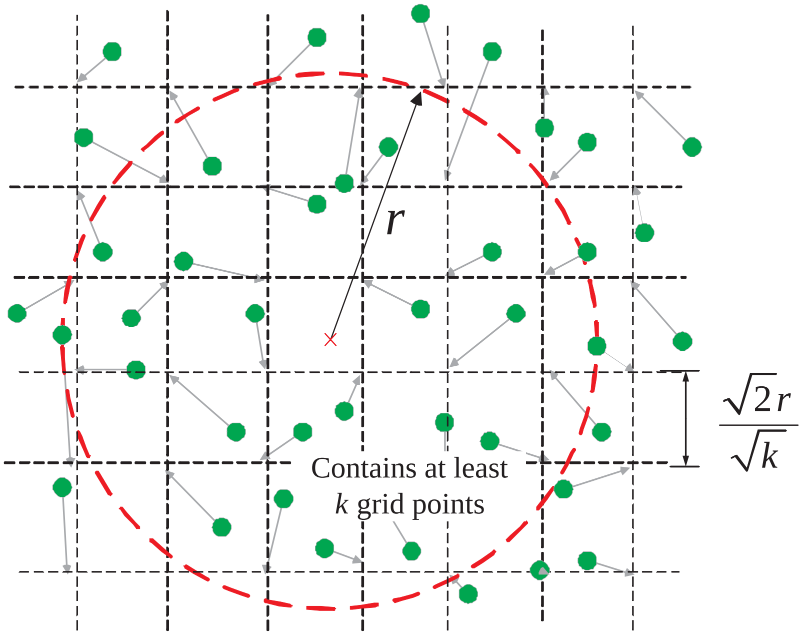


Figure 5.1: Matching mobiles to grid points in all-mobile networks.

over-provisioning factor of $\eta_m = \frac{\pi}{2}$ *and the maximum distance moved by any mobile sensor is* $O(\frac{1}{\sqrt{k}}\log^{3/4}(kL))$ *w.h.p. (with high probability).*

Proof: In order to provide a tight bound on the moving distance, we use a different placement than the deterministic placement discussed in the previous section, which places k sensors at the same grid point of side length $d_s = \sqrt{2}r$. Instead, here we divide the sensing field into square grids with side length of $d_a = \frac{\sqrt{2}r}{\sqrt{k}}$ as shown in Fig. 5.1. It is easy to see that the density of the grid points is $\frac{k}{2r^2} = \eta_m k$.

We first show that the network can be k covered with one mobile at each of these grid points. Then, we will bound the maximum moving distance for uniformly distributed mobiles to achieve such a regular grid deployment.

Assume that the mobiles have been relocated so that each grid point has exactly one mobile on it. First consider coverage on interior network areas which have distance more than r to the field boundary. If a point is within distance r to at least k grid points, it is then k covered. By the lower bounds on lattice points covered by a circle [70], there are at least $W(k)$ lattice points of side length of d_a covered by a circle of radius r centered at an arbitrary point:

$$W(k) \geq \frac{\pi(r - \frac{1}{\sqrt{2}}d_a)^2}{d_a^2} = k \times \frac{\pi}{2}(1 - \frac{1}{\sqrt{k}})^2 \tag{5.3}$$

Note that $W(k)/k$ is a monotonically increasing function when $k \geq 1$, and we have $W(k) > k$ when $k \geq 25$. It is also easy to verify that the network is at least k-covered when $1 \leq k < 25$. Thus, we can see that if there is one sensor at each grid point then the network interior is completely k-covered. To cover points near the boundary, we can slightly increase the deployment field to a $(l + 2r) \times (l + 2r)$ square. This only increases the density by a fraction of $O(\frac{r}{l})$, which is negligible when the network size is large.

After mobiles are randomly deployed in the network, we need to relocate mobiles so that each grid point has exactly one mobile. This is essentially a matching problem between mobile sensors and grid points. The maximum moving distance for mobile sensors can be derived from the results of the minimax grid matching problem studied in [50]:

Consider an $l \times l$ square region with square grids of unit side length. If we randomly and independently scatter $L = l^2$ points in the region according to a uniform distribution, then w.h.p., there exists a perfect match between the L random points and the L grid points with maximum distance between any matched pairs of $O(\log^{3/4} L)$.

Note that the total number of grid points is $\frac{k}{2r^2}L$ in our network instead of L. Therefore, the maximum moving distance will be $O(\log^{3/4}(kL))$ times the side length of the grid. Since our grid size is $d_a = \frac{\sqrt{2}r}{\sqrt{k}}$ instead of 1, we get the maximum moving distance bound of $O(\frac{1}{\sqrt{k}} \log^{3/4}(kL))$. ∎

Theorem 7 shows that it is possible to relocate the mobiles by only a small distance to achieve deterministic sensor placement. The actual relocation algorithm will be discussed in Chapter 6.

An interesting point in an all-mobile sensor network is that the mobiles compensate the randomness in large networks differently when compared to static approaches. The static approach needs to use higher density, scaled as $O(\log L)$, to compensate for the network size. In mobile sensor networks, the sensor density remains constant while mobiles need to increase their moving distance as $O(\log^{3/4} L)$ as the network size increases.

5.2.2 Numerical results

As shown in section 5.2.1, providing k-coverage in an all-mobile network is the same as the 1-coverage case, but with the moving distance divided by $\sqrt{k}$. Hence, we only consider the maximum matching distance for 1-coverage in our numerical experiments.

In the experiments, $M = \Lambda L$ mobiles are uniformly and randomly scattered into the network with area of L, where Λ is fixed as $\frac{\pi}{2}$. Then, the mobiles are matched to M grid points on grids with side length of $d_s = \sqrt{2}r$ so that they can provide full coverage over the field (see section 5.2.1). The matching is performed by a centralized

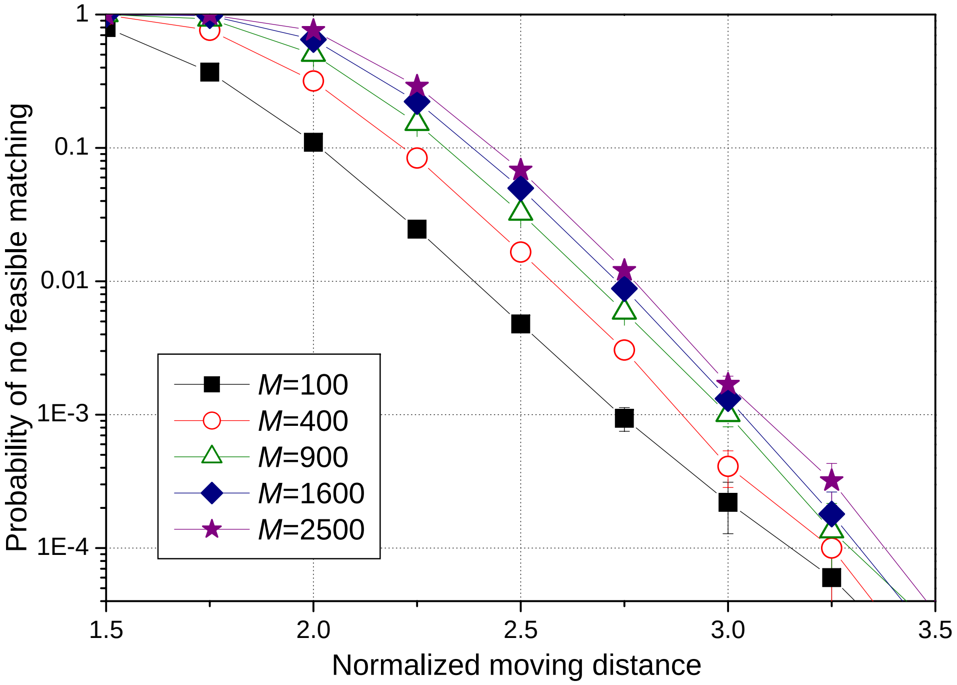


Figure 5.2: Probability that no feasible matching exists for a given moving distance in all-mobile networks (moving distance normalized by $\sqrt{2}r$, confidence interval 95%)

linear programming algorithm which will be described in Chapter 6. By repeating this over 10^5 randomly generated topologies, we find the probability that no feasible matching exists for a given maximum moving distance D.

Fig. 5.2 shows the probability that no feasible matching exists in different network sizes, where the moving distance is normalized by the grid size of d_s. From Fig. 5.2, we see that the probability that no feasible matching exists quickly drops from 1 to 0 as the moving distance increases from $1.5d_s$ to $3.5d_s$. This phenomenon is a consequence of the fact that in random geometric graphs, monotone properties demonstrate critical threshold phenomena [71].

In this experiment, the network size is changed from 10×10 grids to 50×50 grids. Consider the moving distance which can ensure the network to be completely covered by relocated mobiles with probability higher than 99.9% (where curves drop below 10^{-3} in Fig. 5.2). We see that the moving distance is only increased by about $0.4d_s$ while the network size is increased by 25 times. Also, note that the maximum moving distance for large networks is small compared to the networks size. For example, in a network with side length of $l = 50d_s$, the mobiles only need to move for at most $3.5d_s$ to form a regular grid deployment, which is less than one tenth of the network size. For larger networks, the difference will be even greater since the moving distance scales as $O(\log^{3/4} L)$. A online demo for networks smaller than 10×10 grids is available on [72]. When designing mobile networks, we can find the required moving distance D for a given network size and failure probability via Fig. 5.2. Then the mobiles can be designed to carry enough energy for moving for a distance of D so that all mobiles can successfully relocate themselves without exhausting their batteries.

5.3 Coverage in Hybrid Sensor Networks

The all-mobile network can achieve deterministic sensor deployment by moving sensors over a small distance. However, mobile sensors are much more expensive than static sensors. In order to reduce the network cost, it is preferable to use only a small number of mobiles to improve the network performance. In this section, we study the coverage of hybrid networks in which a large number of static sensors and a small fraction of mobile sensors are deployed. We provide a constructive proof to show that the over-provisioning factor is $O(1)$ and the fraction of mobile sensors required is less than $\frac{1}{\sqrt{2\pi k}}$. We further show that for this particular deployment, the maximum distance that any mobile sensor will have to move is $O(log^{3/4}L)$ *w.h.p.*

5.3.1 Mobile sensor density

In the constructive proof, we fix the static sensor density at $\lambda = 2\pi k$. The trade-off between static sensor density and mobile sensor density will be further discussed in later sections.

We divide the network into square *cells* with equal side length of $d_h = r/\sqrt{2}$. Since the sensing range is r, any sensor in the cell can completely cover the cell. The average number of static sensors in each cell will be $2\pi k d_h^2 = k$.

The network will be k-covered if all cells contain at least k sensors. However, some cells may contain fewer than k static sensors due to the randomness in deployment. If a cell i contains $n_i < k$ static sensors, we say cell i has $v_i = k - n_i$ *vacancies.* According to the Poisson approximation, n_i will be asymptotically independently and identically distributed as:

$$\mathbb{P}\{n_i = j\} = \frac{k^j e^{-k}}{j!} \tag{5.4}$$

The random variable $v_i = [k - n_i]^+$, where $[x]^+$ means $\max\{x, 0\}$, will be distributed as:

$$\mathbb{P}\{v_i = j\} = \begin{cases} \frac{k^{k-j}e^{-k}}{(k-j)!} & 1 \le j \le k \\ 1 - \sum_{m=0}^{k-1} \frac{k^m e^{-k}}{m!} & j = 0 \\ 0 & otherwise \end{cases} \tag{5.5}$$

The expected number of vacancies in a cell will be:

$$\begin{aligned} \mathbb{E}\{v_i\} &= \sum_{j=1}^{k} j \frac{k^{k-j}e^{-k}}{(k-j)!} \\ &= \sum_{l=0}^{k-1} (k-l) \frac{k^l e^{-k}}{l!} \\ &= \sum_{l=1}^{k-1} \left[\frac{k^{l+1}e^{-k}}{l!} - \frac{k^l e^{-k}}{(l-1)!} \right] + ke^{-k} \\ &= \sum_{l=1}^{k} \frac{k^l e^{-k}}{(l-1)!} - \sum_{l=1}^{k-1} \frac{k^l e^{-k}}{(l-1)!} \\ &= \frac{k^k e^{-k}}{(k-1)!} = k\mathbb{P}\{n_i = k\} \end{aligned} \tag{5.6}$$

Since v_i are independently and identically distributed random variables, we drop the subscript i in $\mathbb{E}\{v_i\}$ in later derivations. The average number of vacancies per cell will converge to $\mathbb{E}\{v\}$ when the network size is large, by the Law of Large Numbers [65]. In other words, the average number of vacancies per cell will be within a range of $[(1-\epsilon)\mathbb{E}\{v\}, (1+\epsilon)\mathbb{E}\{v\}]$ for arbitrarily small values of ϵ when $L \to \infty$. Therefore, with a mobile density of $\Lambda = \frac{(1+\epsilon)\mathbb{E}\{v\}}{(r/\sqrt{2})^2} = (1+\epsilon)2\pi\mathbb{E}\{v\}$, the number of mobiles is almost surely larger than or equal to the total number of vacancies for large networks. As ϵ can be made arbitrarily small, we just use the asymptotic mobile density of $\Lambda = 2\pi\mathbb{E}\{v\}$ in future derivations.

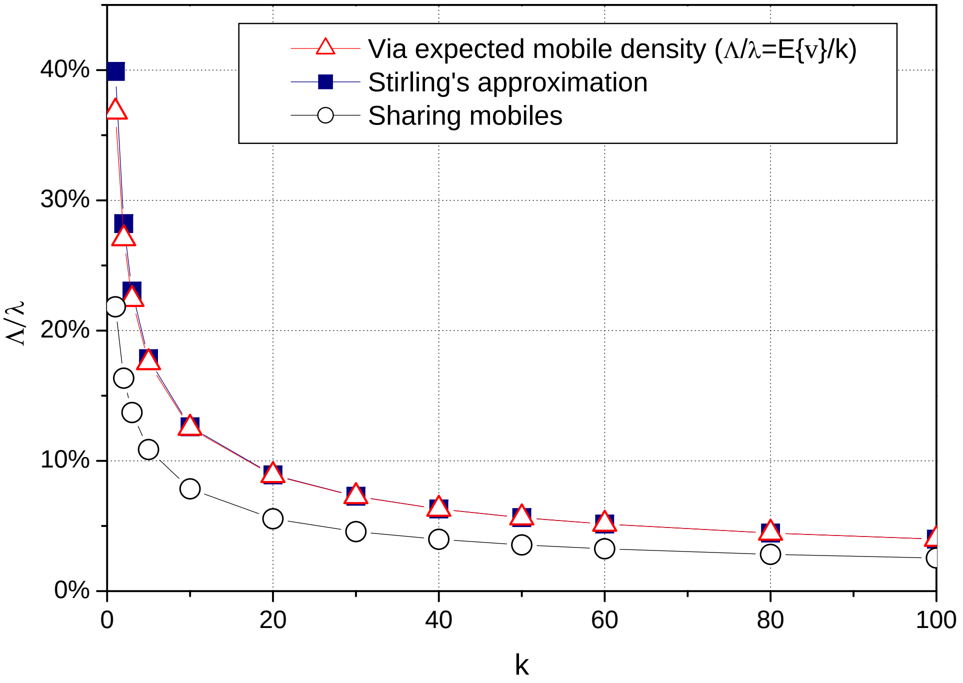


Figure 5.3: Ratio of Λ/λ (with $\lambda = 2\pi k$).

Using Stirling's approximation, $k! \approx k^k e^{-k}\sqrt{2\pi k}$:

$$\mathbb{E}\{v\} \quad = \quad k\frac{k^k e^{-k}}{k!} \approx \frac{\sqrt{k}}{\sqrt{2\pi}} \tag{5.7}$$

Note that the error of Stirling's approximation has the order of $O(e^{1/(12k)})$. Thus, we have $\mathbb{E}\{v\} \to \sqrt{k}/\sqrt{2\pi}$ as $k \to \infty$, see Fig. 5.3. Consequently, we have $\Lambda \approx \sqrt{2\pi k}$. As the static sensor density is $2\pi k$, the density ratio of mobile sensors compared to static sensors is $\frac{\Lambda}{\lambda} \approx \frac{1}{\sqrt{2\pi k}}$. As k increases, a smaller fraction of sensors need to be mobile to fill the vacancies. This agrees with the intuition that the Poisson distributed number of static sensors in a cell will be more concentrated around the mean of k as k increases.

The summation of mobile and static sensor density is still $O(k)$ since we have

$\lambda = 2\pi k$ and $\Lambda \approx \sqrt{2\pi k}$. More precisely, the over-provisioning factor $\eta_h = \frac{\lambda+\Lambda}{k} \leq 2\pi + \sqrt{2\pi}$ for an arbitrarily large network and any integer value of k. The fraction of mobiles needed for different k values are plotted in Fig. 5.3. For k larger than 15, fewer than 10% of the sensors need to be mobile. However, for small k values, the mobile sensor density Λ can be larger than the density of all-mobile networks. For example, when $k = 1$, we can use mobile density of $\frac{\pi}{2}$ to achieve a deterministic square coverage over the field while our solution requires density of $\Lambda = \sqrt{2\pi} > \frac{\pi}{2}$. This problem will be further discussed in section 5.4 where we can use several methods to reduce the mobile density.

5.3.2 Maximum movement distance

In the hybrid network solution discussed above, we need to move mobiles to fill in the vacancies in each cell. In other words, we need to build up a one-to-one matching between mobile sensors and vacancies. In the following, we show that the maximum distance that any mobile sensor will have to move is $O(\log^{3/4} L)$ with high probability.

The matching is built in two steps: First, we match the mobiles to points on a grid with side length of $\frac{1}{\sqrt{\Lambda}}$. The maximum matching distance is $O(\frac{1}{\sqrt{\Lambda}} \log^{3/4}(\Lambda L))$ *w.h.p.*, as shown in section 5.2.1. The function $\frac{1}{\sqrt{\Lambda}} \log^{3/4}(\Lambda L)$ is a decreasing function with Λ when Λ and L are larger than 1. Therefore, the matching distance decreases with Λ and it can be rewritten as $O(\log^{3/4} L)$, since $\Lambda \approx \sqrt{2\pi k} > 1$.

The second step is to match vacancies to the grid points on the grid with side length of $\frac{1}{\sqrt{\Lambda}}$. Unfortunately, the results from [50] cannot be directly applied since the vacancies are not uniformly distributed. We use the following theorem to bound the maximum matching distance:

Theorem 8 *Consider a square network with area L where ΛL vacancies are distributed independently and identically, in cells with side length of $d_h = \frac{r}{\sqrt{2}}$ according to Eq.(5.5). Then, w.h.p., there exists a matching which has maximum matching distance of $O(\log^{3/4} L)$ between vacancies and the grid points on grids with side length of $\frac{1}{\sqrt{\Lambda}}$.*

The proof of Theorem 8 is similar to the proof given by Leighton and Shor for uniformly distributed points [50]. The major differences are that our vacancy distribution is not uniform and the vacancies are distributed in discrete cells rather than in a continuous field. In this proof, we focus on the differences and only give an outline for parts which are same as in [50].

Consider the case of matching ΛL randomly distributed vacancies to the same number of grid points on the grid with side length of $\frac{1}{\sqrt{\Lambda}}$ by maximum matching distance of D. Define the neighborhood of a region $\mathcal{R}$ in the sensing field as $\mathcal{N}(\mathcal{R})$, the set of cells where all points are within a distance of D of at least one point in $\mathcal{R}$. Note that $\mathcal{R}$ is also contained in $\mathcal{N}(\mathcal{R})$. It is easy to see that mobiles on grid points in $\mathcal{N}(\mathcal{R})$ can move into $\mathcal{R}$ by moving by at most a distance of D. By Hall's Theorem [73], there exists a perfect matching with maximum moving distance D between the vacancies and grid points *if and only if* for *every* sub-region $\mathcal{R}$ in the field, the number of vacancies contained in $\mathcal{R}$, denoted as $V_{\mathcal{R}}$, is smaller or equal to the number of grid points in $\mathcal{N}(\mathcal{R})$, denoted as $W_{\mathcal{N}(\mathcal{R})}$.

For our problem, we need to first define the number of vacancies in an arbitrary region which may not contain exactly an integer number of cells with side length $d_h = \frac{r}{\sqrt{2}}$. Denote the number of cells in region $\mathcal{R}$ as $A_{\mathcal{R}} = Area(\mathcal{R})/d_h^2 = 2\pi Area(\mathcal{R})$. Suppose region $\mathcal{R}$ intersects with $C_{\mathcal{R}}$ cells, and it overlaps with only a fraction a_i of the total area of cell i, where $0 < a_i \leq 1$. Then we have $A_{\mathcal{R}} = \sum_{i=1}^{C_{\mathcal{R}}} a_i$. Define the number of vacancies in $\mathcal{R}$ as $V_{\mathcal{R}} = \sum_{i=1}^{C_{\mathcal{R}}} a_i v_i$. In other words, if a region only covers part of the cell i, the number of vacancies contributed by the cell i will be $a_i v_i$. This definition "spreads" the vacancy in cell i uniformly on the area of a cell, thus it has the property that the number of vacancies in a union of disjoint regions will be the sum of vacancies in individual regions.

Since we need to prove that $D = O(\log^{3/4} L)$, it is sufficient to only consider region $\mathcal{R}_\Gamma$ and $\mathcal{N}(\mathcal{R}_\Gamma)$ with boundaries lying along the edges of squares Γ with side length $c \log^{3/4} L$, where c is some constant [50], as shown in Fig. 5.4. We have:

$$\begin{aligned} W_{\mathcal{N}(\mathcal{R}_\Gamma)} &= \Lambda(Area(\mathcal{R}_\Gamma) + Area(\mathcal{N}(\mathcal{R}_\Gamma) \setminus \mathcal{R}_\Gamma)) \\ &= \mathbb{E}\{V_{\mathcal{R}_\Gamma}\} + \Lambda \times Area(\mathcal{N}(\mathcal{R}_\Gamma) \setminus \mathcal{R}_\Gamma) \end{aligned} \tag{5.8}$$

where $Area(\mathcal{R})$ is the area of region $\mathcal{R}$. Define the *discrepancy* $\Delta(\mathcal{R}) = |V_{\mathcal{R}} - \mathbb{E}\{V_{\mathcal{R}}\}|$. As shown in [50], we can make $Area(\mathcal{N}(\mathcal{R}_\Gamma) \setminus \mathcal{R}_\Gamma) \geq c_p Per(\mathcal{R}_\Gamma) \log^{3/4} L$ for an arbitrarily constant c_p by setting $D = O(\log^{3/4} L)$, where $Per(\mathcal{R})$ is the perimeter of region $\mathcal{R}$[1]. Thus, if we have $\Delta(\mathcal{R}_\Gamma) \leq c_p \Lambda Per(\mathcal{R}_\Gamma) \log^{3/4} L$ for all $\mathcal{R}_\Gamma$, we can guarantee that $V_{\mathcal{R}_\Gamma} \leq W_{\mathcal{N}(\mathcal{R}_\Gamma)}$ also holds for all $\mathcal{R}_\Gamma$.

[1] More strictly, this holds for at least one of $\mathcal{R}_\Gamma$ or its complement $\overline{\mathcal{R}_\Gamma}$, so we need to show that $|V_{\mathcal{R}} - \mathbb{E}\{V_{\mathcal{R}_\Gamma}\}| \leq c_p \Lambda Per(\mathcal{R}_\Gamma) \log^{3/4} L$ instead of $V_{\mathcal{R}} - \mathbb{E}\{V_{\mathcal{R}_\Gamma}\} \leq c_p \Lambda Per(\mathcal{R}_\Gamma) \log^{3/4} L$. For details, see [50].

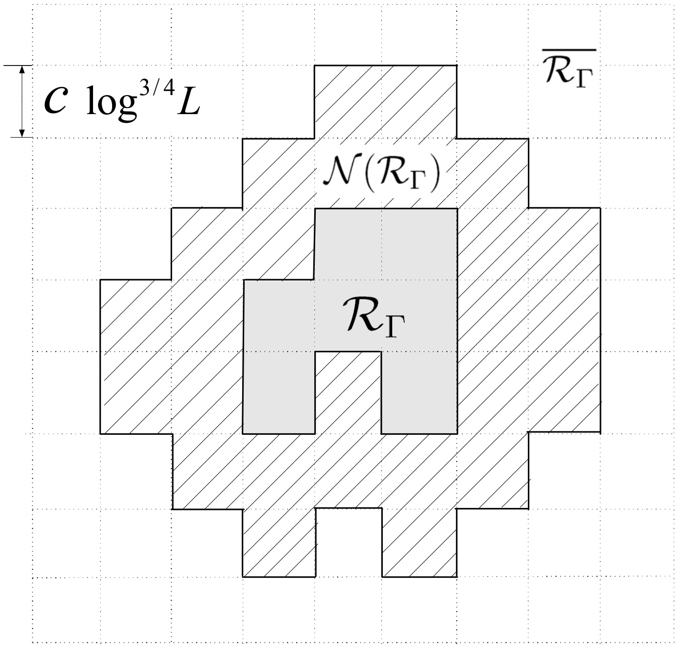

$\mathcal{C}\ \log^{3/4} L$
$\overline{\mathcal{R}_\Gamma}$
$\mathcal{N}(\mathcal{R}_\Gamma)$
$\mathcal{R}_\Gamma$

Figure 5.4: A region and its neighborhood.

Using the region decomposition method, it is shown in [50] that if for every region $\mathcal{R}$ (not necessarily having boundaries lying on Γ):

$$\mathbb{P}\{\Delta(\mathcal{R}) \geq \delta\} < O(e^{-\frac{c_2\delta^2}{Area(\mathcal{R})}})\ \ for\,\delta < Area(\mathcal{R}) \tag{5.9}$$

$$\mathbb{P}\{\Delta(\mathcal{R}) \geq \delta\} < O(e^{-c_2\delta})\ \ for\,\delta > Area(\mathcal{R}) \tag{5.10}$$

then *w.h.p.*, we will have $\Delta(\mathcal{R}_\Gamma) \leq c_p Per(\mathcal{R}_\Gamma)\log^{3/4} L$ for all regions $\mathcal{R}_\Gamma$ which have boundary lying on Γ.

Therefore, we need the following lemma:

Lemma 1 *For any region $\mathcal{R}$, we have the probability*

$$\mathbb{P}\{\Delta(\mathcal{R}) \geq \delta\} \ \ < \ \ 2e^{-\frac{\delta^2}{4\pi Area(\mathcal{R})k}} \tag{5.11}$$

when the vacancies are distributed according to Eq.(5.5).

Proof: Construct random variables $v'_i = v_i - \mathbb{E}\{v\}$ which have mean $\mathbb{E}\{v'_i\} = 0$. Accordingly, we have $V'_{\mathcal{R}} = \sum_{i=1}^{C_{\mathcal{R}}} a_i v'_i = V_{\mathcal{R}} - \mathbb{E}\{V_{\mathcal{R}}\}$ and $\Delta(\mathcal{R}) = |V'_{\mathcal{R}}|$. Since the number of vacancies in a cell can not exceed k, we have $0 \le V_{\mathcal{R}} \le A_{\mathcal{R}} k$. Thus, we only need to consider $\delta < A_{\mathcal{R}} k$, since $\mathbb{P}\{\Delta(\mathcal{R}) \ge \delta\} = 0$ when $\delta \ge A_{\mathcal{R}} k$.

We will bound the probability $\mathbb{P}\{\Delta(\mathcal{R}) \ge \delta\}$ by the Chernoff bound. For $t > 0$, we have:

$$\mathbb{P}\{V'_{\mathcal{R}} \ge \delta\} \quad \le \quad e^{-t\delta} \mathbb{E}\{e^{tV_{\mathcal{R}'}}\} \tag{5.12}$$

Since v'_i are independently distributed, we have:

$$\mathbb{E}\{e^{tV'_{\mathcal{R}}}\} = \prod_{i=1}^{C_{\mathcal{R}}} \mathbb{E}\{e^{ta_i v'_i}\} \tag{5.13}$$

Since $0 < a_i \le 1$, the function $f(x) = x^{a_i}$ is concave when $x > 0$. By Jensen's inequality, we have $\mathbb{E}\{f(x)\} \le f(\mathbb{E}\{x\})$ for a concave function $f(x)$, we get:

$$\mathbb{E}\{e^{ta_i v'_i}\} = \mathbb{E}\{(e^{tv'_i})^{a_i}\} \le (\mathbb{E}\{(e^{tv'_i})\})^{a_i} \tag{5.14}$$

for all i. Since v'_i are identically distributed, we have:

$$\mathbb{P}\{V'_{\mathcal{R}} \ge \delta\} \le e^{-t\delta} \prod_{i=1}^{C_{\mathcal{R}}} (\mathbb{E}\{e^{tv'}\})^{a_i} = e^{-t\delta} \left(\mathbb{E}\{e^{tv'}\}\right)^{A_{\mathcal{R}}} \tag{5.15}$$

Using the vacancy distribution function of Eq.(5.5), we have:

$$\begin{aligned}
\mathbb{E}\{e^{tv'}\} \quad &= \quad \sum_{j=1}^{k} e^{t(j-\mathbb{E}\{v\})} \frac{k^{k-j} e^{-k}}{(k-j)!} + e^{-t\mathbb{E}\{v\}} \left(1 - \sum_{j=0}^{k-1} \frac{k^j e^{-k}}{j!}\right) \\
&< \quad \sum_{l=0}^{k-1} e^{t(k-l-\mathbb{E}\{v\})} \frac{k^l e^{-k}}{l!} + e^{-t\mathbb{E}\{v\}} \\
&= \quad e^{-t\mathbb{E}\{v\}} (e^{tk} \sum_{l=0}^{k-1} e^{-tl} \frac{k^l e^{-k}}{l!} + 1)
\end{aligned}$$

$$
\begin{aligned}
&< \quad e^{-t\mathbb{E}\{v\}}(e^{tk-k}\sum_{l=0}^{\infty}\frac{(ke^{-t})^l}{l!}+1) \\
&= \quad e^{-t\mathbb{E}\{v\}}\left(e^{k(t+e^{-t}-1)}+1\right) \qquad (5.16)
\end{aligned}
$$

where the last equality comes from the expansion of $e^x = \sum_{l=0}^{\infty}\frac{x^l}{l!}$.

Consider two subcases:

A. $e^{k(t+e^{-t}-1)} \geq (e^{t\mathbb{E}\{v\}}-1)^{-1}$

The condition is equivalent to:

$$
e^{k(t+e^{-t}-1)} \quad \geq \quad \frac{e^{-t\mathbb{E}\{v\}}}{1-e^{-t\mathbb{E}\{v\}}} \qquad (5.17)
$$

Since $1-e^{-t\mathbb{E}\{v\}} > 0$, Eq.(5.17) can be converted to:

$$
e^{k(t+e^{-t}-1)} \quad \geq \quad e^{-t\mathbb{E}\{v\}}\left(e^{k(t+e^{-t}-1)}+1\right) > \mathbb{E}\{e^{tv'}\}
$$

By Eq.(5.15), we get:

$$
\mathbb{P}\{V'_{\mathcal{R}} \geq \delta\} \quad < \quad e^{kA_{\mathcal{R}}(t+e^{-t}-1)-t\delta}
$$

Since $\delta < A_{\mathcal{R}}k$, let $t = \log\frac{A_{\mathcal{R}}k}{A_{\mathcal{R}}k-\delta} > 0$, we get:

$$
\mathbb{P}\{V'_{\mathcal{R}} \geq \delta\} < \exp\left((A_{\mathcal{R}}k-\delta)\log\frac{A_{\mathcal{R}}k}{A_{\mathcal{R}}k-\delta}-\delta\right) \qquad (5.18)
$$

Using the inequality of $\log x \geq \frac{x^2-1}{2x}$ for $0 < x < 1$, we get:

$$
\begin{aligned}
(A_{\mathcal{R}}k-\delta)\log\frac{A_{\mathcal{R}}k}{A_{\mathcal{R}}k-\delta}-\delta &= -A_{\mathcal{R}}k(1-\frac{\delta}{A_{\mathcal{R}}k})\log(1-\frac{\delta}{A_{\mathcal{R}}k})-\delta \\
&\leq -A_{\mathcal{R}}k\left(\frac{(1-\frac{\delta}{A_{\mathcal{R}}k})^2-1}{2}+\frac{\delta}{A_{\mathcal{R}}k}\right) \\
&= -\frac{\delta^2}{2A_{\mathcal{R}}k} \qquad (5.19)
\end{aligned}
$$

So, we get $\mathbb{P}\{V'_{\mathcal{R}} \geq \delta\} < e^{-\frac{\delta^2}{2A_{\mathcal{R}}k}}$.

B. $e^{k(t+e^{-t}-1)} < (e^{t\mathbb{E}\{v\}}-1)^{-1}$

In this case we have:

$$\mathbb{E}\{e^{tv'}\} < e^{-t\mathbb{E}\{v\}}\left(\frac{1}{e^{t\mathbb{E}\{v\}}-1}+1\right) = \frac{1}{e^{t\mathbb{E}\{v\}}-1}$$

By Eq.(5.15), we get:

$$\mathbb{P}\{V'_{\mathcal{R}} \geq \delta\} \leq \left(\frac{1}{e^{t\mathbb{E}\{v\}}-1}\right)^{A_{\mathcal{R}}} e^{-t\delta} \tag{5.20}$$

As $\mathbb{E}\{v\} \geq e^{-1}$ for any k (by Eq.(5.7)), we can select $t = 3$, so that $e^{t\mathbb{E}\{v\}} - 1 > e - 1 > 1$. We have:

$$\begin{aligned}\mathbb{P}\{V'_{\mathcal{R}} \geq \delta\} &< \left(\frac{1}{e^{3\mathbb{E}\{v\}}-1}\right)^{A_{\mathcal{R}}} e^{-3\delta} \\ &< e^{-3\delta} \leq e^{-3\frac{\delta^2}{A_{\mathcal{R}}k}} < e^{-\frac{\delta^2}{2A_{\mathcal{R}}k}}\end{aligned}$$

due to $0 < \frac{\delta}{A_{\mathcal{R}}k} < 1$.

Consider both cases, we have $\mathbb{P}\{V'_{\mathcal{R}} \geq \delta\} < e^{-\frac{\delta^2}{2A_{\mathcal{R}}k}}$ when $\delta < A_{\mathcal{R}}k$. For $\delta \geq A_{\mathcal{R}}k$, it is easy to see that $\mathbb{P}\{V'_{\mathcal{R}} > \delta\} = 0$.

Now we consider the other side of the distribution, we need to bound:

$$\mathbb{P}\{V'_{\mathcal{R}} \leq -\delta\} = \mathbb{P}\{-V'_{\mathcal{R}} \geq \delta\} \tag{5.21}$$

With the Chernoff bound, we have:

$$\begin{aligned}\mathbb{P}\{-V'_{\mathcal{R}} \geq \delta\} &\leq e^{-t\delta}\mathbb{E}\{e^{-tV'_{\mathcal{R}}}\} \\ &= e^{-t\delta}\prod_{i=1}^{A_{\mathcal{R}}}\mathbb{E}\{e^{-tv'_i}\} \\ &= e^{-t\delta}\left(\mathbb{E}\{e^{-tv'}\}\right)^{A_{\mathcal{R}}} \end{aligned}\tag{5.22}$$

for $t > 0$.

Since $v' = v - \mathbb{E}\{v\}$, we have:

$$\begin{aligned} \mathbb{E}\{e^{-tv'}\} &= \mathbb{E}\{e^{-t(v-\mathbb{E}\{v\})}\} \\ &= e^{t\mathbb{E}\{v\}}\mathbb{E}\{e^{-tv}\} \qquad (5.23) \end{aligned}$$

Define:

$$f_1(t) = \mathbb{E}\{e^{-tv}\} \qquad (5.24)$$

$$f_2(t) = e^{\frac{kt^2}{2}-t\mathbb{E}\{v\}} \qquad (5.25)$$

Consider the derivative:

$$f_1'(t) = \mathbb{E}\{-ve^{-tv}\} \qquad (5.26)$$

Since $v \geq 0$, we have $f_1'(t) < 0$ for all t.

Note that $e^x \geq 1 + x$ for all x. We get $-ve^{-tv} \leq -v(1-tv)$ for all t and v, therefore:

$$\begin{aligned} f_1'(t) &\leq \mathbb{E}\{-v(1-tv)\} \\ &= t\mathbb{E}\{v^2\} - \mathbb{E}\{v\} \qquad (5.27) \end{aligned}$$

We have:

$$\begin{aligned} \mathbb{E}\{v^2\} &= \sum_{j=1}^{k} j^2 \frac{k^{k-j}e^{-k}}{(k-j)!} \\ &= \sum_{l=0}^{k-1} (k-l)^2 \frac{k^l e^{-k}}{l!} \\ &= k\sum_{l=0}^{k-1}(k-l)\frac{k^l e^{-k}}{l!} - \sum_{l=0}^{k-1} l(k-l)\frac{k^l e^{-k}}{l!} \\ &= k\sum_{l=0}^{k-1}(k-l)\frac{k^l e^{-k}}{l!} - k\sum_{l=1}^{k-1}(k-l)\frac{k^{l-1}e^{-k}}{(l-1)!} \end{aligned}$$

$$\begin{aligned}
&= k\left(\sum_{l=0}^{k-1}(k-l)\frac{k^l e^{-k}}{l!} - \sum_{l=1}^{k-1}(k-(l-1))\frac{k^{l-1}e^{-k}}{(l-1)!} + \sum_{l=1}^{k-1}\frac{k^{l-1}e^{-k}}{(l-1)!}\right) \\
&= k\left(\sum_{l=0}^{k-1}(k-l)\frac{k^l e^{-k}}{l!} - \sum_{l=0}^{k-2}(k-l)\frac{k^l e^{-k}}{l!} + \sum_{l=0}^{k-2}\frac{k^l e^{-k}}{l!}\right) \\
&= k\sum_{l=0}^{k-1}\frac{k^l e^{-k}}{l!} < k \qquad (5.28)
\end{aligned}$$

This gives:

$$f_1'(t) \;<\; kt - \mathbb{E}\{v\} \qquad (5.29)$$

for all t and k.

Now consider $f_2'(t)$, we have:

$$f_2'(t) = (kt - \mathbb{E}\{v\})e^{\frac{kt^2}{2} - t\mathbb{E}\{v\}} \qquad (5.30)$$

We have $f_2'(t) > 0$ when $t > \mathbb{E}\{v\}/k$. Since $f_1'(t) \leq 0$, we have $f_2'(t) \geq f_1'(t)$ when $t > \mathbb{E}\{v\}/k$.

When $0 < t \leq \mathbb{E}\{v\}/k$, we get:

$$\begin{aligned}
e^{\frac{kt^2}{2} - t\mathbb{E}\{v\}} &< 1 \qquad (5.31) \\
(kt - \mathbb{E}\{v\})e^{\frac{kt^2}{2} - t\mathbb{E}\{v\}} &\geq kt - \mathbb{E}\{v\} \qquad (5.32)
\end{aligned}$$

since $kt - \mathbb{E}\{v\} \leq 0$. Combining Eq.(5.29) and (5.32), we have $f_2'(t) \geq f_1'(t)$ when $0 < t \leq \mathbb{E}\{v\}/k$.

Since $f_1(0) = f_2(0) = 1$ and $f_1'(t) \leq f_2'(t)$ for all $t \geq 0$, we have

$$\begin{aligned}
f_1(t) &= \int_0^t f_1'(t)dt + f_1(0) \\
&\leq \int_0^t f_2'(t)dt + f_2(0) \\
&= f_2(t) \qquad (5.33)
\end{aligned}$$

for $t > 0$.

Therefore, we have:

$$\mathbb{E}\{e^{-tv}\} \quad \leq \quad e^{\frac{kt^2}{2} - t\mathbb{E}\{v\}} \tag{5.34}$$

and

$$\mathbb{E}\{e^{-tv'}\} \quad = \quad e^{t\mathbb{E}\{v\}}\mathbb{E}\{e^{-tv}\} \leq e^{\frac{kt^2}{2}} \tag{5.35}$$

for $t > 0$.

Use Eq.(5.22), we get:

$$\mathbb{P}\{-V'_{\mathcal{R}} \geq \delta\} \quad \leq \quad e^{-t\delta} e^{A_{\mathcal{R}} \times \frac{kt^2}{2}} \tag{5.36}$$

Select $t = \frac{\delta}{A_{\mathcal{R}} k} > 0$, we have:

$$\mathbb{P}\{-V'_{\mathcal{R}} \geq \delta\} \quad \leq \quad e^{-\frac{\delta^2}{2A_{\mathcal{R}} k}} \tag{5.37}$$

Combining the results for both sides, we have $\mathbb{P}\{|V'_{\mathcal{R}}| \geq \delta\} < 2e^{-\frac{\delta^2}{2A_{\mathcal{R}} k}}$. This is equivalent to:

$$\mathbb{P}\{\Delta(\mathcal{R}) \geq \delta\} \quad < \quad 2e^{-\frac{\delta^2}{4\pi Area(\mathcal{R}) k}} \tag{5.38}$$

as $A_{\mathcal{R}} = 2\pi Area(\mathcal{R})$. ∎

Note that the discrepancy bound derived above is $c_p \Lambda Per(\mathcal{R}_\Gamma) \log^{3/4} L$ instead of $c_p Per(\mathcal{R}_\Gamma) \log^{3/4} L$. Thus, in our problem the discrepancy should be scaled by a factor of Λ, which is the density of the grid points. We have $\Lambda \approx \sqrt{2\pi k}$. Put $\delta' = \Lambda\delta$ into Eq.(5.38), we can see that the factor of k in the exponent in Eq.(5.38) will be canceled by Λ^2. In this case, it is easy to see that the bound in Lemma 1 satisfies Eq.(5.9) and

Table 5.1: Sensor density and moving distance tradeoff for k-coverage

Network type	*Static sensor density*	*Mobile density*	*Maximum moving distance*
Static	$O(k \log\log L + \log L)$	0	0
All-mobile	0	$O(k)$	$O(\frac{1}{\sqrt{k}} \log^{3/4}(kL))$
Hybrid	$O(k)$	$O(\sqrt{k})$	$O(\log^{3/4} L)$

(5.10) for both cases of $\delta \leq Area(\mathcal{R})$ and $\delta > Area(\mathcal{R})$. Therefore, using the same region decomposition method as in [50], we have $\Delta(\mathcal{R}_\Gamma) \leq c_1 \Lambda Per(\mathcal{R}_\Gamma) \log^{3/4} L$ with high probability. This directly leads to the upper bound of $O(\log^{3/4} L)$ in maximum moving distance.

Thus, the matching distance between vacancies and the grid points is also $O(\log^{3/4} L)$. Since the big O notation hides constant factors, the distance between the mobile and vacancy matched to the same grid point is also $O(\log^{3/4} L)$. This builds up the one-to-one matching between mobiles and vacancies with maximum distance between matched pairs as $O(\log^{3/4} L)$.

Compared to the all-mobile case, the moving distance of the hybrid network is $O(\sqrt{k})$ times larger, yet it is small compared to the network size.

In summary, Table 5.1 shows the sensor density and moving distance for different network structures.

5.3.3 Numerical results

In the numerical experiments of hybrid networks, we divide the network area as cells with side length of $d_h = r/\sqrt{2}$ as in section 5.3.1. We uniformly deploy $N = \lambda L$ static

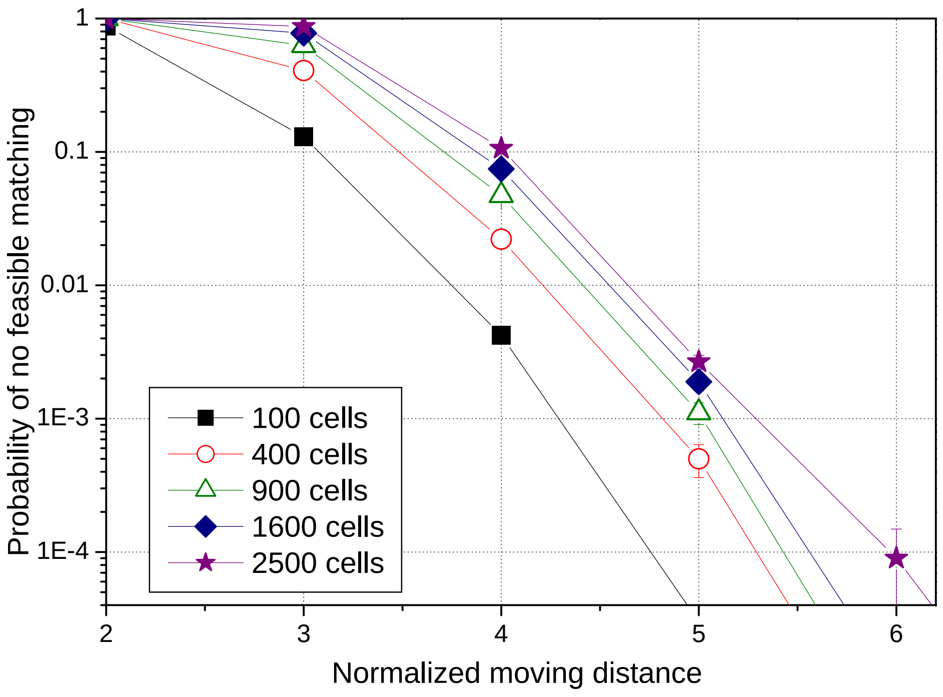


Figure 5.5: Probability that no feasible matching exists for a given moving distance in hybrid networks of different size (k=10, moving distance normalized by $r/\sqrt{2}$, confidence interval 95%)

sensors and $M = \Lambda L$ mobiles in the network, where $\lambda = 2\pi k$ and M is selected so that there are exactly enough mobiles to fill all vacancies. The mobiles in one cell can move to cells within a distance of D. The following results are obtained by solving the linear program which will be described in Chapter 6, on 10^5 randomly generated topologies for each network size.

Fig. 5.5 shows the probability that there is no feasible mobility schedule to fill all vacancies under different network sizes when k is 10. In the hybrid network, the maximum moving distance for mobiles also increases slowly as the network size

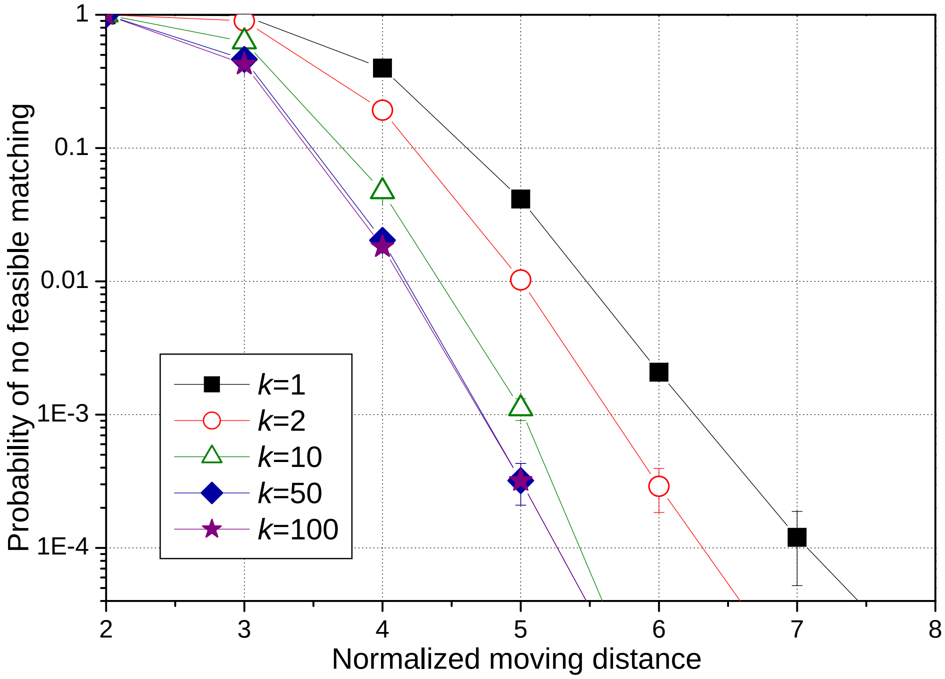


Figure 5.6: Probability that no feasible matching exists for a given moving distance in hybrid networks of different k (900 cells, moving distance normalized by $r/\sqrt{2}$, confidence interval 95%)

increases. Note that the moving distance in hybrid networks is normalized by d_h instead of d_s in the previous section. Since we have $d_h = 0.5d_s$, the actual moving distance for hybrid networks is comparable to the all-mobile case when $k = 1$. The moving distance for networks with varying k is plotted in Fig. 5.6 with network size of 900 cells. We see that the maximum moving distance required for hybrid networks slightly decreases as k increases, while the curve for $k = 50$ and $k = 100$ almost overlap. This shows that when k is small the maximum moving distance is affected by both the matching distance from the mobile to the grid points and matching distance from the grid points to the vacancies. As k increases, the matching distance from the mobile to the grid points will decrease to zero as the mobile density increases, see section 5.3.1. Then, the moving distance is dominated by the matching distance from the vacancies to the grid points which is not changed as k increases.

5.4 Discussions

5.4.1 Network structure alternatives

The mobile density used in section 5.3.1 is quite high especially when k is small. In this section, we provide several methods to reduce mobile density.

5.4.1.1 Sharing Mobiles

The cell used in our hybrid network has side length of $r/\sqrt{2}$, which is quite conservative compared to the sensing range of r. Actually, a mobile at the corner of a cell can cover four cells at the same time, see Fig. 5.7.

Consider the super-cell which contains 9 cells as shown in Fig. 5.7. We need to deploy the mobiles only at the four central points of p_i to provide coverage for the 9 cells. For example, when $k = 1$, if there are no static sensors in the super-cell, we can put one sensor at each p_i to provide full coverage on the 9 cells while the basic algorithm discussed in section 5.3.1 uses 9 mobiles. If cell 1 has at least one static sensor, then at most three mobiles are needed to stay at p_2, p_3 and p_4 to cover the rest of cells. As each mobile can cover four adjacent cells, the mobile density can be reduced by a constant factor.

The density of mobiles required for the sharing mobile scheme can be numerically calculated by enumerating possible vacancy distributions in the super-cell. The reduced mobile density is shown in Fig. 5.3 in section 5.3.1. We see that when k is small, this scheme can reduce the mobile density by half, compared to the original hybrid structure. In this case, our hybrid network can use fewer mobiles than all-mobile networks when $k = 1$. However, the improvement ratio reduces as k increases as shown in Fig. 5.3. When implementing the sharing mobile algorithm, we separate the network region to disjoint super-cells. The number of mobiles required in each super-cell can be determined when the number of static sensors in the 9 small cells is known. We can apply the push-relabel algorithm to find the optimal solution in this case also.

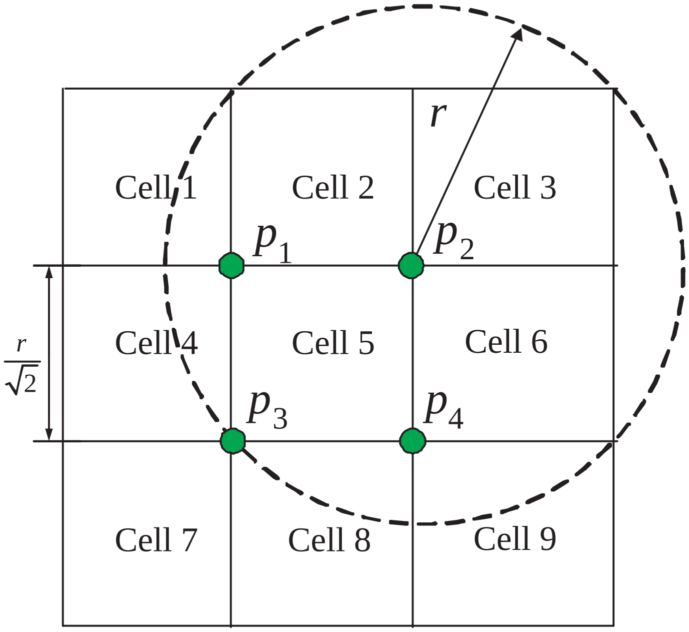


Figure 5.7: Sharing the mobile sensor.

5.4.1.2 Increasing Static Sensor Density

We can also reduce the density of mobiles by increasing the density of static sensors. Suppose that we increase the density of static sensors so that the average number of sensors in each cell of side length $\frac{r}{\sqrt{2}}$ is $g \geq k$.

The number of vacancies will be distributed as:

$$\mathbb{P}\{\hat{v}_i = j\} = \begin{cases} \frac{g^{k-j}e^{-g}}{(k-j)!} & 1 \leq j \leq k \\ 1 - \sum_{m=0}^{k-1} \frac{g^m e^{-g}}{m!} & j = 0 \\ 0 & otherwise \end{cases} \tag{5.39}$$

Similar to Eq.(5.6), the expected number of vacancies in a cell will be:

$$\mathbb{E}\{\hat{v}\} \quad = \quad \sum_{l=0}^{k-1}(k-l)\frac{g^l e^{-g}}{l!}$$

$$
\begin{aligned}
&\leq \left(\frac{g}{k}\right)^{k-1} e^{-(g-k)} \sum_{l=0}^{k-1} (k-l) \frac{k^l e^{-k}}{l!} \\
&\leq e^{\frac{(k-1)(g-k)}{k}} e^{-(g-k)} \mathbb{E}\{v\} = e^{-\frac{g-k}{k}} \mathbb{E}\{v\} \qquad (5.40)
\end{aligned}
$$

The third step uses the inequality of $(1 + \frac{x}{n})^n \leq e^x$ when $x > 0$ and $n > 0$, to get $\left(\frac{g}{k}\right)^{k-1} = (1 + \frac{g-k}{k})^{k-1} \leq e^{\frac{(k-1)(g-k)}{k}}$. Therefore, with a increased static sensor density, the density of mobile sensors is reduced at least exponentially as $e^{-\frac{g-k}{k}}$. Specifically, when the density of static sensor is doubled ($g = 2k$), the density of mobiles can be reduced by at least e^{-1} which is close to one third. Note that the bound in Eq.(5.40) is tight only for small k, we can use even smaller number of mobiles when k is large. Note that we are still maintaining a constant over-provisioning factor in this solution.

5.4.1.3 Maximum Moving Distance

Suppose that we use one of the two methods discussed above and the mobile density is Λ. Using the same argument as in section 5.3.1, we can first match the mobiles to grid points with side length $\frac{1}{\sqrt{\Lambda}}$, then match the vacancies to the grid points. However, the vacancy distributions are different from those in section 5.3.1, and we have a different bound on moving distance here.

Theorem 9 *Consider a square network with area L with ΛL vacancies distributed independently and identically in each square cell with side length of d, where d is some constant. If the number of vacancies in each cell is upper bounded by k, then, w.h.p. we can find a matching which has maximum matching distance as $O(\frac{k}{\Lambda} \log^{3/4} L)$ between vacancies and the grid points (on grids with side length of $\frac{1}{\sqrt{\Lambda}}$).*

Proof: The proof of Theorem 9 is similar to the proof of Theorem 8. Consider a region $\mathcal{R}$ with exactly $A_{\mathcal{R}}$ cells which has side length of d. Since the number of vacancies in each cell is bounded in $[0, k]$, by Hoeffding's inequality [74], we have:

$$\mathbb{P}\{|V_{\mathcal{R}} - \mathbb{E}\{V_{\mathcal{R}}\}| \geq \delta\} < e^{-\frac{2\delta^2}{A_{\mathcal{R}}k^2}} \tag{5.41}$$

With similar arguments as the proof in Theorem 8, this bound can be extended to regions which do not contain exactly an integer number of cells, and we get:

$$\mathbb{P}\{|V_{\mathcal{R}} - \mathbb{E}\{V_{\mathcal{R}}\}| \geq \delta\} < e^{-\frac{2d^2\delta^2}{Area(\mathcal{R})k^2}} \tag{5.42}$$

Comparing Eq.(5.42) and Eq.(5.38), we see that any region will have discrepancy less than $c_p k Per(\mathcal{R}_\Gamma) \log^{3/4} L$ *w.h.p.* This gives an upper bound on moving distance of $O(\frac{k}{\Lambda} \log^{3/4} L)$ since the grid point density is Λ. Note that the bound of Theorem 9 is $\sqrt{k}$ times larger than the bound in Theorem 8 when we set $\Lambda = O(\sqrt{k})$. This is because of the Chernoff bound we derived in Theorem 8 for the specific vacancy distribution is tighter than the Hoeffding bound used here. ∎

Theorem 9 can be directly used for the increasing static sensor density scheme. For the first scheme which shares the mobiles, we can set d as the side length of the super-cell, which is $\frac{3r}{\sqrt{2}}$. The number of mobiles required in each super cell is independently and identically distributed. Also, each super cell needs at most $4k$ mobiles. Applying Theorem 9, we get the maximum moving distance of $O(\frac{k}{\Lambda} \log^{3/4} L)$. Therefore, when Λ and k are fixed, the maximum moving distances for both schemes still increases as $O(\log^{3/4} L)$ with the network size L. From Theorem 9, we also see that when the mobile density decreases, the moving distance for mobiles will increase.

5.4.2 Network Lifetime

The k-coverage problem is closely related to the network lifetime problem. We define the network lifetime as time that the network cannot provide 1-coverage over the sensing field. Suppose that each sensor can monitor the region for time of τ before the battery is exhausted. Then k-coverage is a necessary condition for network to reach $k\tau$ lifetime [6], since any point which is covered by fewer than k sensors can not be monitored for time longer than $k\tau$. On the other hand, ensuring the network to be k-covered does not directly lead to $k\tau$ lifetime since there may not exist a sleep-wake schedule which can operate for more than $k\tau$ lifetime [75].

In our hybrid network scheme, there are at least k mobile or static sensors in each cell with side length $\frac{r}{\sqrt{2}}$. Therefore, we can achieve $k\tau$ network lifetime by using each of the k sensors in a cell for $\frac{1}{k}$ fraction of the network lifetime. The interesting point here is that the scheduling problems for coverage are often NP-complete in random networks [75]. However, this problem can be easily solved in our hybrid network structure.

5.5 Chapter Summary

In this chapter, we considered using mobile sensors with limited mobile to improve the network coverage. We first defined a network metrics called over-provisioning factor to measure the coverage efficiency of sensor networks. We showed that the over-provisioning factor of randomly deployed static sensor network grows unbounded as the network size grows. This means that static sensor networks cannot cover the field efficiently.

We investigated the upper bound on movement distance for an all-mobile sensor network to cover the whole field and showed that the distance increases with the network size as $O(\log^{3/4} L)$. We then constructively showed that a hybrid sensor network with a small fraction of $O(1/\sqrt{k})$ mobile sensors can also achieve full coverage with the movement distance in the same scale of $O(\log^{3/4} L)$. This shows that we can reduce the number of mobiles used in the network while not losing much in network performance.

In the discussion part, we considered other alternatives of hybrid mobile sensor structures. We showed that it is possible to further reduce the mobile density while keeping the movement distance as $O(\log^{3/4} L)$. We also demonstrated that our scheme can facilitate the sleep-wake scheduling for network lifetime optimization.

One important problem not addressed in this chapter is the mobility algorithm of mobile sensors. Mobile sensors need to coordinate with each other so that each mobile can be matched to a unique grid point or vacancy. This coordinate problem will be studied in the next chapter.

Chapter 6

Mobility Algorithms for Full Coverage

6.1 Problem Formulation

As shown in section 5.3.1, the maximum moving distance of mobiles is small compared to the network size. However, we still require a coordinated moving schedule for which the maximum movement distance is $O(\log^{3/4} L)$. Simple greedy movement, such as moving mobiles to the nearest vacancy, may fail to fill all vacancies with short distance movements [26]. Since the matching problem is a special kind of network flow problem, we can use a network flow architecture to solve the movement schedule in a distributed manner.

6.1.1 Balancing sensors in cells

Inspired by the network flow model used in [26], we formulate our movement schedule problem as follows: Suppose for each cell i there are n_i static sensors and m_i mobile sensors. The number of vacancies in cell i will be $v_i = [k - n_i]^+$. The problem of moving mobiles to fill the vacancies is similar to traffic flow problem in networks, see Fig. 6.1.

We construct a graph $G(\mathcal{V}, \mathcal{E})$ with each cell as a vertex. We add a directional edge (i, j) from cell i to j when the distance between their center is smaller than

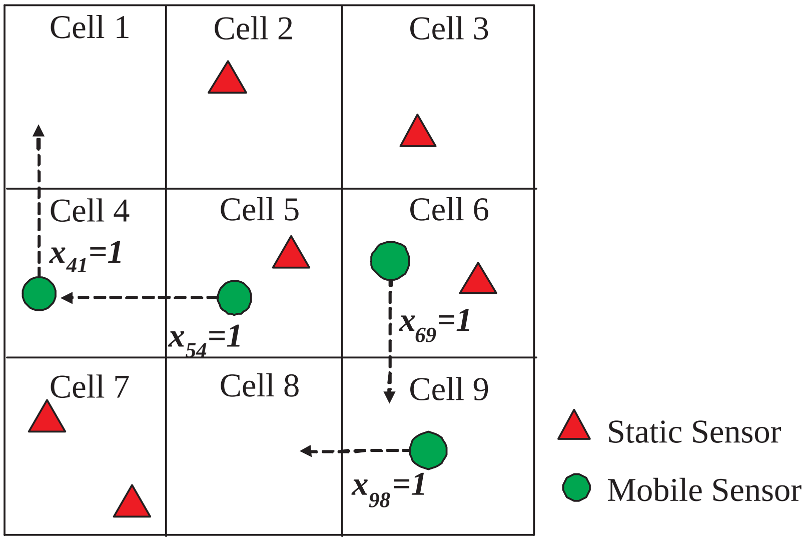


Figure 6.1: Formulating the mobility problem as a network flow problem. The excess mobiles are "flowing" towards the vacancies.

$D = O(log^{3/4}L)$, which is the maximum distance that a mobile sensor can move. The actual value of D can be determined via Fig. 5.2 or Fig. 5.5 when the requirements of coverage failure probability is given for practical networks. Mobiles can move between two cells when the distance between them are smaller than D. Denote the number of mobiles which move from cell i to cell j as x_{ij}, then the movement schedule problem can be formulated as:

$$
\begin{aligned}
& Minimize \sum_{i,j} c_{ij} \quad \times \quad x_{ij} && (6.1)\\
s.t. \quad & \sum_j x_{ji} - \sum_j x_{ij} \quad \geq \quad v_i - m_i \quad \forall i && (6.2)\\
& \sum_j x_{ij} \quad \leq \quad m_i \quad \forall i && (6.3)\\
& x_{ij} \quad \geq \quad 0 \qquad \forall i,j && (6.4)
\end{aligned}
$$

where c_{ij} is the movement cost. In this optimization problem, Eq.(6.2) is the flow conservation condition, which requires the net in-flow for cell i (number of mobiles moving into cell i minus number of mobiles moving out of cell i) to be larger than the number of mobiles it requires (the number of vacancies minus initial number of mobiles in cell i). This constraint guarantees that the final number of mobiles in cell i will be larger than the number of vacancies after the movement. Eq.(6.3) shows that the total number of mobiles moving out of cell i should not be larger than the initial number of mobiles in cell i. The movement cost c_{ij} determines the metrics that need to be minimized. If we set all $c_{ij} = 1$, then the optimal solution will give the movement schedule which has minimum number of movements. If c_{ij} is selected as distance between cells, we will get the scheme with the minimum total moving distance. In this formulation, every mobile will move only once between cells which are not more than D apart. Note that our problem formulation is simpler compared to the formulation in [26].

This problem formulation can work for both the all mobile network and hybrid network. In case of all mobile network, we can simply set $v_i = k$ for all cells in the problem. This formulation can also be applied to irregularly shaped networks by the same graph construction methods as in square networks.

6.1.2 Relationship to network flow problems

We next convert our problem to an equivalent standard network flow problem to show certain important properties of this problem. Note that the linear optimization problem of Eq.(6.1) – (6.4) is similar to the minimum cost flow problem [76], except for the flow conservation constraint of Eq.(6.2). Since in our problem the total number of mobiles $\sum_{i=1}^{2\pi L} m_i$ is slightly larger than the total number of vacancies $\sum_{i=1}^{2\pi L} v_i$ (see section 5.3.1), we make the net in-flow to be larger or equal to $v_i - m_i$ instead of just equal to. If we add a super sink cell c_0 and slack variables x_{i0} to represent the excess number of mobiles, we will get an equivalent problem which is exactly the minimum cost flow problem with the few excess mobiles going to the super sink:

$$\begin{aligned} Minimize \sum_{i,j} c_{ij} &\times x_{ij} && (6.5) \\ s.t. \sum_{j \neq 0} x_{ji} - \sum_{j \neq 0} x_{ij} &- x_{i0} = v_i - m_i, \forall i \neq 0 && (6.6) \\ \sum_{i} x_{i0} &= \sum_{i=1}^{2\pi L} m_i - \sum_{i=1}^{2\pi L} v_i && (6.7) \\ \sum_{j} x_{ij} &\leq m_i \qquad \forall i && (6.8) \\ x_{ij} &\geq 0 \qquad \forall i, j && (6.9) \end{aligned}$$

This minimum cost flow problem has flow capacity constraints on nodes, i.e., the total flow through a node is limited. We can further convert such problem to a traditional network flow problem which has only edge capacity constraints using the well known node-splitting method in network flow problems ([76] page 41–42).

As the constraint matrices for network flow problems are Totally Unimodular ([76] page 447-449), the optimal basic solution x_{ij}^* are integers since v_i and m_i are integers. Therefore, the optimal solution implies that we can just move x_{ij}^* mobile sensors from cell i to cell j to ensure that each cell has at least k sensors in total.

6.2 Distributed Algorithms

In this section, we describe a distributed algorithm to find the movement schedules for the mobiles. To better describe the algorithm, we first provide a distributed algorithm to solve a simpler problem. This algorithm only gives a feasible movement schedule to fill all vacancies without minimizing the total movement cost. We will later show that with several iterations of this algorithm, the minimum-cost flow can also be achieved. Details are at the end of this section.

6.2.1 Push-relabel algorithm

The minimum cost flow problem described in section 6.1.1, gives the optimal sensor movement schedule which minimizes the total movement costs. However, if our goal is only to use mobiles to fill all the vacancies (without minimizing the movement cost), we can just treat the problem as a maximum flow problem, i.e., maximizing the flow from the source to the destination. The solution of the maximum flow will be a feasible movement schedule for mobiles to fill all vacancies when such schedule exists. Several efficient algorithms for the maximum flow problem exist, such as Ford-Fulkerson augmenting path algorithm or the push-relabel algorithm [77]. In this work, we adopt the push-relabel structure which is a naturally distributed algorithm.

We assume that each mobile or static sensor knows its location and knows which cell it is located in. After deployment, mobiles and static sensor in the same cell i communicate with each other to compute v_i and m_i. Each cell elects a mobile or static sensor as the delegator for the entire cell. This sensor stores the necessary information of the cell during the algorithm execution. The delegator of cell i also needs to communicate and exchange information with its neighbors in graph G described earlier. In case there are empty cells, which contain no mobile or static sensors, we can randomly assign a mobile in an adjacent cell as its delegator. Since we have shown the empty cell can be filled with a maximum moving distance of D *w.h.p.*, there will at least be one mobile within distance of D. In case that an empty cell makes the network disconnected, the nearest mobile moves to the empty cell to connect the network before executing the algorithm. In the following discussion, we just use the term "cell" instead of "the delegator of the cell" when an operation needs to be performed.

The basic idea for push-relabel algorithm is to iteratively *push* the excess flow of one vertex to neighboring vertices with lower "heights" or *relabel* itself, which is lift the height of itself, when a push can not be performed. The push and relabel will be repeated until all cells have no excess flow. Details of the original push-relabel algorithm can be found in [77, 78]. Here we only discuss the difference between our algorithm and the original push-relabel algorithm.

- In our algorithm, we use the vacancies as the commodity instead of mobiles. In other words, we push the vacancies from the cells with fewer than k sensors to the cells with free mobiles.

- We have capacity limits on the total flow going through one cell to bound the number of mobiles which can move out of the cell. We adopt the node splitting method to handle capacity bounds on nodes ([76] page 41-42). Each cell i will be split to two vertices i_{in} and i_{out}, the input vertex and the output vertex, respectively. The input vertex is connected with the output vertex by a unidirectional arc (i_{in}, i_{out}) with zero cost and capacity same as m_i, the upper bound on number of mobiles which can move out of cell i. Then, the output vertex i_{out} is connected with neighboring cell's input vertex j_{in} with a unidirectional arc (i_{out}, j_{in}) with cost defined as the c_{ij} in section 6.1.1 and unlimited capacity. Therefore, each cell must maintain two vertices in the push-relabel algorithm. This node splitting method directly comes from the network flow theory [76], and it is simpler than algorithms splitting each cell to three vertices as in [26].

The details of this algorithm are shown in Fig. 6.2. In this algorithm, cells only need to know the heights of vertices in neighboring cells within distance D to perform either push or relabel operation. The push process between i_{in} and i_{out} in the same cell is the same as between different cells except that no message needs to be sent. Note that the push and relabel operations only send messages between cells without actually moving the mobiles. The movements are performed at the end of the algorithm. Each cell will send mobiles to neighboring cells according to the in-flow of their input cells.

6.2.2 Algorithm performance

In our algorithm, every cell needs to communicate only with cells within distance of D and the cell only requires knowledge about these neighboring cells to perform the push and relabel. The network graph has $|\mathcal{V}| = O(L)$ vertices and each vertex has at

```
**Mobility algorithm for cell $i$**
Collect cell information of $v_i$ and $m_i$
Set height of $i_{in}$ and $i_{out}$, denoted as $h(i_{in})$ and $h(i_{out})$ to 0
Set excess of $i_{in}$, denoted as $e(i_{in})$ to 0, and $e(i_{out})$ to $v_i - m_i$
while there exists vertex with positive excess
  Call $Push\text{-}relabel(i_{in})$
  Call $Push\text{-}relabel(i_{out})$
  Update heights of neighboring cells within distance $D$
endwhile
Send mobiles to cell $j$ according the flow on arc $(j_{out}, i_{in})$
```

```
**Push-relabel** (vertex $i$)
If $e(i) > 0$
  while $e(i) > 0$ and exists arc $(i, j)$ s.t. $h(i) = h(j) + 1$ and the residual capacity of
    arc $(i, j)$, $cap(i, j) > 0$.
    Push amount of $y = \min\{e(i), cap(i, j)\}$ through arc $(i, j)$ by sending a message to
      the cell associated to $j$
    $e(i) = e(i) - y$; $e(j) = e(j) + y$; update $cap(i, j)$.
  endwhile
  If $e(i) > 0$
    Update $h(i)$ as $1 + \min\{h(j) : cap(i, j) > 0\}$
    Broadcast $h(i)$ to neighboring cells within distance $D$
  endIf
endif
```

Figure 6.2: Distributed mobility algorithm

most $\pi D^2 = O(\log^{3/2} L)$ arcs. Therefore, the number of edges is $|\mathcal{E}| = O(L \log^{3/2} L)$. Since asynchronous distributed push-relabel algorithm runs in $O(|\mathcal{V}|^2)$ time and uses at most $O(|\mathcal{V}|^2|\mathcal{E}|)$ message exchanges [78], our algorithm takes at most $O(L^2)$ running time and the number of messages exchanged is $O(L^3 \log^{3/2} L)$. Later, via simulations, we will show that this bound is quite loose since the actual running time scales nearly linearly with the network size. Since our algorithm is executed in the delegators of the cells, the complexity of our algorithm scales with the network size (number of cells in the network) instead of the number of sensors in the network. When the network density increases, only the values of m_i and v_i in the algorithm changes and the algorithm complexity remains the same.

If the minimum movement cost schedule is required instead of an arbitrary feasible movement schedule, we can use the cost scaling algorithm proposed by Goldberg *et al.* in [79]. The algorithm uses $O(\log(LC))$ iterations of push-relabel processes to refine the cost of the solution, where C is the maximum cost for any edge. In our problem, our edge cost has positive integer values bounded by $D = O(\log^{3/4} L)$. Therefore, finding the minimum movement cost schedule takes $O(\log L)$ times more computational time and message exchanges than finding an arbitrary feasible movement schedule.

We study the performance of the synchronous push-relabel algorithm which finds a feasible movement schedule without optimizing the total movement cost. The execution process is divided into rounds which contain two phases. In the first phase of each round, cells push excess flow to adjacent cells. If a relabel process is needed, the cell will relabel itself and inform neighboring cells at the second phase. In real networks, we can use an asynchronous algorithm which uses acknowledgement messages for push messages to relieve collisions in information updates [78]. Here, we study the synchronized version, which has similar performance as the asynchronous algorithm when collision rate is low. We execute the algorithm on 10^3 randomly generated topologies to get the average and the maximum running time in all topologies.

Fig. 6.3 shows the number of rounds required by the algorithm. Although the upper bound of running time is $O(L^2)$, the simulation shows that both the average and maximum running time increase linearly with L. For networks with 2500 cells, we need to use on average 800 rounds to get the solution. Fig. 6.4 gives the number of messages used in the algorithm. By curve fitting, the number of messages increases empirically as $O(L^{1.4})$ when the network size increases, which is also much smaller than the bound. Note that the number of messages in Fig. 6.4 is the sum of messages send by all cells in the network. When normalized by the number of cells in the network, the average number of messages sent by a single cell only increases sublinearly with the network size L. This hints that our problem is simpler than network flow problems on general graphs, since the mobiles only move to cells within distance of $O(\log^{3/4} L)$.

As this algorithm only executes once after the deployment, the transmission cost can be amortized over the lifetime of the network and become negligible in small networks. However, the algorithm may still consume considerable energy when the network size is extremely large. In that case, the role of delegator can be rotated

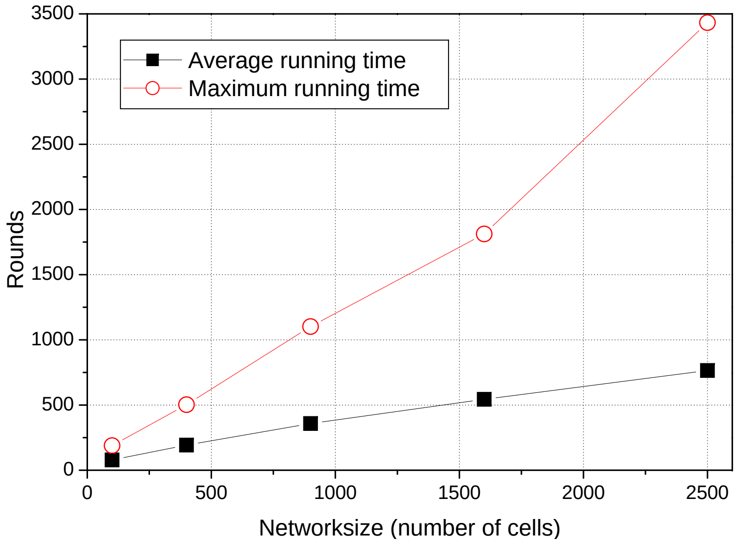


Figure 6.3: Number of rounds used in push-relabel algorithm

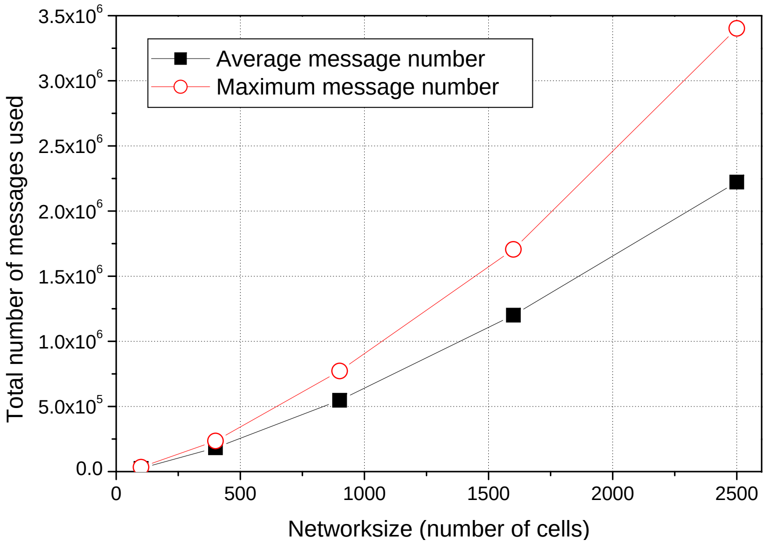


Figure 6.4: Total number of messages used in push-relabel algorithm

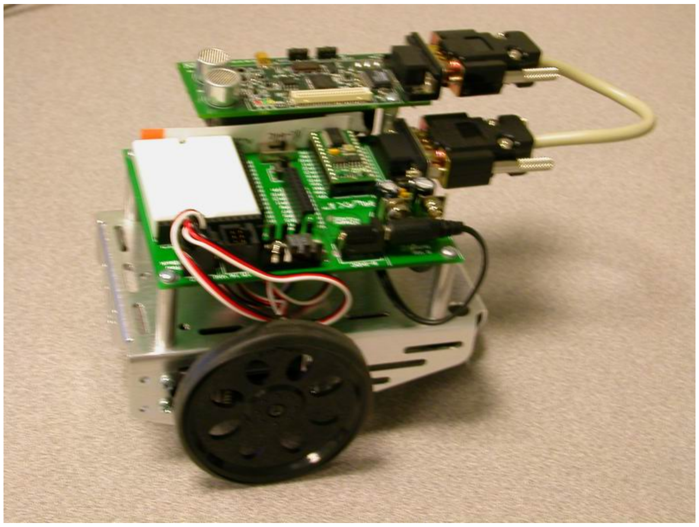


Figure 6.5: The mobile sensor used in implementation.

between nearby sensors when a single cell is extremely highly loaded in the message exchange process.

6.3 Implementation

6.3.1 System description

To demonstrate that our mobility algorithm is implementable and feasible in real world situations, we implemented the push-relabel algorithm on real mobile sensor platforms.

Our mobile sensor platform uses an off-the-shelf mobile robot, called Boe Bot [80], to carry Cricket Motes developed at MIT [56]. Fig.6.5 shows the mobile sensor platform. The Cricket Mote acts as the brain of the mobile sensor. It has one 4MHz ATmega128L processor with 128KB instruction memory and 4KB RAM. It can also communicate with other mobile sensors through the embedded CC1000 radio transceiver. The Cricket platform is a popular indoor localization system for sensors. It measures time difference of arrival of the radio signal and ultra-sound to perform range-based localization. The push-relabel algorithm is also implemented on the Cricket Mote. The Cricket Mote is connected to the Boe Bot through the serial port. The Boe Bot accepts simple instructions, such as "turn left" or "move forward", to perform actual movements.

The algorithm is implemented as described in section 6.2.1. After deployment, the mobile sensors first calculate their location using the Cricket system[1]. Mobiles then send their locations to neighbors to obtain the number of mobiles in each cell. The delegates of each cell execute the push relabel algorithm in a distributed fashion. Each cell maintains the height and excess of its own input and output vertexes. The cell also needs to keep track of the height of its neighboring cells defined as in section 6.2.1. A push message is sent to the destination cell when a push operation is needed, and the push operation is successful only when the destination cell replies with a positive acknowledgement. If the cell relabels its vertices, it needs to broadcast the new height in a relabel message to its neighbors. The algorithm will terminate when the excess of all cells are zero. This can be inferred by individual mobiles when it has not heard any push or relabel message in the neighborhood for more than three iterations. After that, the delegate of the cell will dispatch mobiles according to results of the distributed algorithm to achieve the optimal deployment.

[1] Unlike the original Cricket system, our triangulation algorithm is implemented on individual Cricket Motes.

We build up a test bed with nine mobile sensors on a field with 3×3 cells with cell side length of 60 cm. The movement speed of mobiles is around 10 cm/second. For energy consumption, the mobile use 4 AA batteries and each servo used on the mobile consumes 12 mA when idle and 180 mA when operation with no load at 6V [80]. Therefore, the total enemy consumption of the robot (which has two servos) is larger than 2W which is much larger than the power consumption of MICA2, which only draws 12 mA at 3V (0.036W) when transmitting [5].

Each mobile is allowed to move across one cell to achieve the optimal deployment which has exactly one mobile in each cell. In this case, at most four adjacent cells will be considered as neighbors. Since cells only need to keep track the height of neighboring cells, the memory requirement of our algorithm is quite low. As shown in Tab 6.1, the push-relabel algorithm uses only 10K bytes ROM and less than 300 bytes RAM. In the field test, we see that mobiles can execute the algorithm and redeploy within one minute when the optimal solution exists, see the video available on line [81].

Communication links in the real world are unreliable, where mobiles may randomly lose data packets. Losing important messages, such as push, relabel or push acknowledgment, may cause information inconsistency in neighboring cells. However, the problem can be solved by carefully designed algorithms. In our implementation, a cell always maintains the correct information about excess and height of itself. When a push message from neighboring cell contains incorrect information, i.e., wrong information about the heights, the cell will reject the push and send back the correct

Table 6.1: Code size of the mobility algorithm

Program	ROM (bytes)	RAM (bytes)
Basic Cricket code (modified)	32,236	2,436
Added for mobility algorithm	10,218	267
Total	42,454	2,703

information. Our tests in both real world and TOSSIM simulation environment show that the algorithm can work under a packet loss rate of more than 30%.

6.3.2 Discussions on real world deployment issues

The sensing region may be imperfect due to barriers or noise in real deployments. Although our analysis in section 5.2.1 and 5.3.1 assumes the sensing regions are perfect disks with uniform radius, our results can be easily modified when sensing regions are imperfect. Our algorithm achieves a regular number of sensors in each square cell, irrespective of the shape of sensing region. If the sensing region is not a perfect disk, we can find a proper size for the cells so that each cell can be fully covered by any sensor within it. Our analysis and algorithm can also be applied to other types of grids, such as triangular, rectangular or hexagonal grids. We can choose among these cell shapes according to the sensing model used in deployment.

When certain areas require more sensors due to high sensing noise level, our algorithm can also be used to achieve a non-uniform deployment of sensors. Suppose more sensors are required to be deployed in a particular cell i than other cells. We can increase the number of vacancies, v_i, in cell i in our algorithm to find the movement schedule which sends more sensor to cell i. However, the maximum movement distance bounds may no longer hold in this case.

In previous sections, we assume that the sensing field is a $l \times l$ square. Our analysis and algorithm can also be extended to irregularly shaped network fields with barriers and holes. When the network field is not regular, we can divide the field into cells as in section 6.2.1. In this case, the network size L will be defined as the number of cells in it. We can construct the graph for movement schedule as in section 6.2.1. If there are barriers which mobiles can not pass through, we just remove the edges crossing such barriers when constructing the graph. Then, the movement schedule can be found on the constructed graph by the algorithm described in section 6.2.1.

The result of the maximum movement distance bound of $O(\log^{3/4} L)$ will still hold for certain irregularly shaped network fields. For example, the field can contain sparse convex holes with side length smaller than $O(\log^{3/4} L)$. When the maximum movement distance is $O(\log^{3/4} L)$, mobiles can bypass these holes when holes are not clustered together. Loosely speaking, the requirement on the shape of the field is that every sub-region in the network field should have no less than a constant factor of its perimeter connected to other parts of the field, see the proof of Theorem 8 in the appendix. Therefore, the network field can be disks or convex polygons with side length larger than $O(\log^{3/4} L)$. However, if the field contains a thin strip with width smaller than $O(\log^{3/4} L)$ and length of $O(L)$, the maximum movement distance bound will no longer hold.

6.4 Chapter Summary

In this chapter, we proposed a distributed mobility algorithm for network coverage. We first formulated the mobility algorithm as a linear programming problem and connected it to network flow problems. We then used a distributed push-relabel algorithm to solve this problem. In this algorithm, sensor only needs to have information about its neighborhood. Using the results in Chapter 5, we can see that the neighborhood size grows with the network size slowly as $O(\log^{3/2} L)$. We also provided analysis and simulation results on the computational complexity of the distributed algorithm and showed that it only grows polynomially with the network size.

We showed that the distributed algorithm is simple and can be implemented in real mobile sensors. In this chapter, we also discussed issues about deploying mobile sensors in real world network, where the network shape may be irregular.

In the implementation of mobile sensors, we used a localization system called Cricket system to provide location information to mobile sensors. Such localization system uses beacons with known coordinates to localize the mobiles. In the next chapter, we will further discuss localization systems for mobile sensors and derive a new model for beacon density estimation.

Chapter 7

Localization of Mobile Sensors

7.1 Introduction

In mobile sensor networks, it is important for mobile sensors to know their own locations. First, mobile sensors need to know their own locations to navigate their movements. Second, location information of mobile sensors is vital for network traffic optimization in both the mobile sink approach [9] and the mobile relay approach [10]. Furthermore, location information of mobile sensors can also be useful for other network operations, such as tagging the sensing data gathered by the mobile sensors.

There are two methods for mobile sensor localization. First, we can use external localization systems, such as the GPS system. Second, sensors with known locations can serve as beacons for localizing mobile sensors in the network. As GPS devices normally are expensive and consume considerable energy, we choose the second approach in this work.

When we use static sensors as beacons for localization, one important issue is to determine the necessary density of beacons which can provide accurate localization over the field. If we treat the range of the beacon in the localization problem as the sensing range of a sensor, the density estimation problem can be converted to a coverage problem in sensor networks. In other words, we need to determine the sensor density which can fully cover the network field for localization applications.

The most commonly used model in coverage problems is the disk model, which assumes that the sensing region for a sensor is a circular region centered at it. A point in the field is said to be covered if it is within the sensing region of some sensor. The disk model can be used for coverage problems in target detection applications. However, it has certain limitations in describing the coverage for localization applications. Ensuring that the region is covered by sensing disks can not provide guarantees for precise localization. Even with the concept of *k-coverage*, where every point should be within the sensing range of at least k sensors, the localization error still cannot be tightly bounded. Consider the case where a point is covered by k sensors clustered at one point. The k sensors only provide redundant location information about the mobile sensor when the mobile is at that point. In this case, the localization error at a k-covered point could still be quite large.

Due to the insufficiency of the disk model, it may overestimate or underestimate the node density required for localization coverage. In this work, we demonstrate the insufficiency of the disk model in localization applications and introduce a new coverage estimation method for localization coverage, which is called sector coverage.

7.2 Sensor Density for Localization

7.2.1 System model

We assume that static sensors are used as beacons to provide accurate location information to mobile sensors. We assume that the coordinates of static sensors are already known, and the location of the mobile sensor is estimated based on the measurements and coordinates of nearby static sensors. We also assume that mobile sensors are not using knowledge of their previous locations and movement speed to enhance the localization accuracy. We focus on static sensors which can measure their distance to the mobile sensor, such as Time Difference Of Arrival (TDOA) and Received Signal Strength Indicator (RSSI) sensors.

Due to the existence of noise, the distance estimation will be distributed within a certain range around the true distance. When the true distance between beacon s_i and the mobile sensor m is $d_{i,m}$, we assume the measured distance $\widetilde{d_{i,m}}$ for beacon i will fall in the range $[d_{i,m} - \sigma_l, d_{i,m} + \sigma_u]$, where σ_l and σ_u are the error bounds. We set σ_l, σ_u equal to σ in latter derivation. However, our result can also be easily extended to the case where σ_l and σ_u are not equal. We assume that a beacon can only provide distance measurements when the mobile is within its range r. We define *normalized ranging accuracy* of a beacon as $K = r/\sigma$, which is the ratio of the beacon range compared to the maximum measurement error. Under the assumptions we made, when beacon s_i gives a distance estimate of $\widetilde{d_{i,m}}$, the mobile will fall in $[\widetilde{d_{i,m}} - \sigma, \widetilde{d_{i,m}} + \sigma]$ with high probability. Then, a single beacon i can localize the

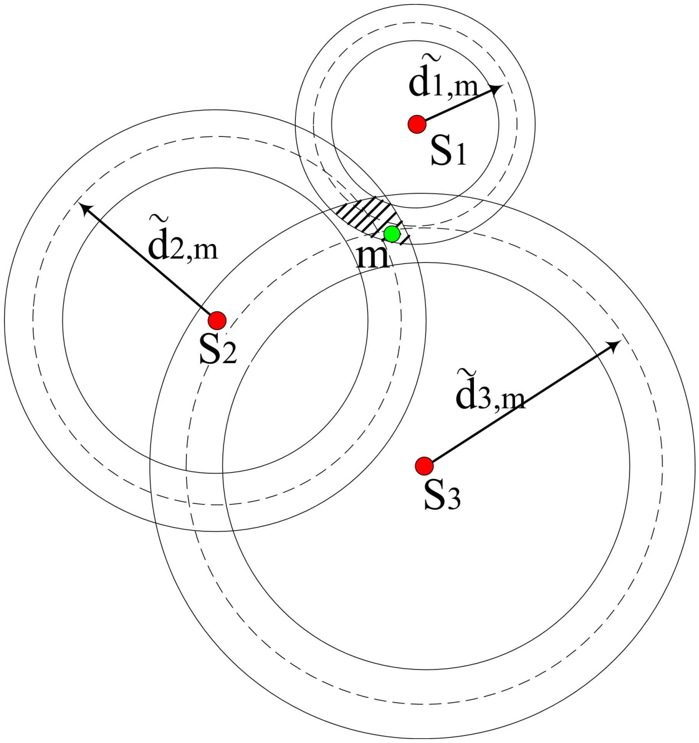


Figure 7.1: Each of the three beacons s_1,s_2 and s_3 can provide distance information of the mobile.The mobile location will be within the intersection of the annuli around them, shown as the shaded area.

mobile within an annulus of $[\widetilde{d_{i,m}} - \sigma, \widetilde{d_{i,m}} + \sigma]$, as shown in Fig. 7.1.

When the mobile can receive signals from several beacons at the same time, the position estimation can be further refined through intersecting their estimations [82]. In this case, we can combine the distance measurements by localizing the mobile within the intersection area of the annuli shown in Fig. 7.1, which is defined as the high probability uncertainty region. The uncertainty region will be reduced as more beacons provide distance information about the mobile. Given the group of distance measurement $\widetilde{d_{1,m}}, \widetilde{d_{2,m}}, \ldots, \widetilde{d_{k,m}}$ provided by all the k beacons within range

r to the mobile, we can determine the smallest uncertainty region. The estimated mobile location can be set as center of the smallest circle which can circumscribe the uncertainty region. Thus, the localization error is smaller than the radius of this circle, since the distance from any point in the uncertainty region to the center is smaller than the radius. If the beacon density is high enough, we can make all uncertainty regions small enough to be well contained in circles with radius smaller than ε. In this case, the localization error is always smaller than ε, no matter where the mobile is. Note that our localization method is much simpler compared to state-of-the-art localization algorithms. However, this simplified model retains the basic ideas of range based localization while at the same time reveling key insights and relationships between the coverage and localization problem.

Now we define the concept of coverage under this localization model. We require the location estimation error to be within a circle of radius ε irrespective of the location of the mobile. In a randomly deployed sensor network, there may exist areas where local sensor density is so low that we can not precisely localize the mobile in these areas. The *coverage hole* is defined as those holes where the localization error exceeds the predefined bound. The objective of this work is to find the sensor density which can ensure there is no coverage hole in the network, or the area of coverage hole is small compared to the total area of the field.

7.2.2 Sufficient coverage conditions

In this section, we give a sufficient condition for the localization error to be bounded over the whole field. We first introduce the concept of network resolution on localization. We say that two points m and m' are distinguishable if the sensor network can always distinguish whether a mobile is at point m or at point m' through the distance measurements provided by static sensors. The *network resolution* is the minimum distance u, such that the network can distinguish any pair of points when the distance between them is larger than u. The network resolution is related to the localization error bound ε by the following lemma.

Lemma 2 *Given the network resolution of u, the localization estimation error ε is upper bounded by $u/\sqrt{3}$ and lower bounded by $u/2$.*

Proof: We see that points in the same uncertainty region cannot be distinguished by the network. If the network resolution is u, then every uncertainty region should not contain two points apart by more than u. Thus, the *Generalized Diameter*, which is defined as the greatest distance between any two points in the shape, is smaller or equal to u for all uncertainty regions. A hexagon with side length of $u/\sqrt{3}$ can fully cover any shape with Generalized Diameter smaller than u [83]. Thus, the circumcircle of such a hexagon, which has a radius of $u/\sqrt{3}$, can also cover any uncertainty region in the network. Using the center of the covering circle as the estimated location of the mobile will provide estimation error smaller than $u/\sqrt{3}$. This bound is tight since when uncertainty region is shaped as an equilateral triangle with side length of u, the smallest circle which can circumscribe it has radius of exactly $u/\sqrt{3}$, see Fig. 7.2.

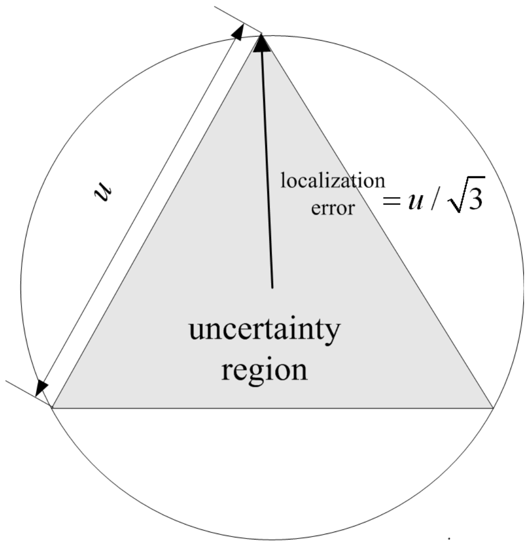


Figure 7.2: Relationship between network resolution and localization error

If the network resolution is u, then there should exist two points in an uncertainty region that are apart by u by definition. Such an uncertainty region cannot be circumscribed by circles with radius smaller than $u/2$. Thus, the estimation error is larger than $u/2$. ∎

Lemma 3 *For two points m and m', if there exists a beacon s_i whose distance to these two points satisfies:*

$$|d_{i,m} - d_{i,m'}| \geq 2\sigma \tag{7.1}$$

then beacon s_i can definitely distinguish point m from point m', given that at least one of these two points is in the range of s_i ($d_{i,m} < r$ or $d_{i,m'} < r$).

Proof: Without loss of generality, suppose the true mobile position is at m. According to the assumption of distance measurement error, the distance measurement

$\widetilde{d_{i,m}}$ provided by beacon s_i should satisfy:

$$|\widetilde{d_{i,m}} - d_{i,m}| \leq \sigma \tag{7.2}$$

Since $|d_{i,m} - d_{i,m'}| \geq 2\sigma$, we have:

$$|d_{i,m'} - \widetilde{d_{i,m}}| \geq |d_{i,m'} - d_{i,m}| - |d_{i,m} - \widetilde{d_{i,m}}| \geq \sigma \tag{7.3}$$

Thus, the point m' will be outside the uncertainty annulus of $[\widetilde{d_{i,m}} - \sigma, \widetilde{d_{i,m}} + \sigma]$ and beacon s_i can determine that the mobile is not at point m' based on distance measurement of $\widetilde{d_{i,m}}$. ∎

Theorem 10 *If there is at least one beacon in any arbitrarily selected sector of radius r (the predefined beacon range) and angle $\frac{2\pi}{3}$, denoted as a sector of $\{r, \frac{2\pi}{3}\}$, then the location estimation error is bounded by $\varepsilon = \frac{4\sqrt{3}\sigma}{3}$ over the network, where σ is the maximal distance estimation error when the mobile is within the beacon range r.*

Proof: As shown in Lemma 2, if the network resolution $u \leq \sqrt{3}\varepsilon = 4\sigma$, we can guarantee a location estimation error bound of $\varepsilon = \frac{4\sqrt{3}\sigma}{3}$. We will prove this by contradiction. Suppose the network resolution is worse than 4σ, which means we can find at least one pair of points m and m' which are apart by more than 4σ, but no beacon can distinguish them.

Suppose points m and m' cannot be distinguished by the network, and the distance between m and m' is 4σ. As we assumed, there is at least one beacon in every arbitrarily selected sector of $\{r, \frac{2\pi}{3}\}$. So, there must be at least one beacon within range r of point m. Suppose this beacon is static sensor s_i. Then the point m' should also be in the range of beacon s_i, otherwise these two points can not be in the same

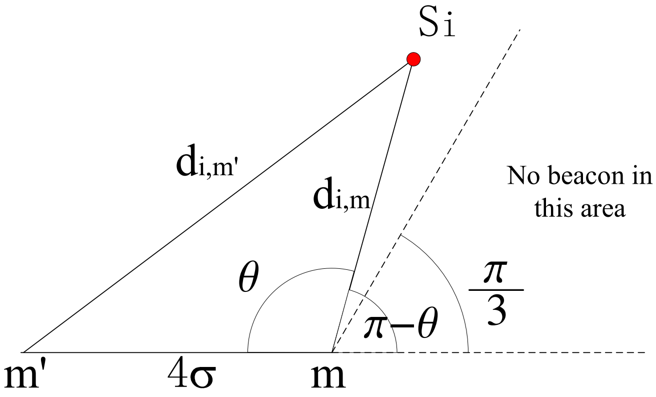


Figure 7.3: Both point m and m' are in the range of beacon s_i, but s_i cannot distinguish them.

uncertainty region and the network can distinguish them. Without loss of generality, suppose beacon s_i is closer to the point m, i.e., $d_{i,m} < d_{i,m'}$, as shown in Fig. 7.3. From Lemma 3, we have $d_{i,m'} \leq d_{i,m} + 2\sigma$ when s_i can not distinguish m and m', then:

$$\cos\theta = \frac{d_{i,m}^2 + (4\sigma)^2 - d_{i,m'}^2}{2 \times d_{i,m} \times 4\sigma} \geq \frac{12\sigma^2 - 4d_{i,m}\sigma}{8d_{i,m}\sigma} \geq -\frac{1}{2} \tag{7.4}$$

Since $\cos\theta$ is monotonically decreasing in $[0, \pi]$, the angle θ must be smaller than $\arccos(-\frac{1}{2}) = \frac{2\pi}{3}$, so the angle $\pi - \theta$ is larger than $\frac{\pi}{3}$. Therefore, no beacon should be in the area with angle $\pi - \theta$ smaller than $\frac{\pi}{3}$ as shown in Fig. 7.3. By symmetry, no beacon can be within the sector of $\{r, \frac{2\pi}{3}\}$ centered at m and bisected by ray $m'm$. This is a contradiction to the assumption that there should be at least one beacon in any arbitrarily selected sector of $\{r, \frac{2\pi}{3}\}$.

Eq. (7.4) considers the case that $d_{m,m'} = 4\sigma$. If $d_{m,m'} > 4\sigma$, this lower bound for $\pi - \theta$ increases monotonically. Thus, if the two indistinguishable points m and

m' are separated by more than 4σ, there also should be no beacon in the sector of $\{r, \frac{2\pi}{3}\}$ centered at m, which contradicts our assumption. Therefore, there does not exist indistinguishable points m and m', which are apart by more than 4σ. Then, the network resolution is better than 4σ, which directly leads to a localization error bound of $\frac{4\sqrt{3}\sigma}{3}$ by lemma 2. ∎

Theorem 10 can be directly extended to the disk coverage model as follows.

Corollary 1 *Given a sensor deployment, if disks of radius* $\frac{\sqrt{3}}{\sqrt{3}+2}r$ *centered at the beacons can cover the entire field, then the location estimation error is bounded by* $\varepsilon = \frac{4\sqrt{3}\sigma}{3}$ *over the network.*

Corollary 1 comes from the fact that a sector of $\{r, \frac{2\pi}{3}\}$ contains an inscribed circle of radius $r_d = \frac{\sqrt{3}}{\sqrt{3}+2}r \approx 0.464r$, as shown in Fig 7.4[1]. If there exists a $\{r, \frac{2\pi}{3}\}$ sector void, then there is no beacon within a radius of r_d from the center o of the inscribed circle. Therefore, if the beacon density is high enough that the field is totally covered by disks of radius r_d centered at beacons, then there will be no $\{r, \frac{2\pi}{3}\}$ sector void and the location estimation error bound is guaranteed. Thus, we can directly use known results in disk coverage for localization applications by shrinking the radius of coverage disk to r_d.

Corollary 1 shows that we need to deploy beacons at a higher density to localize the mobile than detecting it. We can estimate the density required for localization coverage as follows. Assume we have to provide coverage over a square field of area A. If we look at a square field with sides scaled by a factor of 0.464, the number of

[1] Note that Eq. (7.4) gives a lower bound of the void. The white area drawn here, which is derived by Eq. (7.1), is larger than a $\{r, \frac{2\pi}{3}\}$ sector.

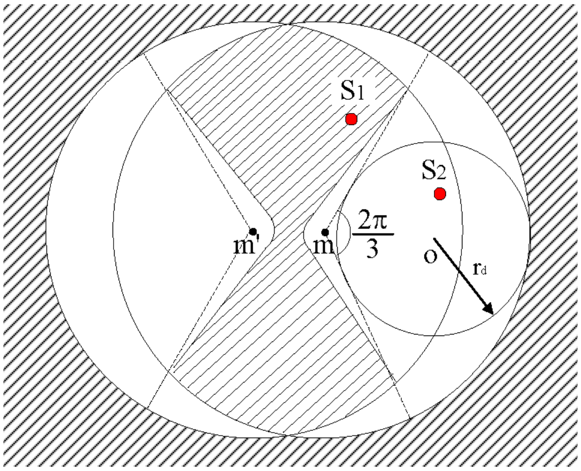


Figure 7.4: Beacons in the shaded area, such as s_1, cannot distinguish the point m from m'. Beacons in the white area, such as s_2, can distinguish these two points, so there should not be any beacon in the white area.

beacons required to cover this field with beacon radius r_d is the same as the number of beacons required to cover the area of A with radius r. However, the same number of beacons only cover an area of $0.464^2 A$ with a shrunk beacon radius of r_d. Therefore the beacon density required for localization coverage is $\frac{1}{0.464^2} \approx 4.64$ times more than that required for a detection coverage, which only requires the area to be covered by disks of radius r.

Using beacon range larger than r_d in the disk model may violate the conditions in Theorem 10. Consider the case where beacons are densely deployed on the border of the shaded area in Fig. 7.4. With disks larger than r_d, we can get the area disk covered (or even k-covered). However, the conditions in Theorem 10 are violated. Those beacons in the shaded area can not distinguish point m from m' and the localization error can exceed the bound. Thus, shrinking the disk radius to r_d is necessary in disk model.

7.2.3 Relationship between Resolution and Density

Theorem 10 shows how to guarantee network resolution of 4σ. As the uncertainty region of a single beacon is an annulus with width of 2σ, we may further improve the network resolution to 2σ by increasing the beacon density. However, we show below that this will not be cost efficient.

Suppose that we require a network resolution of $u = \alpha\sigma$. Here we have $\alpha \geq 2$, since the network resolution is lower bounded by the annulus width of 2σ used in localization. Substituting this into Eq. (7.4), we have:

$$\cos\theta = \frac{d_{i,m}^2 + (\alpha\sigma)^2 - d_{i,m'}^2}{2 \times d_{i,m} \times \alpha\sigma} \geq \frac{(\alpha\sigma)^2 - 4d_{i,m}\sigma - 4\sigma^2}{2\alpha d_{i,m}\sigma} \geq -\frac{2}{\alpha} \tag{7.5}$$

When $d_{i,m}/\sigma$ is large, θ can be exactly equal to $\arccos(-\frac{2}{\alpha})$. As α decreases from 4 to 2, $\arccos(-\frac{2}{\alpha})$ grows from $2\pi/3$ to π. As shown in the proof of Theorem 10, we require every sector of $\{r, 2(\pi-\theta)\}$ to contain at least one beacon. For small values of α, the angle of such sectors decreases to zero; thus an extremely high beacon density is required to drive network resolution close to 2σ.

To demonstrate this, we investigate the relationship between α and beacon density under the disk coverage model. The maximal radius for a circle which can be packed in the $2(\pi - \theta)$ sector is $\sin\theta/(1 + \sin\theta)$. Based on this, the relationship between beacon density and network resolution is plotted in Fig. 7.5. When the resolution requirement is close to 2σ, beacon density increases quickly, yet the gain in network resolution is small. On the other hand, when network resolution is worse than 4σ,

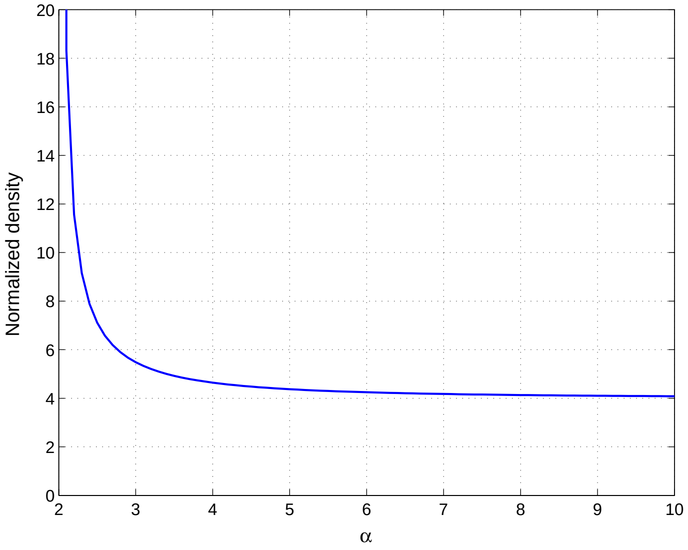


Figure 7.5: The relationship between network resolution and beacon density. α is network resolution divided by σ, the distance measurement error. The normalized density is ratio of localization coverage compared to detection coverage.

beacon density remains nearly constant and it converges to 4 times the detection density as α goes to infinity. The radius of the beacon is upper bounded by $0.5r$ when α is increased. This means that we need at least one beacon in each disk of radius $0.5r$ to meet the basic density requirement for localization, e.g., the mobile is within the range of at least 3 beacons at appropriate positions for triangulation. Fig. 7.5 hints that a network resolution of 3σ to 4σ is the best trade-off between beacon density and network resolution. Therefore, when the ranging accuracy for sensors is low, we cannot simply increase the beacon density to achieve high network resolution.

7.3 Sector Coverage

7.3.1 Density estimation through sector coverage

From the proof of Theorem 10, we can see that when the localization error exceeds the bound, there are voids shaped as the white area in Fig. 7.4. Showing that there is no disk shaped void of radius r_d in the network can eliminate the existence of such sector-shaped voids. However, a network may not contain such sector shaped voids even if it is not covered by disks of radius r_d. Thus, the sufficient condition derived by disk coverage is too strict. It may give a higher estimation on beacon density.

In this section, we introduce the concept of *sector coverage* to provide a better estimate on beacon density required for a randomly deployed network to guarantee a bounded localization error. The derivation is based on the assumption that beacons are randomly scattered in the field with distribution of a stationary Poisson point process with intensity λ [67]. We use the average vacancy over a unit area, which is often called porosity in coverage theory [67], to measure the quality of coverage. The average vacancy over a unit area is defined as the ratio of uncovered area divided by the total area of the field. It is equal to the probability that an arbitrary point is not covered, i.e., the probability that localization error exceeds the bound at that point.

Instead of estimating whether there are voids shaped as the white area in Fig. 7.4, we approximate the voids using two opposing sectors of $\{r, \frac{2\pi}{3}\}$ around point m. We call this approximation *sector coverage*: a point is sector covered if there is at least one beacon in any pair of opposing sectors of $\{r, \frac{2\pi}{3}\}$ around it in any arbitrary orientation. We need to check sectors in all orientations since the sufficient condition in Theorem 10 can be violated when there exists void of any orientation.

The approximation of sector coverage is based on the following observations. Suppose that m is uncovered. Then there will be a point m' which cannot be distinguished from point m and $d_{m,m'}$ is larger than 4σ. Following the arguments in Theorem 10, there should also be a sector void of $\{r, \frac{2\pi}{3}\}$ around m' by symmetry. When the ranging accuracy K is large, the two white areas around m and m' can be treated as two opposing sectors of $\{r, \frac{2\pi}{3}\}$ around point m.

The approximation can be broken into two steps. First, we approximate the two white areas around m and m' in Fig. 7.4 by two $\{r, \frac{2\pi}{3}\}$ sectors at point m and m'. When K is large, we have $r \gg \sigma$. The angle θ defined in Eq. (7.4) converges to $\frac{2\pi}{3}$ as $d_{i,m}$ approaches r, which means the border of the shaded area in Fig. 7.4 will overlap with the border of the sectors. Furthermore, as $d_{m,m'}$ is small compared to r, the two circles will almost overlap with each other. Therefore, the difference between the $\{r, \frac{2\pi}{3}\}$ sector and the white area in Fig. 7.4 can be ignored in this case. Note that the white areas are strictly larger than the two $\{r, \frac{2\pi}{3}\}$ sectors for all K values by Theorem 10. Therefore, when K is small, our approximation can serve as an upper bound for the probability that a point is not covered.

Second, we approximate the two sectors around m and m' as a pair of opposing sectors centered at point m. This approximation is valid since the beacons are distributed as a Poisson point process. In this case, the beacon distribution in disjoint areas are independent. Thus, the probability that there exist voids of a pair of opposing sectors centered at point m is same as the probability that there exist voids of two sectors which are apart by $d_{m,m'}$, given that the sector orientation and $d_{m,m'}$ are fixed [67]. Section 7.3.2 will further validate our approximation by numerical examples.

We now proceed to find the density requirement for sector coverage. Consider a Poisson point process with intensity λ. The probability that there are k beacons in range r of point m is given by:

$$p_k = e^{-\pi\lambda r^2}\frac{(\pi\lambda r^2)^k}{k!} \tag{7.6}$$

For beacons falling within the range r, we denote the beacon's position as (a_i, ϕ_i) in polar coordinates with the origin at m. Using the property of Poisson process, the ϕ_is are independently and uniformly distributed over $[0, 2\pi]$ given there are k beacons in the circle.

If there is no more than one beacon in range r, we can always find two opposing

$\{r, \frac{2\pi}{3}\}$ sector voids, thus the point m cannot be sector covered in this case. If there are two beacons within range r and the angle between these two beacons falls in the range of $[0, \frac{\pi}{3}] \cup [\frac{2\pi}{3}, \pi]$, then the point m is not sector covered[2]. Since the angle $|\phi_1 - \phi_2|$ is uniformly distributed in $[0, \pi]$, the probability that a point m is not sector covered given there are two beacons in range r is $\frac{2}{3}$.

If there are $k > 2$ beacons in the circle, we first convert the problem of finding two opposing sector voids of $\frac{2\pi}{3}$ to finding one continuous sector void of $\frac{4\pi}{3}$. For every beacon, define ϕ'_i as

$$\phi'_i = \begin{cases} 2\phi_i, & 0 \le \phi_i < \pi, \quad i = 1, 2 \ldots, k \\ 2(\phi_i - \pi), & \pi \le \phi_i \le 2\pi, \quad i = 1, 2, \ldots k \end{cases} \tag{7.7}$$

Two opposing beacons with angle of ϕ_i and $\phi_i + \pi$ will be transformed to the same angle of $\phi'_i = 2\phi_i$. Thus, any beacon deployment that has a void of two opposing sector of $\frac{2\pi}{3}$ will have a void of $\frac{4\pi}{3}$ in the transformed coordinate system of (a_i, ϕ'_i). It is easy to see that the angle ϕ'_i is also independently and uniformly distributed on $[0, 2\pi]$.

For k independently and uniformly distributed ϕ'_i, the probability that the range of the samples, defined as $\max\{\phi'_i\} - \min\{\phi'_i\}$, is smaller than $\frac{2\pi}{3}$ is given by [84]:

$$q_k = k(\frac{1}{3})^{k-1} - (k-1)(\frac{1}{3})^k \tag{7.8}$$

q_k is the probability that all k beacons are confined in a $\frac{2\pi}{3}$ sector with the remaining $\frac{4\pi}{3}$ sector as a void. Since sectors can cross the zero angle, q_k does not account for the case that all k beacons are confined in a sector with an angle smaller than $\frac{2\pi}{3}$

[2] Note that sector coverage is an approximation. A point can be sector covered when there are only two beacons in range r, yet the localization error may exceed the bound.

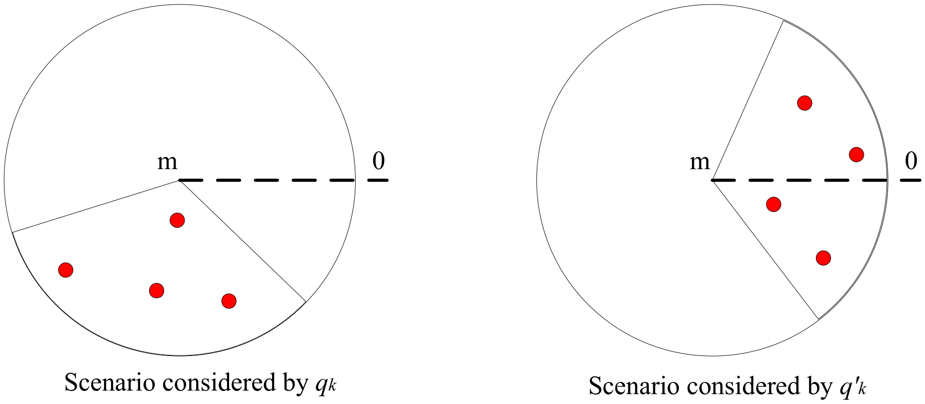


Figure 7.6: Bounding the probability that there exists a sector void of $\{r, \frac{2\pi}{3}\}$, two scenarios are considered based on whether the sector crosses the zero angle.

and it crosses the zero angle (see Fig. 7.6). We define the probability of this scenario as q'_k. It is clear that q'_k is smaller than the probability that k beacons are confined in a $\{r, \frac{2\pi}{3}\}$ sector *and* the sector starts from $[\frac{4\pi}{3}, 2\pi]$. This probability is half the probability that beacons are confined in a $\{r, \frac{2\pi}{3}\}$ sector *and* the sector starts from $[0, \frac{4\pi}{3}]$ due to the uniform distribution of ϕ'_i. Since q_k includes all cases that beacons are confined in a $\{r, \frac{2\pi}{3}\}$ sector *and* the sector starts from $[0, \frac{4\pi}{3}]$, q'_k is upper bounded by $q_k/2$. Combining all these cases, the probability that an arbitrary point m is not sector covered is upper bounded by:

$$p_{sector} \leq p_0 + p_1 + \frac{2p_2}{3} + \frac{3}{2}\sum_{k=3}^{\infty} q_k p_k \tag{7.9}$$

For disk coverage, the probability that one point is not covered is given by:

$$p_{disk} = e^{-\pi\lambda r_d^2} \tag{7.10}$$

which is the chance that there is no beacon in the disk of radius r_d. For a given cover-

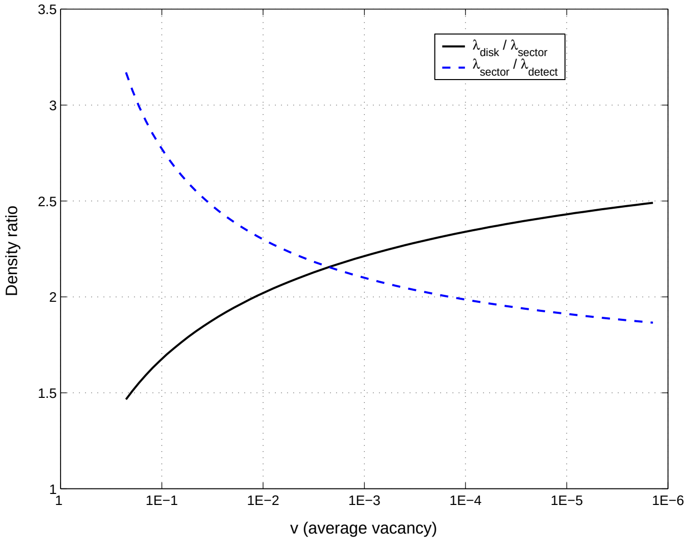


Figure 7.7: Comparing the density requirements for disk coverage and sector coverage.

age requirement of average vacancy v over a unit area, we can calculate the necessary density for both sector coverage and disk coverage, denoted as λ_{disk} and λ_{sector}. We also denote the necessary density for disk coverage in detection based applications as λ_{detect} to compare the density requirement of localization applications and detection applications. Fig. 7.7 shows that the density requirement for sector coverage is much smaller than the disk model. For sector coverage, the density requirement for localization is nearly 1.8 times higher than the density required for detection coverage when v is small. As v becomes smaller, $\lambda_{disk}/\lambda_{sector}$ converges to 2.5, which means sector coverage requires 2.5 times fewer beacons than the equivalent disk coverage. This indicates that disk coverage is not accurate when used for localization applications.

Another important observation is that the value of $\lambda_{disk}/\lambda_{sector}$ is not a constant. This means the sector coverage cannot be approximated by fixed shapes, e.g., two opposite sectors of $\{r, \frac{2\pi}{3}\}$ or a pair of circles of radius r_d with fixed orientations. In that case, if the area of the shape is A_c, the average vacancy will be $e^{-\lambda A_c}$ [67]. So, the required density will simply be $\frac{A_c}{\pi r_d^2}\lambda_{disk}$, which is λ_{disk} multiplied by a constant.

7.3.2 Experimental results

7.3.2.1 Experimental Setting

We use Monte Carlo methods to verify our theoretical analysis. The experimental setting is as follows: We randomly deploy λA beacons in a large region with area A, so we approximately get a Poisson point process with intensity λ in a small area inside A. We check whether there exists a pair of points which cannot be distinguished by the randomly deployed beacons. The minimum distance between two points which cannot be distinguished will be the experimental network resolution. We repeat over 100,000 network instances to obtain the probability that the network resolution is worse than $\alpha\sigma$.

7.3.2.2 Average Vacancy

The average vacancy is the probability that the network resolution is worse than the predefined bound. The experimental average vacancy is obtained through counting the vacancy probability at randomly picked point in different network instances. The result is shown in Fig. 7.8. The sector coverage method provides good estimates for the average vacancy when the ranging accuracy $K = r/\sigma$, is larger than 50. For small K values, the white area in Fig. 7.4 is much larger compared to the $\{r, \frac{2\pi}{3}\}$ sector

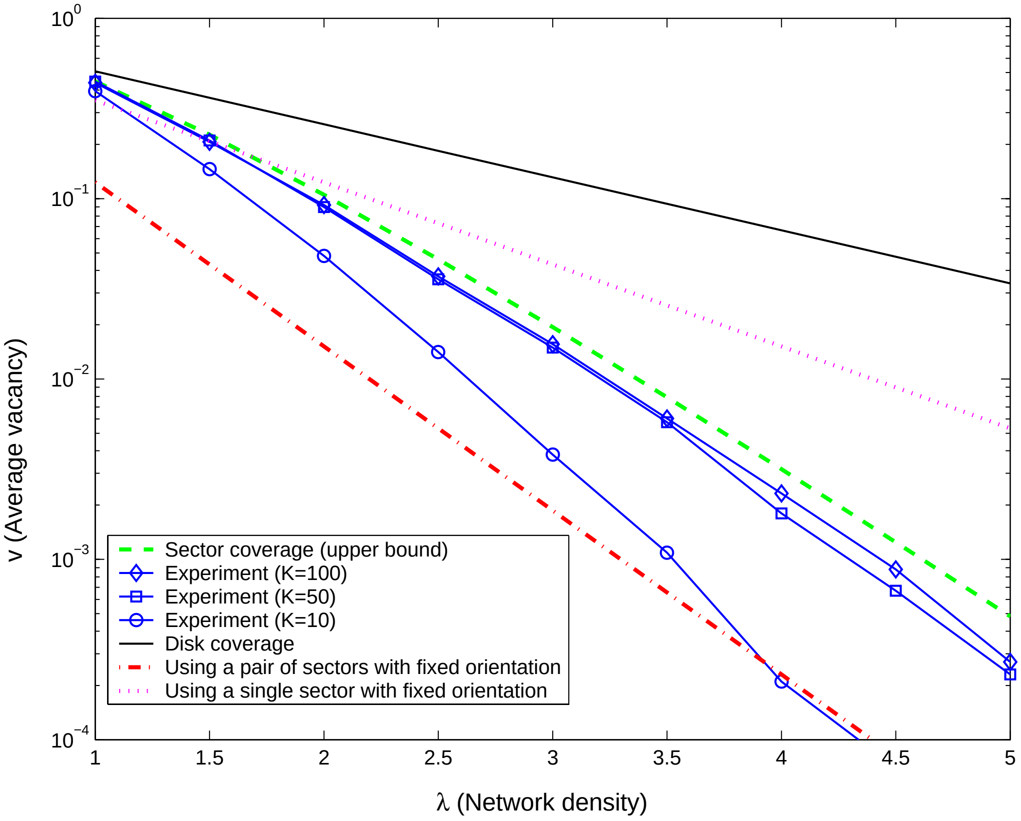


Figure 7.8: Experimental results on average vacancy for different ranging accuracy.

approximation we used in the sector coverage, so it is much easier to get a point to be covered. Thus, the average vacancy is smaller than the estimated value. The vacancy estimation by the disk coverage method is much larger than the experimental result, which means the disk model is not accurate in estimating the vacancy. Sector coverage provides better estimates, however it is more complex for a static sensor to determine whether the area around it is sector covered. In Fig. 7.8, we also compare the sector coverage with estimations based on sectors with *fixed* orientations. In this case, the average vacancy estimation will be $e^{-2\pi\lambda r^2/3}$ and $e^{-\pi\lambda r^2/3}$ when using two opposite $\{r, \frac{2\pi}{3}\}$ sectors and a single $\{r, \frac{2\pi}{3}\}$ sector, respectively. We see that checking for all orientations greatly improves the estimation accuracy.

7.3.2.3 Average Vacancy with Different α

Recall that we can represent the network resolution as $\alpha\sigma$, where σ is the distance estimation error. Fig. 7.9 shows the average vacancy under different α values. For $\alpha = 2$, more than 95% of the network area remain as vacancies even when the network density is high. This verifies our theoretical results in Section 7.2.3, which shows it is difficult to get localization error of 2σ by increasing beacon density. For $\alpha > 3$, the average vacancy decreases sharply as the beacon density increases. For larger α, the average vacancy decreases faster as the network density increases. However, the difference becomes smaller when α is large. The curve for $\alpha = 4$ is very close to the one for $\alpha = 5$, which means achieving $\alpha = 4$ requires nearly the same density for $\alpha = 5$. This result agrees with our theoretical analysis in Section 7.2.3.

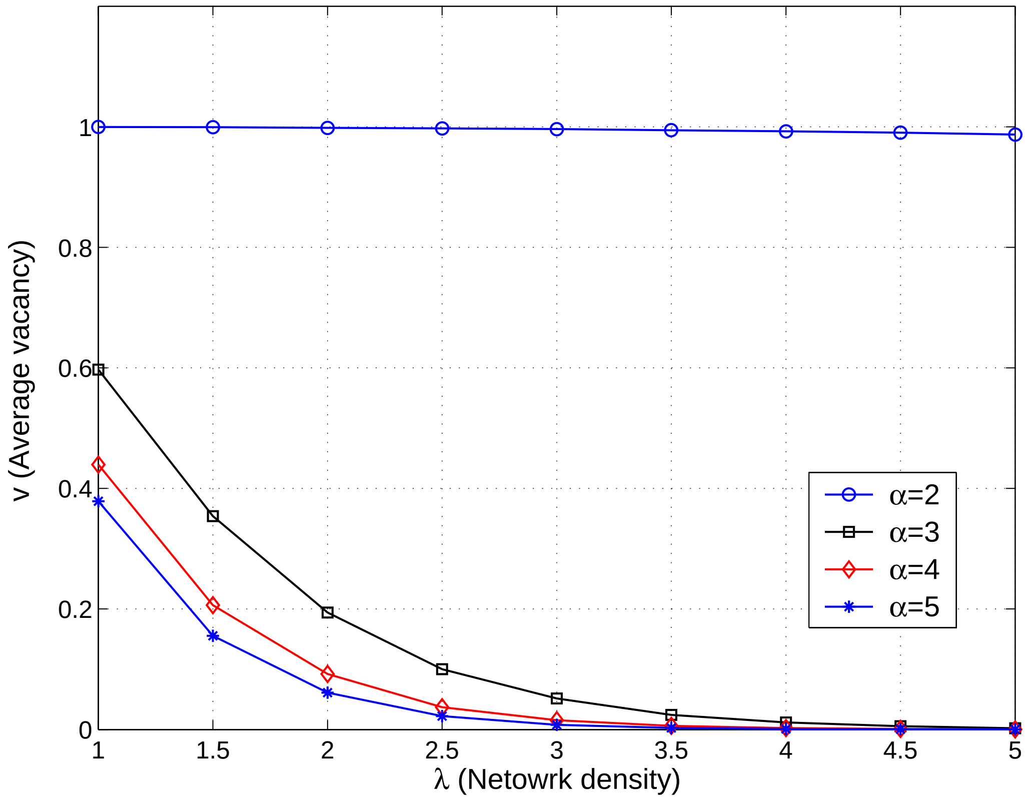


Figure 7.9: Experimental results on average vacancy for different network resolution ($K = 100$).

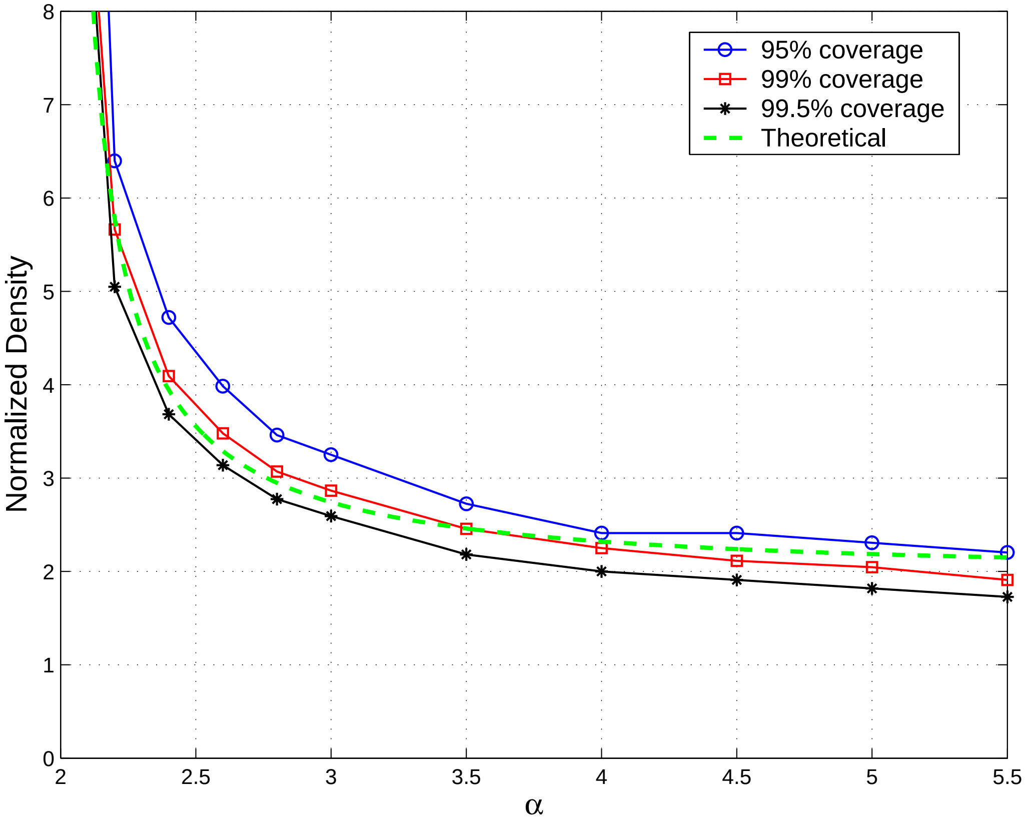


Figure 7.10: Experimental results of necessary network density with different average vacancy requirements ($K = 100$). Normalized density is the ratio of density compared to the detection coverage density.

7.3.2.4 Network Resolution

Fig. 7.10 shows the simulated relationship between the network density and network resolution. In the numerical experiment, we find the smallest network density which can guarantee that at least a given coverage ratio, say 99%, of the whole field can get a resolution smaller than $\alpha\sigma$. In Section 7.2.3, we derive this relationship based on the disk coverage model. As we know that the disk coverage model may over-estimate the network density by about 2-2.5 times, we need to verify whether this overestimation will change the trend shown in Fig. 7.5. Since Fig. 7.7 shows $\lambda_{disk}/\lambda_{sector}$ is about 2 when average vacancy is around 1%, we divide the density derived through disk coverage (as in Fig. 7.5) by two and set this curve as our theoretical value. Comparing the experimental results and the theoretical value, we see that the relationship derived through disk coverage is quite accurate, except the density is halved due to the difference between λ_{disk} and λ_{sector}. The numerical results show that the curve in Fig. 7.5 truly reflects the relationship between the network density and the network resolution and a network resolution of 3σ to 4σ is still a good trade-off as we expected. We also see that the normalized density decreases as the coverage ratio becomes higher for a given α. This agrees with the normalized density ($\lambda_{sector}/\lambda_{detect}$) curve in Fig. 7.7, where the normalized density decreases with the average vacancy.

7.3.3 Sleep-wake algorithms for sector coverage

In high density wireless sensor networks, not all static sensors need to be a beacon. Static sensors not working as beacons can turn off the beacon function to save energy. This is similar to the sleep-wake scheduling for wireless sensor networks [47]. To ensure the localization accuracy, the "waking" beacons should provide localization coverage over the whole field. In other words, a static sensor can only go to "sleep" when there is no coverage hole in its coverage area when it turns off the beacon function. Thus, we often need a distributed coverage algorithm to check whether there are coverage holes in the field when the position of the beacons are known. Similar to the algorithms in [47], we assume that nodes periodically wake up and execute this coverage algorithm.

For the disk coverage model, a sensor can determine whether its coverage region is k-covered by checking the intersection points of the sensing boundary of its neighbors [68]. This gives an algorithm of complexity $O(n^3)$, where n is the number of waking neighbors within range of $2r$. However, in sector coverage, a sensor needs to check whether there are sector voids around it, which requires more computation.

As in the classical coverage theory for arbitrary shapes, we define a point set $C \subseteq \mathbb{R}^2$ as the sensing region of one beacon [67]. The *Minkowski sum* of a point $\{x\}$ and C is defined as $\{x\} + C \equiv \{x + y : y \in C\}$, which is a translation of C by $\{x\}$.

For a field with beacons deployed at points $s_1, s_2, \ldots, s_n$, a point m is said to be not covered when: $m \notin \{s_i\} + C, \forall i$. Conversely, we have $s_i \notin \{m\} + C^*, \forall i$, where $C^* = \{-y : y \in C\}$. For example, we have $C = \{x \in \mathbb{R}^2 : |x| < r\}$ for the disk model. Then, a point m is said to be not covered when it is not in any circle with radius r centered at a beacon. Also, there will not be any beacon in the circle of radius r centered at the uncovered point m.

By Theorem 10, the network resolution can be guaranteed when there are no sector voids of $\{r, \frac{2\pi}{3}\}$ in the network. Consider a particular point m in the network which has a sector void of $\{r, \frac{2\pi}{3}\}$ with orientation of β around it, see Fig. 7.11. Accordingly, we have the sensing regions C^* as sectors with orientation of $\beta + \pi$ and point m will lie in the uncovered area in this case, see Fig. 7.11. Therefore, if the field can be fully covered by $\{r, \frac{2\pi}{3}\}$ sectors with orientation of $\beta + \pi$, then there will be no sector void with orientation of β in the field. Since a point is sector covered only

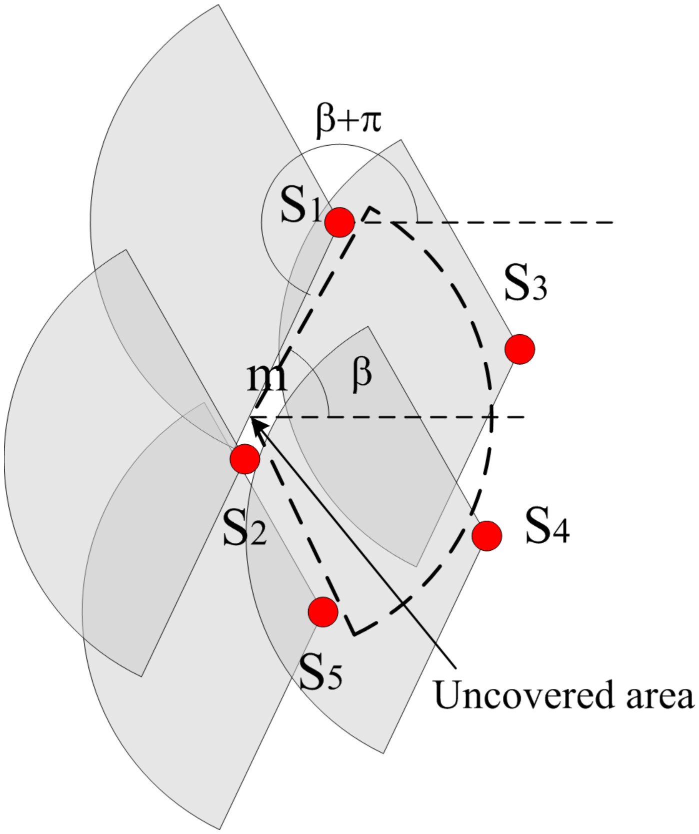


Figure 7.11: Sector coverage for a given orientation of β.

if it is covered for all orientations, we need to exhaustively check for all $\beta \in [0, 2\pi]$, which is practically impossible.

There are several ways to reduce the computational complexity of sector coverage algorithms.

First, we can only check for discrete orientations of β. Instead of using a sector of $\{r, \frac{2\pi}{3}\}$ as the sensing region, we can use a sector of $\{r, \frac{2\pi}{3} - \delta\}$, where δ is some small value. Then, we can increase β from 0 to 2π with step size of δ. If there is a sector void of $\{r, \frac{2\pi}{3}\}$ around point m for a certain orientation, there always exist a certain step that the $\{r, \frac{2\pi}{3} - \delta\}$ sector is fully included in it when the step size is δ, as shown in Fig. 7.12. Thus, we can definitely detect the void using step size of δ and sensing region of a $\{r, \frac{2\pi}{3} - \delta\}$ sector. However, this scheme may mistake a point as an uncovered point when there are only voids of $\{r, \frac{2\pi}{3} - \delta\}$ sectors around it. The network resolution at the point will actually be better than the desired value of 4σ, but it can still be taken as uncovered.

Using Eq. (7.5), the network resolution at a point is $u = \alpha\sigma$, where $\alpha = -\frac{2}{\cos\theta}$. For a point with a void of $\{r, \frac{2\pi}{3} - \delta\}$ sector, we can easily get $\theta = \frac{2\pi}{3} + \frac{\delta}{2}$. Thus, the network resolution is $u = -\frac{2\sigma}{\cos(\frac{2\pi}{3} + \frac{\delta}{2})}$. When δ is small, this can be approximated by Taylor expansion as: $u = (4 - 2\sqrt{3}\delta)\sigma$. For example, when $\delta = 0.01\pi$ (check for 200 different β values), the lower bound of network resolution around an uncovered point detected by the approximate scheme will be 3.89σ, which is close to 4σ. Also, for all points with network resolution larger than 4σ, our scheme can always detect it. This shows our discrete scheme is a good approximation to the exhaustive checking scheme. By choosing the step size δ, we can control the approximation ratio as required by the application.

For a fixed orientation, the sector coverage can be checked using a similar method as in the disk coverage [68]. If all intersection points of sectors of different beacons can be covered, then the field is fully covered. For the sector-shaped sensing region with the same orientation, the border of the sensing region of two beacons can intersect at most on 4 different points[3]. Calculating the intersection points for two beacons can be done in constant time. When there are n beacons within distance of $2r$, there

[3] If a line segment of a sector overlaps with border of other sectors, it is enough to check only the end points of the line segment.

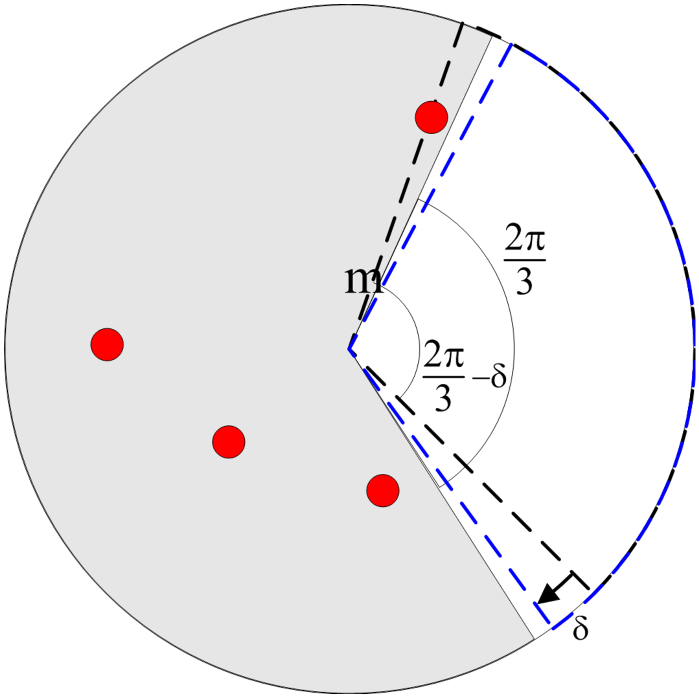


Figure 7.12: Increasing β by step size of δ. The void is two opposite sectors of $\{r, \frac{2\pi}{3}\}$, as the two white sector in the figure. The sensing region is the dotted sectors, which is $\{r, \frac{2\pi}{3} - \delta\}$.

are $O(n^2)$ intersection points to be checked. For each intersection point, we need to check whether it is within the sensing region of the other $n-1$ beacons. This gives computation complexity of $O(n^3)$, which is similar to the complexity of disk coverage algorithms. However, we need to repeat this algorithm $\frac{2\pi}{\delta}$ times. The overall computation complexity for our approximate sector coverage algorithm is $O(\frac{n^3}{\delta})$.

We can use the disk coverage model to exclude certain scenarios before using the sector coverage algorithm. As we have shown in Section 7.3.1, a point can only be sector covered when it is disk covered by more than 2 beacons. So, if a beacon's sensing disk is not 2-covered by other beacons, it cannot go to sleep. Furthermore, if there are sector voids of $\{r, \frac{2\pi}{3}\}$, there would be voids of disks with radius r_d. If shutting down a beacon s_i will generate a new sector void, the center o of an inscribed circle of the sector containing s_i will be an uncovered point when the sensing region are disks with radius r_d. Any point in the sector of $\{r, \frac{2\pi}{3}\}$ will be at most $\sqrt{r_d^2 + (r - r_d/\sqrt{3})^2} \approx 0.867r$ away from the point o. Then, if there are no voids of disk coverage of range r_d within distance of $0.867r$ to the beacon s_i, we can safely shut down beacon s_i without generating new sector voids.

In summary, we list the steps for a static sensor s_i to check whether it can go to sleep:

1. First, the sensor needs to know the position of all waking beacons within $2r$ distance from it.

2. If the disk with radius $0.867r$ around it can be fully covered by disks of radius r_d centered at other waking beacons, it can go to sleep and the algorithm terminates.

3. If the disk with radius r around it cannot be fully 2-covered by disks of radius r centered at other waking beacons, it cannot go to sleep and the algorithm terminates.

4. For each β from 0 to 2π with step δ, check if the disk with radius r around it can be fully covered by sectors of $\{r, \frac{2\pi}{3} - \delta\}$ with orientation of $\beta + \pi$ centered at other waking beacons. If there is any such void, it cannot go to sleep.

Compared to disk coverage algorithms, the sector coverage algorithm has computation complexity of $O(\frac{n^3}{\delta})$ rather than $O(n^3)$. So, sector coverage requires more computation when δ is small. The communication overhead of sector coverage is the same as the disk coverage. Our sector coverage algorithm only needs to know the information of neighboring sensors within range of $2r$ (their location and sleep-wake state). Such information is also required by disk coverage algorithms [68]. So, the information exchange protocols for disk coverage can also be used in our sector coverage algorithm. Note that in this algorithm we check for voids of single sector to guarantee network resolution. We can also use a similar algorithm to check whether there are two opposite sector voids of $\{r, \frac{2\pi}{3}\}$ in the network. Such algorithm can use fewer beacons to cover the network, while it cannot provide guarantees on the network resolution.

7.4 Chapter Summary

In this chapter, we studied the localization problem in mobile sensor network. We showed that conventional disk coverage model is insufficient for localization problems as it overestimates the beacon density by nearly 2 times. We proposed a new estimation method called sector coverage in this chapter. This new method is more complex than the simple disk coverage model. However, it can provide much better estimation on the coverage status for localization applications.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In this thesis, we studied three aspects of hybrid mobile sensor networks: the network lifetime optimization problem, the network coverage problem and the localization problem.

The main results and contributions of this thesis are summarized as follows:

1. **Network lifetime problem in hybrid mobile sensor networks**

   - We proposed a new way for resource redistribution in wireless sensor networks, which uses resource rich mobile nodes to help simple static sensors. We proved that one energy rich mobile node can improve the lifetime of a large and dense network by 4 times in the ideal case.

   - We derived the upper bound on network lifetime for a single mobile relay and multiple mobile relays. We then construct joint mobility and routing algorithms to show that this bound is asymptotically achievable in large and dense networks.

   - We studied the performance of mobile relay in random and finite network by formulating it as a linear programming problem. We compared the performance of mobile relay to various other static and mobile approaches and demonstrated the advantages of mobile relay approach.

2. **Coverage problem in hybrid mobile sensor networks**

   - We showed that *w.h.p.* (with high probability) sensor networks of all mobile sensors can use a sensor density of $\frac{\pi k}{2}$ and a maximum moving distance of $O(\frac{1}{\sqrt{k}} \log^{3/4}(kL))$ to provide k-coverage over a network with an area of L.

   - We proposed a hybrid network structure which uses a static sensor density of $\lambda = 2\pi k$ and mobile sensor density of $\frac{\lambda}{\sqrt{2\pi k}}$ to provide k-coverage over the field. The maximum moving distance for mobile sensors is $O(\log^{3/4} L)$ *w.h.p.* in our scheme.

   - We described a distributed algorithm to find the movement schedule for mobile sensors. Mobile sensors only need to have knowledge of neighbors within distance of $O(\log^{3/4} L)$ in the algorithm. The algorithm has time complexity of $O(L^2)$ and uses $O(L^3 \log^{3/2} L)$ message exchanges.

   - We observed that there is a trade-off between mobile sensor density and static sensor density. With higher static sensor densities, the mobile density can be reduced exponentially while the moving distance for mobiles still scales with the network size as $O(\log^{3/4} L)$.

   - We showed that the mobility algorithm is simple enough that it can be implemented in popular sensor platforms. We demonstrated an implementation on the Cricket Motes [56], integrated with the Boe-Bot Robot [80]. The algorithm is robust to packet loss, which often happens in real world.

3. **Localization problem in mobile sensor networks**

   - We derived the sufficient condition on sensor distribution which guarantees a lower bound of localization resolution over the field. We then showed that the conventional disk coverage model in inefficient when used for estimation localization coverage as it overestimates the necessary sensor density for localization coverage.

   - We proposed a new coverage estimation method which is called sector coverage for localization applications. This method provides more accurate estimations on the beacon density which is 2 times less than in the disk coverage model.

   - We also provided a sleep-wake algorithm for sector coverage which can guarantee certain localization accuracy. We showed that this sleep-wake algorithm has a similar computation complexity as conventional algorithms in disk coverage model.

In conclusion, the results in this thesis showed that hybrid mobile sensor network is a promising way to solve the network resource allocation problem and network deployment problem in wireless sensor networks. We demonstrated the advantages of hybrid mobile sensor network structure through the network lifetime optimization problem and network coverage problem. It was shown that only a small number of mobile sensors can greatly improve the network performance in these two cases.

The improvements in network performance in hybrid networks are due to the dynamical redistribution of network resources carried by mobile sensors. In the network lifetime optimization problem, the extra energy carried by the mobile relay is dynamically relocated to bottleneck static sensors so that the network lifetime can be extended. In the coverage problem, the sensing abilities of the mobile sensors are relocated to the vacancies which are the bottleneck of coverage so that the network coverage can be improved.

We showed that the mobile sensors only need to move for a limited distance in our hybrid network structures. In the network lifetime problem, the mobile relay only needs to move within 2 hop distance of the sink. In the coverage problem, the maximum moving distance of mobiles increases slowly as $O(\log^{3/4} L)$ with the network size L. Therefore, mobiles in our networks can be made simple and cheap and the hardware cost of the hybrid network can be reduced.

We observed that conventional disk coverage model is insufficient in localization problems for mobile sensor networks. This observation shows that we may need to reexamine the underlying assumptions of static sensor networks when the sensors can move. In many cases, these assumption may no longer hold and new models are needed for hybrid mobile sensor networks.

8.2 Future Directions

The research in this thesis revealed the potential of hybrid mobile sensor networks in network lifetime problems and network coverage problem. However, there are still many performance aspects in wireless sensor network can benefit from mobility.

First, mobility can potentially improve the network capacity of wireless sensor network. As sensor network normally contains only a single sink. The wireless channel around the sink will be extremely congested as all data packets need to be delivered to the sink. When all sensors around the sink are trying to send packets to the sink, they may interfere with each other and the network capacity will be limited by the wireless channel around the sink [85]. It has been shown that mobility can resolve the interference problem in wireless channel and increase network capacity by several magnitudes [15]. However, the approach in [15] incurs large packet delay. The remaining unsolved problem is whether there is a mobility scheme which can greatly increase the network throughput of the sink while ensuring real time delivery? Is there a tradeoff between packet delay and network capacity?

Second, mobility can also improve the robustness of wireless sensor networks. Sensors may die due to depletion of energy or due to physical damage. If critical sensors die, the network may be disconnected or lose coverage. The robustness of wireless sensor networks in this case can be quantified by measuring the damage that the sensor network can sustain. The damage can be measured as the portion of dead sensors or area of the region in which all sensors die. Mobile sensors can dynamically replace failure sensors which stop functioning. It is important to understand how many mobile sensors are required to be deployed in the field to significantly increase chance that the sensor network can recover from a devastating damage.

Third, the mobile sensors studied in this thesis are not much more powerful than static sensors except they can move. Mobile sensors can have superior abilities such as directional antennas, high-speed/long-range communication abilities and high quality sensors. Compared to the simple mobile sensors studied in this paper, these powerful mobile sensors may bring new benefits in network performance. However, little research has been done in the mobility algorithm and performance boundaries of these powerful mobile sensors.

Moreover, this thesis mainly focuses on deriving the network performance bounds in hybrid mobile sensors. The algorithms in this thesis are often based on simplified models. There are still many realistic issues which need to be studied before mobile sensors can be deployed in the real world. These issues include: routing algorithms for mobile sensor networks, mobility algorithms which can work when the network contains barriers and fault tolerant algorithms for mobile sensors.

Bibliography

[1] A. Cerpa, J. Elson, M. Hamilton, and J. Zhao, “Habitat monitoring: Application driver for wireless communications technology,” in *Proceedings of ACM SIGCOMM*, 2001.

[2] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, “An analysis of a large scale habitat monitoring application,” in *Proceedings of ACM SenSys*, 2004.

[3] R. Brooks, P. Ramanathan, and R. Sayeed, “Distributed target classification and tracking in sensor networks,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1163 – 1171, 2003.

[4] M. Li and Y. Liu, “Underground structure monitoring with wireless sensor networks,” in *Proceedings of IPSN*, Apr 2007.

[5] “MPR-MIB users manual,” Crossbow Technology, 2007.

[6] H. Zhang and J. Hou, “On deriving the upper bound of α-lifetime for large sensor networks,” in *Proceedings of ACM MobiHoc*, 2004.

[7] S. Kumar, T. Lai, and J. Balogh, “On k-coverage in a mostly sleeping sensor network,” in *Proceedings of ACM Mobicom*, 2004, pp. 144–158.

[8] P.-J. Wan and C.-W. Yi, “Coverage by randomly deployed wireless sensor net-

works," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2658–2669, 2006.

[9] J.Luo and J. P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *Proceedings of IEEE INFOCOM*, Mar 2005.

[10] W. Wang, V. Srinivasan, and K. C. Chua, "Using mobile relays to prolong the lifetime of wireless sensor networks," in *Proceedings of ACM MobiCom*, Aug 2005.

[11] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," in *Proceedings of IEEE INFOCOM*, 2004.

[12] G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in *Proceedings of IEEE INFOCOM*, Mar 2005.

[13] M. Zhang, X. Du, and K. Nygard, "Improving coverage performance in sensor networks by using mobile sensors," in *Proceedings of IEEE MILCOM*, 2005.

[14] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks," in *Proceedings of ACM MobiHoc*, 2005.

[15] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, 2002.

[16] Z. Wang, Z. Song, P. Chen, A. Arora, D. Stormont, and Y. Chen, "MASmote-a mobility node for MAS-net (mobile actuator sensor networks)," in *Proceedings of IEEE International Conference on Robotics and Biomimetics*, 2004, pp. 22–25.

[17] M. Batalin, G. Sukhatme, and M. Hattig, "Mobile robot navigation using a sensor network," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[18] M. McMickell, B. Goodwine, and L. Montestruque, "MICAbot: a robotic platform for large-scale distributed robotics," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[19] R. Suzuki, K. Makimura, H. Saito, and Y. Tobe, "Prototype of a sensor network with moving nodes," in *Proceedings of International Workshop on Networked Sensing Systems*, 2004.

[20] A. Kansal, A. Somasundara, D. Jea, M. Srivastava, and D. Estrin, "Intelligent fluid infrastructure for embedded networks," in *Proceedings of ACM MobiSys* , 2004.

[21] G. Sibley, M. Rahimi, and G. Sukhatme, "Robomote: A tiny mobile robot platform for large-scale ad-hoc senosr networks," in *Proceedings of International Conference on Robotics and Automation*, 2002.

[22] J. T. Feddema, R. H. Byrne, J. J. Harrington, D. M. Kilman, C. L. Lewis, R. D. Robinett, B. P. V. Leeuwen, and J. G. Young, "Advanced mobile networking, sensing, and controls," Technical Report SAND2005-1661, Sandia National Laboratories, 2005.

[23] W. Wang, V. Srinivasan, and K. C. Chua, "Trade-offs between mobility and density for coverage in wireless sensor networks," in *Proceedings of ACM MobiCom*, 2007.

[24] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling a three-tier architecture for sparse sensor networks," in *Proceedings of the IEEE Workshop on Sensor Network Protocols and Applications (SNPA)*, 2003.

[25] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proceedings of IEEE INFOCOM*, 2003.

[26] S. Chellappan, X. Bai, B. Ma, and D. Xuan, "Sensor Networks Deployment Using Flip-Based Sensors," in *Proceedings of IEEE MASS*, 2005.

[27] J. Wu and S. Yang, "Smart: A scan-based movement-assisted sensor deployment method in wireless sensor networks," in *Proceedings of IEEE INFOCOM*, 2005.

[28] G. Wang, G. Cao, and T. L. Porta, "A bidding protocol for sensor deployment," in *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, 2003.

[29] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks," in *Proceedings of IPSN*, April 2003.

[30] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proceedings of ACM MobiHoc*, 2004, pp. 187–198.

[31] S. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy-efficient schemes for wireless sensor networks with multiple mobile base stations," in *Proceedings of IEEE GLOBECOM*, Dec 2003.

[32] Z. M. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, “Exploiting sink mobility for maximizing sensor networks lifetime,” in *Proceedings of the* 38*th Hawaii International Conference on System Sciences (HICSS)*, 2005.

[33] J. Luo, J. Panchard, M. Piorkowski, M. Grossglauser, and J.-P. Hubaux, “Mobiroute: Routing towards a mobile sink for improving lifetime in sensor networks,” in *Proceedings of International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2006.

[34] J. H. Chang and L. Tassiulas, “Energy conserving routing in wireless ad-hoc networks,” in *Proceedings of IEEE INFOCOM* , Mar 2000.

[35] A. Shankar and Z. Liu, “Maximum lifetime routing in wireless ad-hoc networks,” in *Proceedings of IEEE INFOCOM*, Mar 2004.

[36] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, “Prolonging sensor network lifetime with energy provisioning and relay node placement,” in *Proceedings of IEEE SECON*, Sep 2005.

[37] J. Chou, D. Petrovic, and K. Ramchandran, “A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks,” in *Proceedings of IEEE INFOCOM*, Mar 2003.

[38] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS)*, 2000.

[39] O. Younis and S. Fahmy, “Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach,” in *Proceedings of IEEE INFOCOM*, Mar

2004.

[40] N. Li, J. Hou, and J. Sha, "Design and analysis of an MST based topology control algorithm," in *Proceedings of IEEE INFOCOM*, 2003.

[41] J. Pan, Y. Hou, L. Cai, Y. Shi, and S. Shen, "Topology control for wireless sensor networks," in *Proceedings of ACM MobiCom*, 2003.

[42] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.Srivastava, "Coverage problems in wireless ad-hoc sensor network," in *Proceedings of IEEE INFOCOM*, 2001.

[43] C. Huang and Y. Tseng, "The coverage problem in a wireless sensor network," in *Proceedings of ACM WSNA*, Sep 2003, pp. 115–121.

[44] S. Megerian, G. Koushanfar, F.and Qu, and M. Potkonjak, "Exposure in wireless sensor networks," in *Proceedings of ACM MobiCom*, 2001.

[45] S. Kumar, T. H. Lai, , and A. Arora, "Barrier coverage with wireless sensors," in *Proceedings of ACM MobiCom*, Aug 2005.

[46] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: coverage, connectivity and diameter," in *Proceedings of IEEE INFOCOM*, Apr 2003.

[47] D.Tian and N.D.Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of ACM WSNA*, 2002.

[48] G. Wang, G. Cao, and T. L. Porta, "Proxy-based sensor deployment for mobile sensor networks," in *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2004.

[49] S. Chellappan, W. Gu, X. Bai, and D. Xuan, "Deploying wireless sensor networks under limited mobility constraints," *IEEE Transactions on Mobile Computing*, vol. 6, no. 10, pp. 1142–1157, 2007.

[50] T. Leighton and P. W. Shor, "Tight bounds for minimax grid matching, with applications to the average case analysis of algorithms," *Combinatorica*, vol. 9, no. 2, pp. 161–187, 1989.

[51] Z. Lotker and A. Navarra, "Unbalanced points and vertices problem," in *Proceedings of PerCom Workshops (FAWN)*, 2006, pp. 96–100.

[52] ——, "Managing random sensor networks by means of grid emulation," in *Proceedings of Networking (LNCS 3976)*, 2006, pp. 856–867.

[53] H. Zhang and J. Hou, "Maintaining coverage and connectivity in large sensor networks," in *International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless and Peer-to-Peer Networks*, 2004.

[54] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks," in *Proceedings of Annual ACM Symposium on Principles of Distributed Computing (PODC)*, Aug 2001, pp. 264–273.

[55] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of ACM MobiCom*, Aug 2003.

[56] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *Proceedings of ACM MobiCom*, Aug 2000.

[57] A. Savvides, A. Sachin, W. Garber, R. Moses, and M. Srivastava, "On the error characteristics of multihop node localization in ad-hoc sensor networks," in *Proceedings of IPSN*, Apr 2003.

[58] H. Wang, L. Yip, K. Yao, and D. Estrin, "Lower bounds of localization uncertainty in sensor networks," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2004.

[59] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, Aug 2001.

[60] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proceedings of ACM MobiCom*, Aug 2004.

[61] N. Bulusu, J. Heidemann, and D. Estrin, "Density adaptive algorithms for beacon placement in wireless sensor networks," in *Proceedings of IEEE ICDCS*, Apr 2001.

[62] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network," in *Proceedings of IPSN*, Apr 2003.

[63] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *Proceedings of ACM MobiHoc*, Jun 2003.

[64] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of IEEE INFOCOM*, Jun 2002.

[65] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Pro-*

cesses. 4th Ed. McGraw Hill, 2002.

[66] W. Wang, V. Srinivasan, and K. Chua, "Power control for distributed mac protocols in wireless ad hoc networks," to appear on IEEE trans. on Mobile Computing, 2008.

[67] P. Hall, *Introduction to the theory of coverage processes.* John Wiley & Sons, Inc, 1988.

[68] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of ACM Sensys*, 2003.

[69] R. Williams, *The Geometrical Foundation of Natural Structure: A Source Book of Design.* Dover Publications, 1979.

[70] E. Kratzel, *Lattice points.* Kluwer Academic Publishers, 1989.

[71] A. Goel, S. Rai, and B. Krishnamachari, "Sharp thresholds for monotone properties in random geometric graphs," in *Proceedings of the ACM symposium on Theory of Computing*, 2004, pp. 580–586.

[72] Web based demo for mobile sensors, http://cnds.ece.nus.edu.sg/mobile/mobile.html.

[73] S. Janson, T. Luczak, and A. Ruciński, *Random Graphs.* John Wiley & Sons, Inc, 2000.

[74] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

[75] M. Cardei, M. Thai, Y. Li, and W. Wu, “Energy-Efficient Target Coverage in Wireless Sensor Networks,” in *Proceedings of IEEE INFOCOM*, 2005.

[76] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows : theory, algorithms, and applications.* Prentice Hall, 1993.

[77] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms.* 2nd Ed. MIT Press and McGraw-Hill, 2001.

[78] A. Goldberg and R. Tarjan, “A new approach to the maximum-flow problem,” *Journal of the ACM*, vol. 35, no. 4, pp. 921–940, 1988.

[79] ——, “Finding minimum-cost circulations by successive approximation,” *Mathematics of Operations Research*, vol. 15, no. 3, pp. 430–466, 1990.

[80] Parallax, http://www.parallax.com/html_pages/robotics/boebot/boebot.asp.

[81] Video of mobile implementations, http://cnds.ece.nus.edu.sg/mobile/mobile4.mpg.

[82] C. Liu, K. Wu, , and T. He, “Sensor localization with ring overlapping based on comparison of received signal strength indicator,” in *Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, Oct 2004.

[83] D. Wells, *The Penguin Dictionary of Curious and Interesting Geometry.* London: Penguin, 1991.

[84] J. F. Kenney and E. S. Keeping, *Mathematics of Statistics.* Princeton, 1962.

[85] D. Marco, E. Duarte-Melo, M. Liu, and D. Neuhoff, “On the many-to-one trans-

port capacity of a dense wireless sensor network and the compressibility of its data," in *Procedings of IPSN*, 2003, pp. 1–16.

List of Publications

Journals

1. Wei Wang, Vikram Srinivasan and Kee-Chaing Chua, “Coverage in Hybrid Mobile Sensor Networks”, to appear in *IEEE transactions on Mobile Computing*, Nov 2008 (Extended version of Mobicom’07 paper)

2. Wei Wang, Vikram Srinivasan and Kee-Chaing Chua, “Power Control for Distributed MAC Protocols in Wireless Ad Hoc Networks”, to appear in *IEEE transactions on Mobile Computing*, Aug 2008.

3. Wei Wang, Vikram Srinivasan, and Kee Chaing Chua, “Extending the Lifetime of Wireless Sensor Networks through Mobile Relays”, to appear in *IEEE/ACM Transactions on Networking*, 2008 (Extended version of Mobicom’05 paper)

4. Wei Wang, Vikram Srinivasan, Bang Wang and Kee Chaing Chua, “Coverage for Target Localization in Wireless Sensor Networks”, *IEEE Transactions on Wireless Communications*, Vol.7, no.2, pp. 667 - 676, Feb 2008 (Extended version of IPSN’06 paper)

5. Bang Wang, Kee Chaing Chua, Vikram Srinivasan, and Wei Wang, “Information Coverage in Randomly Deployed Wireless Sensor Networks”, *IEEE Transactions on Wireless Communications*, Vol.6, no.8, pp. 2994 - 3004, Aug 2007

6. Bang Wang, Wei Wang, Vikram Srinivasan, Kee Chaing Chua, “Information coverage for wireless sensor networks”, *IEEE Communications Letters*, Vol.9, no.11, pp. 967 - 969, 2005

Conferences

1. Wei Wang, Vikram Srinivasan and Kee-Chaing Chua, “Trade-offs Between Mobility and Density for Coverage in Wireless Sensor Networks”, in *proceedings of ACM MobiCom*, Sep 2007, pp. 39 - 50 (Both regular paper and demo)

2. Wei Wang, Vikram Srinivasan and Mehul Motani, “Adaptive Contact Probing Mechanisms for Delay Tolerant Applications”, in *proceedings of ACM MobiCom*, Sep 2007, pp. 230 - 241

3. Wei Wang, Vikram Srinivasan, Kee Chaing Chua and Bang Wang, “Energy-efficient Coverage for Target Detection in Wireless Sensor Networks”, in *proceedings of IPSN*, Apr 2007 pp. 313 - 322 (Poster presentation)

4. Bang Wang, Kee Chaing Chua, Vikram Srinivasan, and Wei Wang, “Scheduling Sensor Activity for Point Information Coverage in Wireless Sensor Networks”, in *proceedings of WiOpt*, Apr 2006

5. Wei Wang, Vikram Srinivasan, Bang Wang and Kee Chaing Chua, “Coverage for Target Localization in Wireless Sensor Networks”, in *proceedings of IPSN*, Apr 2006, pp. 118 - 125

6. Bang Wang, Kee Chaing Chua, Vikram Srinivasan, and Wei Wang, “Sensor Density for Complete Information Coverage in Wireless Sensor Networks”, in *proceedings of EWSN*, Feb 2006

7. Bang Wang, Kee Chaing Chua, Wei Wang, Vikram Srinivasan, “Worst and Best Information Exposure Paths in Wireless Sensor Networks”, in *proceedings of MSN, LNCS 3794*, Dec 2005, pp. 52 - 62

8. Bang Wang, Kee Chaing Chua, Vikram Srinivasan, Wei Wang, “Localized Recursive Estimation in Wireless Sensor Networks”, in *proceedings of MSN, LNCS 3794*, Dec 2005, pp. 390 - 399

9. Wei Wang, Vikram Srinivasan, and Kee Chaing Chua, “Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks”, in *proceedings of ACM MobiCom*, Aug 2005, pp. 270 - 283