ALGORITHMS FOR SINK MOBILITY IN WIRELESS SENSOR NETWORKS TO IMPROVE NETWORK LIFETIME

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Metin Koç
August, 2008

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. İbrahim Körpeoğlu (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Ali Aydın Selçuk

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Adnan Yazıcı

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

ALGORITHMS FOR SINK MOBILITY IN WIRELESS SENSOR NETWORKS TO IMPROVE NETWORK LIFETIME

Metin Koç
M.S. in Computer Engineering
Supervisor: Assist. Prof. Dr. İbrahim Körpeoğlu
August, 2008

A wireless sensor network (WSN) consists of hundreds or thousands of sensor nodes organized in an ad-hoc manner to achieve a predefined goal. Although WSNs have limitations in terms of memory and processor, the main constraint that makes WSNs different from traditional networks is the battery problem. Since sensor nodes are generally deployed to areas with harsh environmental conditions, replacing the exhausted batteries become practically impossible. This requires to use the energy very carefully in both node and network level. Different approaches are proposed in the literature for improving network lifetime, including data aggregation, energy efficient routing schemes and MAC protocols, etc. Main motivation for these approaches is to prolong the network lifetime without sacrificing service quality. Sink (data collection node) mobility is also one of the effective solutions in the literature for network lifetime improvement.

In this thesis, we focus on the controlled sink mobility and present a set of algorithms for different parts of the problem, like sink sites determination, and movement decision parameters. Moreover, a load balanced topology construction algorithm is given as another component of network lifetime improvement. Experiment results are presented which compare the performance of different components of the mobility scheme with other approaches in the literature, and the whole sink mobility scheme with random movement and static sink cases. As a result, it is observed that our algorithms perform better than random movement and static cases for different scenarios.

Keywords: Wireless Sensor Networks, Network Lifetime, Sink Mobility, Topology Construction.

ÖZET

TELSİZ ALGILAYICI AĞLARDA AĞ ÖMRÜNÜ GELİŞTİRMEK İÇİN ÇIKIŞ DÜĞÜMÜ YER DEĞİŞİMİ KONUSUNDA ALGORİTMALAR

Metin Koç
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. Dr. İbrahim Körpeoğlu
Ağustos, 2008

Telsiz algılayıcı ağları önceden belirlenmiş bir amacı gerçekleştirmek için tasarsız bir biçimde örgütlenen yüzlerce veya binlerce algılayıcı düğümden oluşur. Telsiz algılayıcı ağlarda bellek ve işlemcide sınırlamalar olsa da, onları geleneksel ağlardan ayıran en önemli kısıt pil problemidir. Algılayıcı düğümler zor çevresel koşulların olduğu alanlara yayıldığından, biten pilleri yenileriyle değiştirmek pratik olarak mümkün olmamaktadır. Bu durum her bir algılayıcı düğümünün enerjisinin hem düğüm hem de ağ seviyesinde dikkatlice kullanılmasını gerektirmektedir. Literatürde, ağ ömrünü geliştirmek için veri toplaşımı, enerji etkin yönlendirme düzenleri ve ortama erişim protokolleri gibi bir çok yaklaşım önerilmiştir. Bu yaklaşımların temel motivasyonu servis kalitesinden ödün vermeyerek ağ ömrünü geliştirmektir. Çıkış düğümü (veri toplanan düğüm) yer değişimi literatürde ağ ömrünü geliştirmek için sunulan etkin çözümlerden biridir.

Bu tezde, kontrol edilebilir çıkış düğümü yer değişimine odaklanıp, bu problemin çıkış düğümü yerleri belirleme, hareket kararı parametreleri gibi değişik kısımlarının çözümü için bir algoritma kümesi sunulmuştur. Ayrıca, ağ ömrü uzatma çözümünün farklı bir bileşeni olarak yük dengeli topoloji yapılandırması konusunda da bir algoritma verilmiştir. Yer değişimi düzeninin farklı bileşenlerini literatürdeki diğer ilgili yaklaşımlarla, bütün çıkış düğümü yer değişimi düzenini ise, rasgele yer değişimi ve hareketsiz çıkış düğümü durumlarıyla karşılaştıran deney sonuçları sunulmuştur. Sonuç olarak, algoritmalarımızın rasgele yer değişimi ve hareketsiz durumlara göre daha iyi sonuçlar verdiği görülmüştür.

Anahtar sözcükler: Telsiz Algılayıcı Ağlar, Ağ Zamanı, Çıkış Düğümü Yer Değişimi, Topoloji Kurulumu.

Acknowledgement

I would like to express my gratitude and sincere thanks to my supervisor, Dr. İbrahim Körpeoğlu. I am indebted to Dr. Körpeoğlu for his invaluable comments and continuous support, guidance, and friendship throughout my master study.

I would like to thank to Assist. Prof. Dr. Ali Aydın Selçuk and Prof. Adnan Yazıcı for evaluating this thesis and being on my thesis defense committee.

I also thank to The Scientific and Technological Research Council of Turkey (TÜBİTAK) for supporting this work with project EEEAG-104E028.

There are very special people around me during this period who continuously support and encourage me about completing this study. I would like to express my sincere thanks to Özgür for his invaluable friendship, my brother Levent for his continuous energy and motivation he gave to me, and my parents for their love, sacrifice, and trust which make me feel very lucky throughout my life.

Finally, I am grateful to Buket, my wife, for her endless love, patience, understanding and support that she gave at every moment of this journey. She did try her best to provide the most comfortable study environment for me and to solve every annoying detail of daily life issues before reaching to me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The emergence of tiny sensor nodes as a consequence of the advances in micro-electro-mechanical systems has introduced the *wireless sensor networks.* Sensor networks consist of hundreds, even thousands of sensor nodes which are deployed to an area of interest and construct a multi-hop network topology in order to achieve a goal. A sensor network has typically two different kinds of nodes: sink and sensor nodes. A sensor node is a low cost, low power device that is responsible from sensing and communicating. Sink node (base station)[1] is the node where data are collected and interpreted. Generally, in the literature, it is assumed that sink node has sufficient amount of energy which cannot be depleted during the network operation.

Each sensor node has mainly three basic units in order to achieve its task: sensing, processing, and communication [1]. A node can use different kind of sensors in its sensing unit for interacting with the medium and gathering data related to assigned task. After sensing, a node can process the data (applying various functions like *max*, *min*, *average*) using its processing unit and transmit the data to its parent. It can also receive and relay its children's packets destined to the sink node. Base station receives and processes all of these packets, and an application that can run on the computer should interact with it and enable the

---

[1] *Sink* and *base station* are used interchangeably throughout the thesis

end user to display and query the current and past information about the area of interest.

There are various types of sensor network applications. [1] categorizes these applications and gives typical examples for each category. Here we briefly summarize these applications.

- *Military Applications*: The rapid deployment, self-organization and fault tolerance characteristics of sensor networks make them appropriate for military purposes. Monitoring friendly forces, battlefield surveillance, reconnaissance of opposing forces, targeting, and nuclear and chemical attack detection are examples of military applications. They can be used in hostile environments where it is too dangerous for humans to operate.
- *Environmental Applications*: The most widely used sensor network application is environmental monitoring. Various types of sensors enable the nodes to sense the environment and perform given tasks continuously. This kind of application includes forest fire and flood detection, habitat monitoring, tracking the movement of targeted animals.
- *Health Applications*: Some of the health applications for sensor networks are integrated patient monitoring; diagnostics; drug administration in hospitals; monitoring the movements and internal processes of small animals; telemonitoring of human physiological data (heart rate, blood pressure detection, so on); and tracking and monitoring doctors and patients inside a hospital.
- *Home Applications*: Sensor nodes can be used for home automation to provide smart home environment in which all the appliances can interact with each other and be controlled remotely outside the home.
- *Commercial Applications*: This type includes managing and controlling inventory, detecting and tracking vehicles, and factory process control and automation.

Wireless sensor networks have special characteristics that differentiate them from ad hoc networks [1, 36]. The number of sensor nodes deployed to the area of interest is much higher (even order of thousands or more in some cases) than that of ad hoc networks. The data communication is generally many to one (each data packet is destined to sink in order to be processed and interpreted) whereas in ad-hoc networks each node can communicate to one another (point-to-point). Deployed sensor nodes can be inaccessible due to harsh environmental conditions in some of the applications of WSN, like forrest fire detection, and battlefield surveillance. In this case, cost of replacing the battery of a sensor node can be more expensive than deploying a new sensor node to the area. The other important difference is the limited computational and memory characteristics of sensor nodes. This property restricts the programmers that design algorithms for sensor networks in order not to exceed the limits of node memory and processing capabilities (a typical Mica2 mote [13] has 128 KB programming memory).

Beside all of these differences, the most important characteristics of WSNs is limited energy resources of sensor nodes. A typical sensor node has generally irreplaceable limited capacity battery attached to the programming interface board. Consuming less amount of energy is the most critical criterion when designing any sensor network related protocol. Since the energy is the most precious resource, and in most of the applications, replacing the batteries are very hard or impractical, utilizing each node's and total energy of the network becomes much more important for a given task.

Several schemes are proposed in the literature in order to minimize the total energy consumption in the network for improving the network lifetime: power adjusting when transmitting messages [9], developing energy efficient MAC or routing protocols [14, 30, 41, 42], minimizing the number of messages traveling in the network (since most of the energy is consumed when transmitting data packets), putting some sensor nodes into sleep mode and using only a necessary set of them for sensing and communication [50].

Making the sink mobile appears as another approach for improving the network lifetime for WSNs. In a WSN, each sensor node not only transmits its own packet to sink, but also relays the packets of its children, when *data aggregation*[2][3] is not used. Since most of the time a tree topology is constructed from up to bottom (from base station to leaf nodes), all packets of the network are delivered to the sink node via its first hop neighbors. As it can be seen from Fig. 1.1, snapshot of the energy map in the network after first node died, this situation causes these nodes to deplete their energy faster than the other nodes in the network. So, the main motivation behind the *sink mobility* is to change these nodes periodically (in each round) in order to delegate sink's neighbor role among the sensor nodes in a fairly manner for balancing the remaining energy level of the nodes and finally improving the network lifetime. A node that was sink's neighbor in the previous round should have smaller packet load in the next round so that on the average all nodes have nearly equal packet load (so equal remaining energy levels) at an arbitrary time.

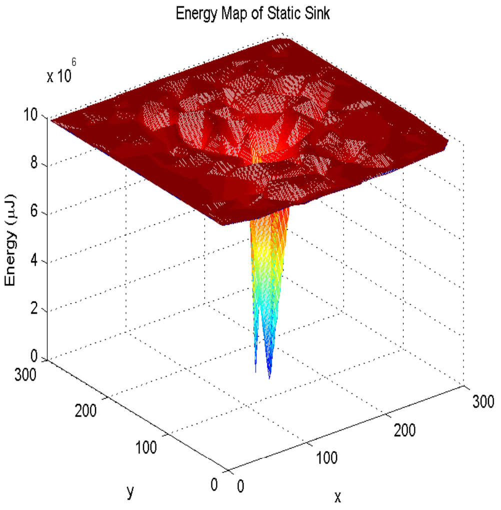


Figure 1.1: Energy map of a static sink after first node death

A sink mobility scheme has to address the issue of *when* (round duration, sojourn time) and *where* (migration point, sink site, anchor point) to move the sink node next questions and also it should explain which network parameters should be used in order to regulate this operation.

---

[2] Data Aggregation is the process of expressing data in summary form.

[3] Each node processes its child's packet, applies a function to both of its own packet and that of its child (min, max, avg, etc.) and finally transmits one and only one packet to its parent in this case.

1.1 Motivation

Sink mobility has been drawn the attention of the researchers for a few years. Several papers have been published about different aspects of the topic. In order to solve *when* and *where* questions of the mobility scheme, to the best of our knowledge, all of them either uses Linear Programming (LP) or Integer Linear Programming (ILP)/ Mixed Integer Linear Programming (MILP). Although this situation provides optimal solution to the given problem formulation and assumptions, they also bring scalability problem together. Except [3], none of proposed solutions exceeds a hundred number of nodes (whereas [3] uses at most 600 nodes) in their simulation scenarios, which can be treated ”insufficient” when considering the typical sensor network applications like environmental monitoring and battlefield surveillance where hundreds or even thousands of nodes are deployed to the area of interest. When the number of nodes increases, time needed for the solution of formulation (which also equals the time elapsed while sink is deciding where to move) increases exponentially. Since sensor nodes cannot transmit or receive any data packet during the decision and movement phase, a huge-size buffer requirement problem arises which also increases delay. Due to limited computational and memory characteristics of sensor nodes, scalability problem makes those approaches impractical for the applications that require very large number of nodes.

Because of the buffer and delay problem described above, the sink node should not move to points far away from the location that it currently stays, since moving to a distant location also takes time. However, depending on the application, the area of interest that the nodes are deployed can be very large (order of kilometers). On the other hand, some regions of the area (hill, boggy) should have difficulties that prevent the sink node to move a point which lies on this kind of areas even they are closer to the current staying point. These conditions restrict the points that the sink node can move when it wants. To the best of our knowledge, a dynamic sink site determination algorithm has not been proposed before, in which such network parameters are taken into consideration. In previous works, the area of interest is assumed to have square properties, and the corners and some predefined points in this square are defined off-line as the possible migration points independent from any topological or deployment information. However, as mentioned above, these predefined points should have some restrictions to move (no alternative points are given in those works). We have proposed two different sink site determination algorithms which consider number of nodes and their positions (and some probable special conditions) when deciding possible migration points.

1.2 Contributions

In this thesis, we propose a set of algorithms for different aspects of the sink mobility problem in wireless sensor networks. First, we propose two different sink site determination algorithms. To the best of our knowledge, these are the first proposed algorithms in which possible sink sites are not given as the predefined points of the area, instead, they will be determined according to the distribution of nodes in the area and their coordinates or neighborhood information. Secondly, a *cost function* is introduced and an algorithm is given when the buffer size limitation of sensor nodes prevents the sink to move to any candidate site. Additionally, an energy efficient topology construction algorithm and local tree topology construction algorithms are presented which are important for improving network lifetime. These issues were not addressed all together in most of the previous studies.

1.3 Thesis Structure

In Chapter 2, some preliminary information about wireless sensor network is given. The chapter also gives a brief explanation of the previous related work about different general approaches for improving network lifetime (other than sink movement), mobility schemes in general, and specifically sink mobility issue in wireless sensor networks. Pros and cons of each work has been discussed in detail. Chapter 3 presents the heuristic algorithms about the sink mobility issues with sink site determination, efficient topology construction and local tree update mechanism. In Chapter 4, details about the simulation environment and parameters are given and results of the experiments are presented. Finally, Chapter 5 concludes the thesis and gives future research directions.

Chapter 2

Background and Related Work

Sink mobility is one of the approaches, among many others, that have been used for prolonging network lifetime. It has been extensively discussed over the last few years by treating one or more different aspects of the network, such as routing, sink location, sojourn time of the mobile sink, etc. In this chapter, important articles that are closely related to our work in this thesis are discussed. However, it is first preceded by a section that classifies the works related to network lifetime improvement and cites some of the important papers in each group.

2.1 Network Lifetime Improvement Approaches: A Summary

Since energy is the most precious resource of a sensor network, it should be carefully taken into consideration in any algorithm or approach related to sensor network operations. This situation causes researchers to deal with network lifetime improvement in different aspects of networking. Before going into more detail about the works related to our study, it would be better to give a brief summarization about the papers that use different mechanisms to improve network lifetime. These works generally lie in physical layer (power control), data link layer (MAC protocols), network layer (routing), or upper layers (data gathering, clustering). Most of the papers deal with one of the aspects that lie only in one layer, whereas some other works [5, 23, 24, 32] use cross-layer design where different issues related to more than one layer are taken into consideration in order to maximize network lifetime. Fig. 2.1 shows a brief description of network lifetime maximization techniques.

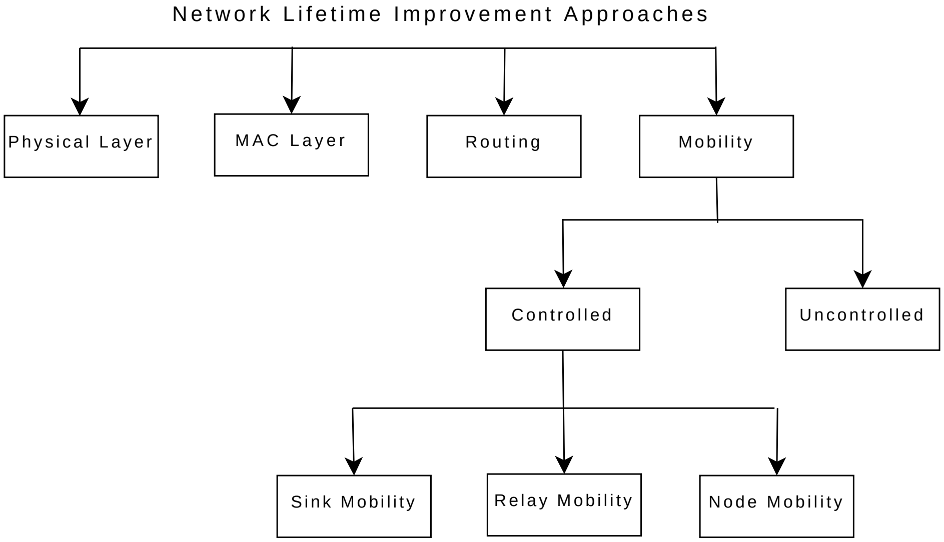


Figure 2.1: Network lifetime improvement techniques

Since communication is the main source of energy consumption, efficient power management while transmitting and receiving messages can effectively extend the operational lifetime of the network. Different mechanisms are used in order to achieve this task. [55] combines both routing and power adjustment mechanisms for network lifetime improvement. It basically proposes a routing mechanism in which each node adjusts its transmission power to send packets to its neighbors. [56] tries to evenly distribute the traffic load by dividing the area to ring zones and resets the transmission radius of each node according to which zone it belongs to. [10] emphasizes that sensor nodes are generally densely deployed to the area of interest, and therefore targets are redundantly covered. It adjusts the sensing ranges of sensor nodes and aims to find maximum number of set covers and the ranges of each sensor in this set such that each set covers all the targets.

Numerous MAC protocols are proposed in the literature that consider the special characteristics of wireless sensor networks. Almost all of them carefully treat the energy issue in sensor nodes and propose *energy-efficient* MAC layer protocols. These energy efficient protocols directly contribute to the network lifetime, since they avoid redundancy in typical operations of this layer (synchronization, control packet exchange, etc.). [14, 30, 38] are some important and widely used energy-efficient MAC protocols in the literature (more protocols, their pros and cons and comparison table, and causes of energy waste at this layer can be examined in a related survey paper [16]). Unlike the others, [33] proposes an adaptive MAC protocol that guarantees network lifetime for wireless sensor networks. It both guarantees the pre-configured network lifetime and reduces end-to-end latency by introducing an adaptive duty cycle depending on ratio of the remaining energy to the initial energy considering the pre-configured network lifetime.

Routing is another area that researchers concentrate in order to improve network lifetime. [11] states that energy consumption rate per unit information transmission depends on the choice of the next hop if the transmission power level can be adjusted. It proposes a shortest cost path distributed routing algorithm that uses link costs in which residual energy levels between two nodes are considered. [35] shows that the problem of routing messages in a sensor network in order to maximize network lifetime is NP-hard. They develop an online heuristic to maximize network lifetime which also provides larger network capacity. [51] emphasizes that performance of sensor networks are related to both time (network lifetime) and space (network coverage - source level fairness) domains. The authors develop simple, localized, and a probabilistic protocol which addresses performance issues in both time domain, by exploring multiple paths when sending messages for uniform energy consumption, and spatial domain, by reducing the load over congested nodes.

There are another classes of approaches, other than these three major ones, which are used for improving network lifetime. Topology control [2, 29], data gathering and aggregation [15,25–28,57], clustering [17,21,37], and sleep scheduling mechanisms [7,45,54] can be given as examples for these classes. Sink mobility has been emerged as another approach in this domain for the last few years.

2.2 Sink Mobility for Network Lifetime Improvement

Although the goal of all the approaches mentioned in the previous section are same, they differ in the way they consider the resulting energy consumption behavior in the network. [47] categorizes energy consumption strategies while dealing with network lifetime. Most strategies aim to minimize *average*, *maximum*, or *relative* energy consumption by using a related technique. Some routing protocols (reducing the transmission cost of a packet via optimal route) and power management techniques (each node adjusts its transmission power while sending a message) minimize the maximum energy consumption, while some energy efficient MAC protocols minimize average energy consumption (all nodes use the same sleep/idle schedules and reduce the average energy consumption). As stated in [47], in most studies neither *average* nor *maximum* strategies consider current energy status of a node. That is why, they cannot avoid the nodes whose their batteries are getting exhausted to die. Unlike these approaches, in (controlled) sink mobility, current remaining energy values of sensor nodes are taken into consideration, and this helps to extend the lifetime of nodes as much as possible. This brings a serious advantage in the case that network lifetime is defined as the time passed until the first node depletes all its energy, which is commonly used definition in the literature.

Sink mobility can be classified into two categories according to the moving strategy used: uncontrolled (random), and controlled [4, 19].

2.2.1 Uncontrolled Mobility in Wireless Sensor Networks

Uncontrolled mobility is the scheme used when mobility in wireless networks has been introduced to WSN domain [22, 43]. In this type of mobility, a third tier is used in the network (as seen in Fig. 2.2 - redrawn from [22]), in which mobile agents (a.k.a. MULEs - Mobile Ubiquitous LAN Extensions) are deployed between access points (base stations) and sensor nodes in order to collect data from sensor nodes when in close range, buffer them and finally transmit them to the sink [43]. It is called as *uncontrolled*, since movement is random and MULEs (for instance vehicles) move according to their needs and only exchange data if they encounter any node as a result of their movement [22, 43]. The main motivation behind MULEs is to reduce energy cost for data transmission by using single-hop communication (from node to MULE or MULE to sink), instead of the more expensive multihop routing. Since communication cost is the most energy consuming part in network operations, this approach effectively increases network lifetime. However, since the arriving time of any MULE (either to a node or to the sink) is not known a priori, this causes two important problems: large size of buffers that nodes should have, and large data latency. Sensor nodes should have large buffers in order to save all packets generated between two consecutive visits of the MULE. It is also unpredictable when a MULE comes close to the sink node and transmits packets to it. This can cause a huge delay between the time that data is generated and received by the sink. It is obvious that there is a trade-off between latency and energy consumption. If our application is delay tolerant, then uncontrolled sink mobility becomes a good option. Packet loss risk should also be evaluated if nodes do not have large enough buffers that can save all the packets generated between two consecutive visits of a MULE.

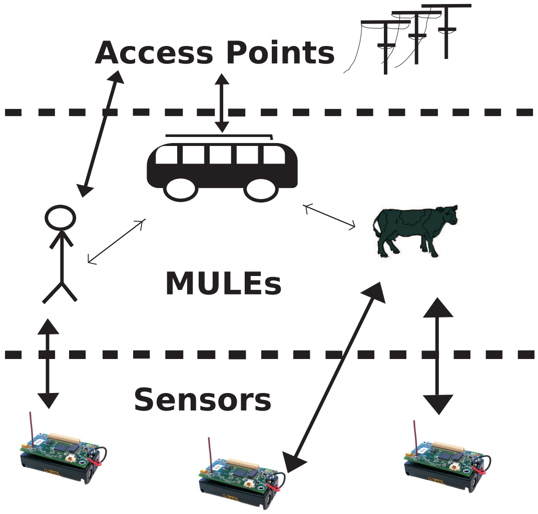


Figure 2.2: Mules with three-tier architecture

2.2.2 Controlled Mobility in Wireless Sensor Networks

Contrary to its counterpart, in controlled mobility, movements are done depending on the conditions of the network (like current energy map, node density in the regions, etc.). Currently, there are three main approaches used in controlled mobility [3]. In first and mostly used one, the sink moves among the nodes and collects data without any additional entity (which is also the case in this thesis). In the second approach, mobile relays are used as forwarding agents - like MULEs, but in a controlled manner in this case - for the communication between sensor nodes and the base station [46, 48]. In the third approach, sensor nodes themselves are mobile [49,52]. Generally, sink node or relay nodes are assumed to have more powerful energy resources such that their energies are not being depleted during the network lifetime. Therefore it is expected that mobility of these types of nodes does not adversely affect the network lifetime. However, for sensor nodes this is not the case. As it was mentioned before, sensor nodes have very limited energy batteries, which cannot be wasted for mobility, topology reconstruction, etc. unless it is certainly necessary. That is why, the first two approaches appear to be the more promising for energy efficiency and longer network lifetime [3].

Controlled and uncontrolled mobility in WSN domain is compared against each other using some performance measures in [4]. As discussed above, uncontrolled mobility has higher data latency but lower energy consumption than that of controlled one. When network traffic is low, deployment area is small, buffer size is large enough and MULE speed is quite fast, there is no packet loss in both approaches. However, as the deployment area grows and/or MULE becomes slower (inter-arrival time at the same cell increases), overflows occur in sensor node's buffers and packet delivery ratio decreases as a result. Moreover, since movements are done in random manner in uncontrolled mobility, computational cost is lower than that of controlled one. As a result, both mobility schemes have pros and cons. Basic comparison is summarized in Table 2.1 from [4]. Choosing the appropriate scheme completely depends on the application that we have. If we can tolerate data latency and some possible packet losses and/or we have relatively small deployment area and MULEs travel faster in that area, than it will be important to use data MULEs for communication in order to effectively reduce the energy consumption. However, if we have a critical application (which is the case in our work) that is intolerant to any latency or packet loss, like earthquakes, fire detection, or battlefield surveillance, then controlled mobility (via either relays or sink) become crucial. In this thesis, we focus on controlled mobility , and we propose algorithms for controlled sink mobility case, mobile relays are out of the scope of this work.

	Controlled	**Uncontrolled**
Data latency	Low	High
Energy Consumption	Medium	Low
Computational Needs	Medium	Low

Table 2.1: General comparison between controlled and uncontrolled mobility.

2.2.3 Controlled Sink Mobility

In this work we focus on the case where sensor nodes are stationary and there is not any additional tier, instead just the sink is mobile and moves among different migration points. However, movement of the sink depends on different parameters of the network (hence it is controlled). There are different works done in this track that deal with issues regarding sink mobility.

[18] examines sink mobility problem with multiple base stations (unlike our case, where we have one mobile sink). The main motivation behind this choice is to have more options for routing and reducing and retaining the hop count (so energy consumption). It presented two different integer linear programming (ILP) formulations, which have objective function either to minimize the maximum energy spent by a sensor node (BSL^{mm}) or to minimize the total energy consumption (BSL^{me}) in a round subject to some constraints, for relocation of the multiple mobile sinks (maximum 3) in each round (equal period of time, T timeframes) to prolong the lifetime of the sensor network. The paper evaluates the performance of the static and mobile approaches with 3 mobile base stations.

Since lifetime is defined as time until first node dies, (BSL^{mm}) outperforms other four schemes. They also examine the impact of the number of available base stations over the network life time and see that increasing the number of base stations beyond a certain threshold value does not improve the network lifetime (since at that time there are sufficient number of base stations in the network such that each sensor node can transmit messages via single hop communication).

Mobility and routing are considered together in [31]. It is assumed that sensors are densely deployed (as a Poisson process with density ρ) within a circle. They define the network lifetime as the time span until the first *loss of coverage*. The authors prove that, the center of the circle is the optimum location for the sink in terms of energy efficient data collection and mobility helps to balance the load and prolong the network lifetime. They define the problem with a linear programming formulation (minimizing the load on each sensor node N), and solve it first finding the optimum mobility strategy by fixing the routing strategy as shortest path routing, then use the output strategy in order to find the final routing strategy with better performance than shortest path one. After the claims and proofs which limit the mobility trajectories, finally it is proved that optimum mobility strategy is the trajectory around the periphery of the network. The authors find a 'better' routing strategy by concentrating on an inner circle in the network area and develop a heuristic using this structure. Simulations are done in order to see how results from the analytical analysis overlap with those coming from the simulation. Since there is not any result related to a comparison with any other mobility approach (like random), we cannot make comment about the performance of the proposed scheme. The main drawback of the work is the assumption that network region is circular. There is not any discussion that explains how the solution is transferred to another region types in the paper.

One of the works closer to our work is presented in [34]. In this paper, N sensor nodes and a sink node s are randomly deployed to an area of interest. There is a constant information generation rate at every sensor node and a set of locations where the base station can move and stay. The authors present two complementary algorithms for solving the sink mobility and routing problem together. One is the *scheduling* algorithm that determines the duration for each candidate sink site that the base station can stay, and the other is the *routing* algorithm in order to find the energy-efficient paths for each packet from a sensor node to the sink. Linear Programming (LP) formulation is given which maximizes network lifetime, the sum of sink sojourn times at all possible locations, subject to some constraints and compare mobile and static sink approaches with different routing schemes. In simulations, there are two scenarios including just four (centers of four sub-squares) and five different (corners and center) sink sites, respectively. Experiments are done and compared via adding the routing parameter which prevents us to observe the performance of the proposed mobility model in the paper.

One of the more recent and detailed work about controlled sink mobility is presented in [3]. They present a centralized Mixed Integer Linear Programming (MILP) model that determines sojourn times and order of the visits to sink sites. Moreover, one of the first fully distributed and localized heuristic called *Greedy Maximum Residual Energy (GMRE)* is developed as a solution of the same problem. Network model is quite similar to the one given in [34]. Unlike that model, deployment area is divided into grids and the corners of these grids are determined as sink sites. They introduce two different parameters in order to make the model more realistic. The d_{MAX} parameter represents an upper bound for the distance between the current and next site that sink can travel. This parameter bounds the time of the sink movement and enables the sensor nodes to save the packets in buffer without any loss. MILP formulation aims to maximize total sojourn time, as in [34], subject to some constraints. They evaluate the performance of MILP, GMRE, Random Movement (RM), and Static Sink approaches. MILP and GMRE gives better results than the others. MILP performs better than GMRE between 30% to %50 (for increasing t_{min} values).

Previous works that use (Mixed) Integer Linear Programming force the authors to limit the number of sink sites and number of nodes in their simulations. [18] uses maximum 30, while [34] uses 100 nodes in their simulations. Since Integer Programming/Mixed Integer Programming is NP complete [53], increasing number of nodes will cause very long delays during the sink decision process. However, it is more likely to deploy order of ten-thousand nodes to a terrain in a classical WSN application (especially when the cost of a sensor node becomes cheaper), and it is unrealistic to neither expect each sensor node has infinite or very large buffer capacity (since they have limited resources) that will not waste the packets during migration decision process, nor tolerate such a large delay for most of the typical WSN applications. Heuristic algorithms can be more advantageous and effective in this case than the optimal solution with less realistic assumptions, especially for very large number of nodes.

In this chapter, network lifetime improvement approaches are summarized using the networking layers perspective: physical, datal link (MAC), network, and upper layers. Important works in each group are explained briefly. Sink mobility is one of the approaches that lie in application layer. Uncontrolled and controlled mobility is discussed in detail. Since our work is in sink mobility part, related work in this area is given. The drawbacks of these works, which are the reasons behind the motivation of this thesis, are also discussed.

Chapter 3

Our Solution

Our goal with this thesis is to propose a set of algorithms for different aspects of the sink mobility problem in wireless sensor networks in order to improve network lifetime. Briefly, in this chapter, first our general algorithm and its steps are presented preceded by the network model and the problem definition. After that each sub-algorithm will be given in detail.

3.1 System Model and Problem Definition

We consider a wireless sensor network that has N static sensor nodes and a mobile base station. Sensor nodes are deployed to the region of interest in a random manner. After the mobile sink moves to its initial location, it broadcasts messages in order to construct a routing topology, from up to bottom. Each node, that receives the message, which means each node in the transmission range of the sender, re-broadcasts the message, putting its ID as 'parent ID' to the related part of the packet. Each node that receives the broadcast packet saves the parent ID. After the topology construction, nodes start to sense the environment. There is a constant packet generation rate Q_i for each sensor node $i \in N$. There is no data aggregation in the network; that is each parent relays its children's packets to the sink. Base station knows the exact location of the sensor nodes using an available technology like GPS or GPS-less [6,39,40] localization algorithms. Each packet will be received by the sink which implies there is no packet loss in the network (perfect MAC layer). Each sensor node has enough buffer size in order to avoid loosing packets during the traveling time of the sink from the current site to the next one (or this time is negligible). In this work, we define the network lifetime as the period of time until the first node dies, which is commonly used in the literature.

We present different heuristic based algorithms in order to bring a new approach to the sink mobility issue using the answers of three questions:

-**When** the sink decides to move to another site?
-**Where** the sink will go to as new site?
-**How** the sink decides to move or stay (obtaining the parameters)?

To decide when to move, generally, we use remaining energy level changes in the neighbors of the sink.

In typical sensor network applications (like forest fire detection), hundreds or even thousands of sensor nodes can be deployed to the region of interest. This area can be very large such that it should be very difficult to consider each node point in the area as a candidate sink site (point within the deployment area the sink can visit), since this will dramatically increase the time to decide whether to stay at the current point or move to another point in the area (and where to move if it does not stay). However, the decision process should be completed as quickly as possible. This requires determining a set of $S = 1, \ldots, q$ sink sites after the deployment, and consider only those points when deciding to change position in the area.

The main motivation behind the sink site determination algorithms is to decrease the candidate migration points in the deployment area the base station

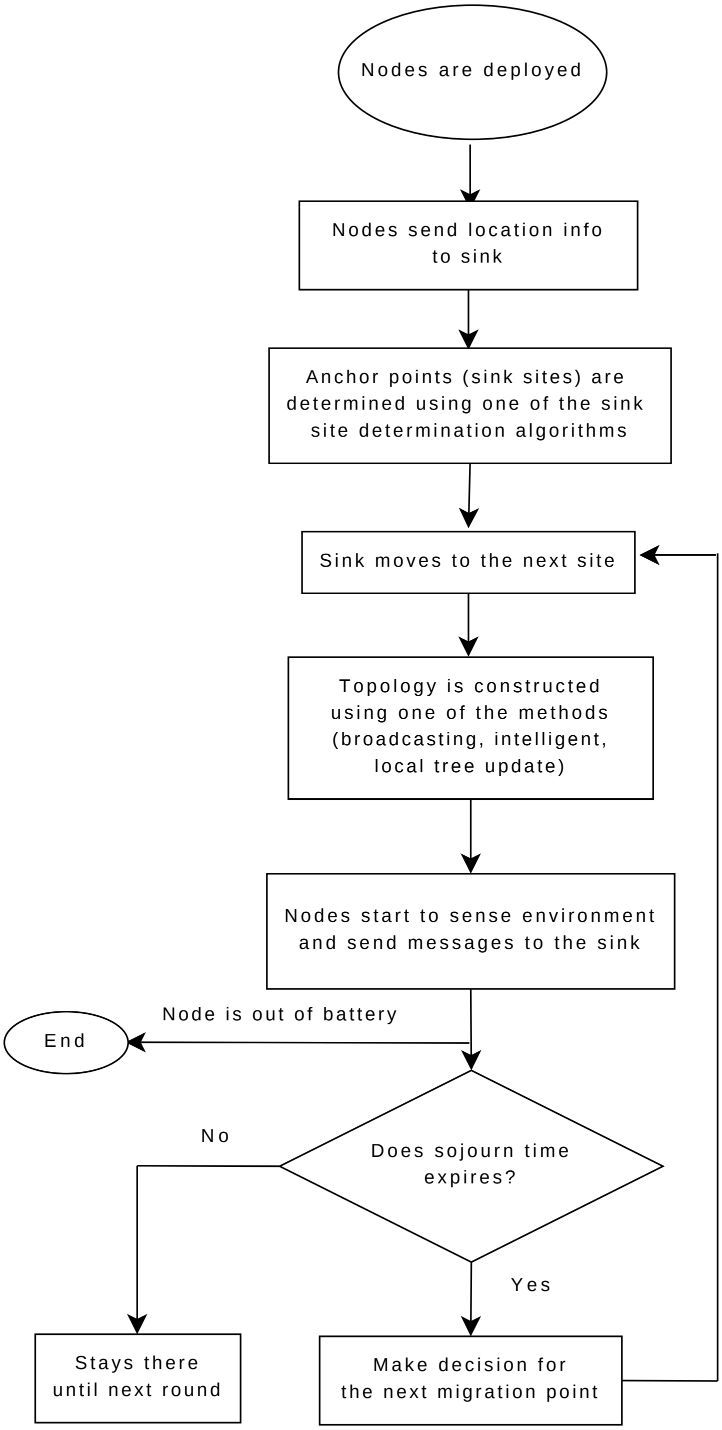


Figure 3.1: General scheme

can visit in order to minimize the decision time, in other words reduce the computational cost, when deciding the next sink site after the sojourn time [1] expires. In some scenarios, sensor nodes are deployed to an area where some points can be inaccessible or very difficult to access. This implies that sink can not visit any point in the deployment area due to some harsh environmental conditions. These reasons force us to choose sink sites before network starts operating.

In the literature, as in [3, 34], the deployment area is divided into grids (like 4x4 or 8x8) and sink sites are determined as the corners of those grids without any computation. However, it would be better to determine those sites with more intelligent algorithms using the current deployment or neighborhood information. In a network without data aggregation, nodes within the transmission range of the sink are going to relay the packets of all the nodes that are under the lower levels of the logical tree topology. This causes those nodes to deplete their energy faster than the other nodes. That is why the most important thing to consider is to group (cluster) those nodes and choose a point in such a group when determining the sink sites. By this method, we should balance the energy consumption of this *first order nodes* and prolong the network lifetime.

3.2 General Scheme

This thesis gives a set of algorithms for sink mobility and some related problems, as itemized in Table 3.1, in order to improve network lifetime. The big picture of the network operations are given in Fig. 3.1. First solution is about sink mobility which consists of both sink site determination and movement criterion. We present two different sink site determination algorithms. Each algorithm uses different mechanisms in order to choose sink sites. After that, both algorithms use the max-min based (maximum of the minimums) approach or its modified version (visit added) for moving from one point to another. Another solution we propose is a tree topology construction algorithm which aims to fairly distribute

---

[1] The sojourn time is the time the sink spends at a fixed location.

the packet load among the nodes by considering the remaining energy information. Finally, our local tree topology update algorithm tries to prevent redundant message exchanges inside the network when topology is being reconstructed at each movement of the sink.

<table>
<tr><td>Sink Mobility Scheme</td><td>Sink Site Determination
Movement Criterion</td></tr>
<tr><td colspan="2">Balanced Tree Based Topology Construction</td></tr>
<tr><td colspan="2">Local Tree Topology Update</td></tr>
</table>

Table 3.1: Network lifetime elongation elements proposed in the thesis

3.3 Sink Mobility Scheme

Sink mobility scheme has two major steps: finding candidate migration points (sink sites), moving through these migration points. For the first step, two different sink site determination algorithms are presented which are based on locations (coordinates) of the nodes on the deployment area, and on neighborhood relationships, respectively, in the following two sections. After that, how (using which parameters and how to obtain them) and when to migrate to those points are discussed. Finally, a cost function is introduced in order to examine the problem of how to choose a new migration point from a set of points when the nodes have limited buffer capacity.

3.3.1 Sink Site Determination

After hundreds of nodes are deployed to an area of interest, there is infinite number of points in the area that sink can move. The number of possible points will be in the order of hundreds, even we limit the points to the coordinates of the sensor nodes. Large amount of candidate points causes too much computational cost while deciding where to move at the end of each round. If sensor nodes do not have an infinite buffer (or sufficiently large one that can store all generated traffic during decision process), and if decision takes too long, loosing data packets becomes inevitable. There can be lots of applications which cannot tolerate such a case. Also, because of some typical sensor network applications (like environmental monitoring), nodes can be deployed to areas that can have harsh conditions (like boggy, hilly) into which the sink node cannot move. All these cases require to select some set of points (i.e. sites, anchor points, candidate migration points) that the sink node can potentially migrate before the network starts to transport data.

To the best of our knowledge, there is not any study about sink mobility that includes a dynamic sink selection process. All of them are off-line and does not include any detail about the deployment of sensor nodes or other physical conditions. Most of them just use corner points of the square or grids in the area [3, 34]. Unlike other works, by these two algorithms, we try to give different sink site selection mechanisms which treat the network conditions as well.

3.3.2 Neighborhood Based Sink Site Determination Algorithm

Sometimes, it can be difficult to know the exact boundaries of the deployment area and the coordinates of each sensor node in the region. In this case, the neighborhood information of the nodes can be used for determining candidate sink positions. Since every node receive/hear messages during network operation, or can exchange neighborhood information using control packets, each node can know its neighbors independent of their coordinate information.

Since the main motivation of sink mobility is to evenly distribute the heavy load of the base station's neighbors, we want to group those nodes via an algorithm in order to be chosen as sink's neighbors. The neighborhood relationships are used for solving this problem. If we are given n nodes and neighborhood relationship information, then our aim is to choose q nodes from the list such that,

$1 \rightarrow \{2, 3, 5\}$
$2 \rightarrow \{1, 3, 5\}$
$3 \rightarrow \{1, 2\}$
$4 \rightarrow \{5\}$
$5 \rightarrow \{1, 2, 4\}$

Table 3.2: Neighborhood relationships of a sample toy network

union of the neighbors of these nodes cover all the nodes in the area. This process is quite similar to finding a *dominating set* for a graph $G = (V, E)$, which is defined as every vertex not in dominating set D is adjacent to at least one vertex in D [8]. Dominating set problem is a special instance of *set covering* problem, and it is NP-complete [44]. [2]

A sample neighborhood relationship of a five node network is given in Table 3.2. Each node's neighbors are given in curly brackets. We can see $\{1, 2, 3, 4, 5\}$ as *universal* set and their neighborhood information as subsets of this set. If nodes 1 and 5 are chosen then, all of the nodes will be covered with repetitions (for 1 and 2)

We present a greedy heuristic algorithm for dealing with dominating set problem. In the beginning, after determining the neighborhood information of each node (either by computing at the sink using coordinates or collecting neighbor information from the nodes to the base station), the sink node sorts the nodes in descending order with respect to their number of neighbors. Then the heuristic algorithm takes the coordinate of the node (a contributed node) with the most number of neighbors in the beginning and put those neighbors to the current neighbor list. After this step, the Algorithm 1, keeps the list of covered and uncovered nodes at each step. After first contributed node is chosen (the node with the most number of neighbors), its neighbors are saved in $coveredNodes$ list. The $uncoveredNodes$ list is simply calculated via taking set difference of universal set (all nodes) and $coveredNode$ list. After initialization of those lists, node that has

---

[2] The decision version of set covering is NP complete, and the optimization version of set cover is NP hard since it generalizes the NP-complete vertex-cover problem [12].

the maximum number of common elements with $uncoveredNodes$ is chosen as the next contributed node. Then its neighbors are added to $coveredNodes$ list and $uncoveredNodes$ list is updated. This iteration continues until $uncoveredNodes$ list becomes empty ($coveredNodes$ equal to universal list). Figure 3.2 shows the possible sink sites (stars) as the output of the algorithm. If we look at the complexity analysis of the algorithm, *while* loop will iterate $n-1$ times (where n is the number of nodes) if each node has at least one common element, and inner loop also iterates n times, therefore the complexity is $O(n^2)$ in the worst case.

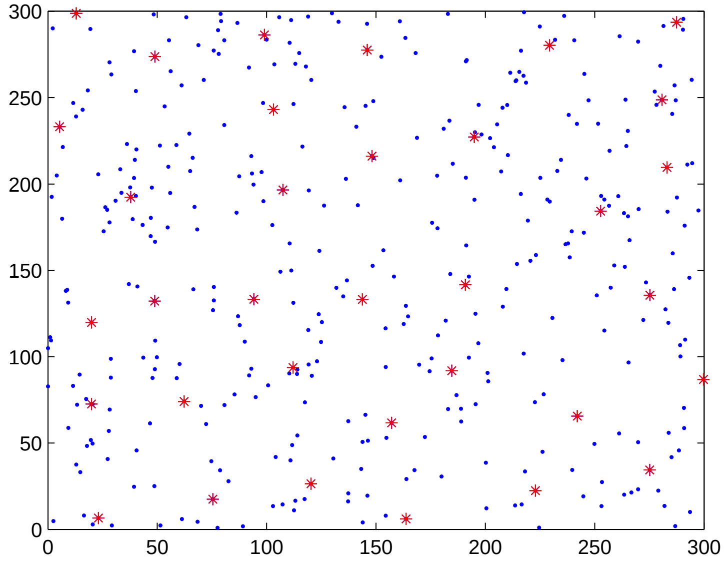


Figure 3.2: Neighbor based determined sink sites with dominating set heuristic

3.3.3 Coordinate Based Sink Site Determination Algorithm

In the literature, it is generally assumed that base station knows the exact location of sensor nodes using either GPS modules located on the sensor nodes or GPS-less localization algorithms. With that information it would become possible to group those nodes using their coordinate values on the sink side.

Algorithm 1: Neighborhood based sink site determination algorithm - dominating set heuristic(*neighborhoodRelationships*)

```
foreach node i in area do
    $neighbors[i]$ = nodes in the transmission range of node $i$;
    $neighborSize[i]$ = size($neighbors[i]$);
end
$sortedIndices \leftarrow sort(neighborSize, `descendingOrder')$;
$ind = 1$;
$contributedNode(ind) = sortedIndices(1)$;
$coveredNodes = contributedNode(ind).neighbors$;
$uncoveredNodes = universalSet - coveredNodes$;
$currentNeighborsList = neighbors(sortedIndices(1))$;
$ind = ind + 1$;
while $uncoveredNodes = \phi$ do
    foreach node i in sortedIndices do
        $commonElements(i)$ = size(intersect($neighbors(sortedIndices(i))$,
        $uncoveredNodes$));
    end
    $indx$ = index of $max(commonElements)$;
    $contributedNode(ind) = sortedIndices(indx)$;
    $ind = ind + 1$;
    $coveredNodes = coveredNodes + contributedNode(ind).neighbors$;
    $uncoveredNodes = universalSet - coveredNodes$;
end
foreach node i in contributedNode do
    $candidateMigrationPoints(i) = coordinate(contributedNode(i))$;
end
return   $(candidateMigrationPoints)$;
```

In the coordinate based sink site determination algorithm, we divide the deployment area into squares such that each one's length is equal to the transmission range. That enables us to group (cluster) the nodes that can be the sink's neighbors in any round and compare their energy levels and decide which subarea to move in the next round. The number of areas is dynamically changing according to the transmission range values.

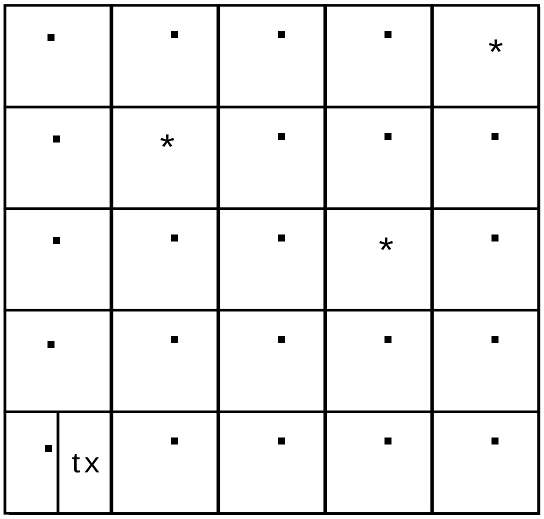


Figure 3.3: Coordinate based determined sink sites using squares

As it can be seen in Fig. 3.3, the distance between any two neighbor sink sites is R, where R is the maximum transmission range. Each sink site is ideally placed at the center of the allocated area. The detailed algorithm is given in Algorithm 2. After determining the centers of each sub square, sparse areas are eliminated if their *density* is below the threshold, where threshold is determined by dividing the number of nodes to the number of sub squares.

Since each node's coordinate is known by sink node, the area is not needed to be regularly allocated to the squares in order to use this algorithm. The sink node can choose the node that has *minimum (x,y)* pair and assumes that it located on the left lower corner of the imaginary sub square. Then it chooses the center of this sub square as candidate migration point, and continues this operation, until all the nodes in the area are covered. Since the algorithm iterates two times over the number of sub squares, its complexity is $O(s)$, where s is the number of sub squares and is calculated as $(edgeLengthOfSquare/transmissionRange)^2$.

A dynamic sink site selection algorithm (either neighborhood or coordinate based) provides us to eliminate the areas that are on inaccessible terrains which prevents sink to move and stay at that point. For instance, in Fig. 3.3, the ones with * are inaccessible areas that have hard physical conditions (boggy area, hills, so on) are eliminated before network operation starts. We also eliminate the parts (in coordinate based part, either square or circle) that do not have any node in the sub area during the algorithm.

Algorithm 2: Coordinate based sink site determination algorithm $(nodeCoordinates, txRange)$

```
$numberOfSubsquares \leftarrow$
$square(edgeLengthOfSquare/transmissionRange)$;
$ind = 1$;
foreach $i \leq numberOfSubsquares$ do
    $precmp(i, :) = [x_{center} y_{center}]$
end
$ind = 1$;
foreach $i \in precmp$ do
    if $precmp(i, :)$ is accessible and density $\geq$ threshold then
        $candidateMigrationPoints(ind, :) = precmp(i, :)$;
        $ind = ind + 1$;
    end
end
return $(candidateMigrationPoints)$;
```

3.3.4 Sojourn Time and Movement Criterion

After candidate sink sites are determined, the sink node moves to the densest point of the area (first migration point) and the routing topology (i.e. tree) is constructed (either via broadcasting, or some intelligent topology construction mechanism). Sink gets the remaining energy values from its neighbors in order to learn the minimum energy level in the one-hop neighbors before packets start arriving. Since energy levels are piggybacked in each packet, sink can have the chance of comparing the current minimum energy value and the starting one. If the difference between them is one or more levels, then the sink starts the decision process to determine the location to move for the next round. The number of energy levels (L) determine the sojourn time of the sink for that location, which is the time until any node spends $\frac{1}{L}$ of its energy. Too many levels means shorter sojourn times. For example, if we have small number of energy levels, like 4, then 25% of the node's energy must be depleted in order to trigger another migration process. Some applications should require a minimum sojourn time on a site in order to ensure data quality. For these cases, we introduce the value t_{min}, which is defined as the minimum time that a sink should stay on the current site.

Such a dynamic approach is more advantageous than a static approach where fixed number of rounds are used. For instance, sink can immediately move to another site if a sink neighbor has tremendous packet load and dramatically looses its energy, however fixed round approach will wait there until the number of rounds are completed and possibly causes the node to die.

If sojourn time expires (either exceeds t_{min} or a change in energy level occurs), the sink examines the minimum remaining energy value in each candidate migration point, which means the minimum energy value among the nodes' energy values that fall into the circles or the squares, using the information in the last received packets. Then it moves to the point where the minimum remaining energy level is maximum among the sites that have not been visited yet (max-min approach). When we say '*have not been visited*', we mean a site cannot be visited until sink has moved to all of the candidate migration points once. After all visits are done, then the *visited* flag will be set to zero for all of the sink sites and they all become available to visit again. The motivation behind this approach is this: if we just use max-min approach then we may stuck to a single local maximum and cannot focus more into a general picture. In other words, when we are only interested in energy dimension of the problem, then we can ping pong among a few sink sites and have similar packet load patterns (if a deterministic topology construction algorithm is used like in the next section). However, if we visit different sites, then we can achieve more uniform packet load distribution. Therefore this visit added max-min approach, which is summarized in Algorithm 3, corresponds to visiting possible sink sites with an order in which the node with maximum of the minimum energy values in the sites takes precedence. Since the algorithm iterates over the number of migration points (m), and calculates minimum energy among the nodes on each site, its complexity is $O(m \cdot n)$ (where n is the number of nodes) in the worst case.

Algorithm 3: Migration criterion - visit added max-min approach

```
$notVisitedList = migrationPoints$;
start with $migrationPoints[0]$;
$notVisitedList = notVisitedList - 0$;
while first nodes depletes its energy do
    update $EnergyList$ after transmissions;
    if energyLevelChange of any node j >= 1 then
        foreach point i in notVisitedList do
            $neighbors[i]$ = nodes in transmission range of
            $migrationPoints[i]$;
            foreach node k in $neighbors[i]$ do
                $energyList(k)$ = energy of node $k$
            end
        end
        $minimumEnergyNode$ = index of $min(energyList)$
        $minimumEnergyList[i] = minimumEnergyNode$ $newIndex$ =
        index of $max(minimumEnergyList)$;
        $nextSite = notVisitedList[newIndex]$
        $notVisitedList = notVisitedList - newIndex$;
        if $notVisitedList == \phi$ then
            $notVisitedList = migrationPoints$;
        end
        migrate to $nextSite$;
    end
end
```

3.3.5 Introducing Cost Function

Some applications can require high number of sensor nodes that are deployed to an area that can be order of tens of kilometers square. In that case, movement time of the sink from one point to another should increase dramatically. This situation directly affects other nodes since they continue sensing the environment and generating packets. Because of not using data aggregation, high data generation rate (like a packet for each second) causes each node to buffer many packets until the sink node reaches the new site for the next round. In previous sections, it is assumed that nodes have unlimited buffer capacity and do not loose packets, and therefore the sink node can move to any migration point in the area. However, in real scenarios, this is not the case. Sensor nodes have limited buffer capacity and after a while they start loosing packets when new packets enter to the queue. For some critical applications, including battlefield surveillance, earthquake and fire detections, it is generally undesirable to loose any single packet. This forces us to define a cost function. For instance, assume that the sink node is on a vehicle which has a speed of 1 m/s. Each sensor node has buffer capacity of, say 50 packets, and packet generation rate Q_i is 1 packet per second. In this case, if we ignore the topology reconstruction time, this means that the sink node can move to a point that is at most 50 meters (called d_{max}) away from its current position without loosing any packet. For such cases, we propose to consider only the set of nodes inside a circle of d_{max} radius as candidate migration points, and try to move to one of them which is the most energy efficient. This approach is summarized in Algorithm 4. In the worst case scenario, its complexity is $O(m \cdot n)$ where m is the number of migration points and n is the number of nodes.

Algorithm 4: Centralized migration algorithm with cost function

```
foreach point $i$ in migrationPoints do
    $neighbors[i]$ = nodes in transmission range of $migrationPoints[i]$;
    foreach node $k$ in $neighbors[i]$ do
        $energyList(k)$ = energy of node $k$
    end
    $minList[i]$ = index of $min(energyList)$;
end
$maxIndex$ = index of $max(minList)$ ;
$newSite = migrationPoints(maxIndex)$;
migrate to $newSite$
```

3.4 Load Balanced Topology Construction Algorithm

Usually tree-based routing topology is constructed using a broadcast mechanism as follows: after the mobile sink moves to its initial location, it broadcasts messages in order to construct the topology from up to bottom. Each node that receives the message, which means the node is in the transmission range of the sender, re-broadcasts the message after putting its ID as 'parent ID' to a field of the packet. Each node that receives the broad-cast packet saves the parent ID.

In the approach above, current energy levels of the nodes are not taken into consideration. Proposing an intelligent algorithm, which considers the current energy level and packet load of the nodes should yield a better network lifetime. Algorithm 5 gives a balanced tree based topology construction mechanism. Sink's neighbors are in the first level in the logical tree, neighbors of its neighbors are in the second level, and so on. For each node in level l, if a neighbor node is in the logical level $l-1$, it becomes a candidate parent and its ID is put into the parent list. After calculating these values, the algorithm is started to run in the last logical level of the tree, namely leaves. Nodes in the last level of the tree are sorted according to the number of candidate parents in ascending order. The main motivation behind sorting the nodes is to give priority to the nodes with less number of options. By this way, when we come to nodes with more options, nodes have updated packet loads, that's why a better decision can be made among many options. If a node has only one parent in its list, then this node is assigned as the parent node and its packet load is also incremented by the packet load of the child. If a node has more than one candidate parent, the ratio of $energylevel/packetLoad^2$ is calculated for each candidate. The candidate with maximum ratio value is assigned as the current node. Since the algorithm is run from bottom to up, the packet load of the most critical nodes (i.e. sink neighbors) can be determined using the full information of the nodes below.

The algorithm consists of two main *for* loops. The first loop's complexity is $O(n)$. The second loop iterates for each candidate parent of each node in each level in the tree. The outer two loops iterate over the all nodes in the area (iteration is over the nodes level by level). In the worst case, a node can access to all nodes in one hop, therefore its number of neighbors can be equal to $n - 1$. In this case, we have two loops which iterate for n and $n - 1$ nodes, respectively, which yields $O(n^2)$ complexity.

Algorithm 5: Load balanced topology construction algorithm

```
sink initiates broadcast;
foreach node i in area do
    $logicalLevel[i]$ = hop distance to sink node ;
    $parentList[i]$ = nodes which packets reach from upper level;
end
for level = bottom to first do
    sortedNodeInCurrentLevel = sort(size(ParentList), 'ascendingOrder');
    foreach node i in sortedNodeInCurrentLevel do
        if $size(parentList[i]) == 1$ then
            $nodes[i].parentId = parentList(1, 1)$;
        end
        else
            foreach node j in parentList do
                $ratio(j) = energylevel(j)/currentPacketLoad^2$
            end
            $maxRatioIndex$ = index of $max(ratio)$
            $nodes[i].parentId = parentList(maxRatioIndex)$;
        end
        update packetLoad for the parent
    end
end
```

3.5 Local Tree Topology Reconstruction

One of the disadvantages in sink mobility scheme is to re-construct the topology for each movement of the sink. This will increase the number of control/management packets in the system, which also implies higher overhead. Since each node transmits (broadcasts) and receives a packet during the topology re-construction process, for short period of sojourn times the number of sink movements will be increased and energy of each node will be wasted for control packets. For higher number of nodes, this process will be annoying and will take more time and will prevent the sensor nodes from transmitting data packets (instead store them in the buffer) until the topology is fully constructed. This can be a problem for some applications, such as earthquake and fire detection, which do not tolerate delay in transmission of packets. Therefore, for both energy and application specific purposes, local tree topology re-construction can be a necessity in some cases as part of a sink mobility scheme in order to improve network lifetime and decrease data latency. Details of such an algorithm is given in Algorithm 6. When a node receives a packet from sink, it always rebroadcasts the packet, even it was one of the sink's neighbors in previous round. For others that receive packet from non-sink nodes (i.e. ordinary nodes), it checks whether the sender was its parent in the previous round or not. If it was, then it stops to rebroadcasting packet to the network, otherwise rebroadcasting continues. Since the algorithm iterates over the number of nodes in the area, its complexity is $O(n)$.

Algorithm 6: Local tree re-construction algorithm

```
sink initiates broadcast;
foreach node i in area do
    if packet received from sink then
        re-broadcast the packet with id $i$
    end
    else
        if packet received from parent in the previous round then
            stop broadcasting
        end
        else
            re-broadcast the packet with id $i$
        end
    end
end
```

The main disadvantage of a local update algorithm is having longer paths than its fully reconstructed counterpart. This can increase the energy needed for a packet while reaching to the sink (number of transmissions should increase). However, for highly mobile environments (sink's sojourn time is very short or it is continuously moving), local update mechanism become more advantageous, since topology construction cost increases for full update one.

In this chapter, we define the problem of sink mobility and give the general solution scheme that is used in the thesis. Two different sink site determination algorithms are given for each of the cases that either the sink node does not know the exact coordinates of the nodes (neighborhood based) or it can calculate these coordinate values either with a GPS module or a localization algorithm (coordinate based). After the selection of candidate sink sites, sojourn time expiration and movement criteria are given to answer the *when* and *where* to move during the network lifetime.

Since topology is reconstructed after each sink movement, it should be advantageous to do it with an energy efficient mechanism in order to increase network lifetime. Load balanced topology construction algorithm is proposed in order to reduce the related cost by using current energy levels of nodes in the network. Finally, local tree topology reconstruction algorithm is given for reducing the number of messages exchanging during the topology reconstruction period. By this algorithm only local changes are made in the topology instead of the whole topology reconstruction from the beginning.

Chapter 4

Performance Evaluation of the Algorithms

In this chapter, we present the results of the experiments that are done for evaluating the performance of the algorithms presented in the previous part. We used MATLAB as the simulation environment.

Simulations are done for all of the three main elements of the general scheme: sink mobility, balanced tree based construction, and local tree update mechanism. Different metrics (network lifetime, packet latency, etc.) are examined for each of the category.

4.1 System Model and Main Parameters of the Simulation

Sensor networks in the simulation have N static sensor nodes and a mobile base station. Those nodes are deployed to a region of interest randomly (if not stated otherwise). Square areas are used in the simulations, which are generally either 300x300 or 400x400 m. After the mobile sink moves to its initial location, it broadcasts messages in order to construct a tree-based routing topology, from up to bottom (if balanced tree based topology construction is not used). After the topology construction, nodes start sensing the environment. There is a constant packet generation rate Q_i (1 packet/s) for each sensor node $i \in N$. In this work, we define the network lifetime as the period of time until the first node dies, which is a commonly used definition in the literature.

The energy model and the radio characteristics used in the simulations comes from [20]. In this first order radio model, to transmit a k-bit message to a distance d, the radio expends energy as given below:

$$E_{Tx}(k,d) = E_{Tx-elec}(k) + E_{Tx-amp}(k,d) \tag{4.1}$$

$$E_{Tx}(k,d) = E_{elec} \cdot k + \epsilon_{amp} \cdot k \cdot d^2 \tag{4.2}$$

To receive a k-bit message:

$$E_{Rx}(k) = E_{Rx-elec}(k) \tag{4.3}$$

$$E_{Rx}(k) = E_{elec} \cdot k \tag{4.4}$$

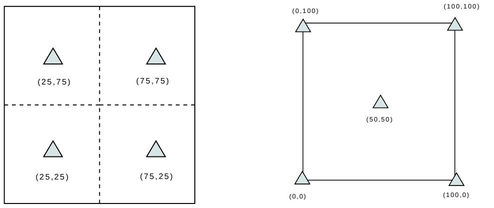
Transmission energy cost is related with the number of bits and the square of distance, whereas receive energy cost is related with the number of bits. In our simulations, this energy model is applied with 50 bytes data packets and 20 bytes control packets. The radio dissipates $E_{elec} = 50$ nJ/bit to run the transceiver circuitry and $\epsilon_{amp} = 100$ pJ/bit/m^2 for the transmit amplifier to achieve an acceptable $\frac{E_b}{E_n}$ [20]. Each sensor node has energy of 10 J, initially. If not stated otherwise, this energy is divided to 10 levels, and represented in four bits which are piggybacked in the data packets.

4.2 Sink Mobility Scheme Experiments

In this section, we investigate the performance of the algorithms related to mobility aspect of the network. First, we evaluate the two proposed sink site selection methods with another three in the literature in terms of network lifetime and data latency. Next, experiments about performance of different movement criteria (Residual Energy Based and Random Movement) are done for given sink sites. Moreover, different parameters (minimum sojourn time, maximum movement distance, etc.) of sink mobility scheme are examined in detail.

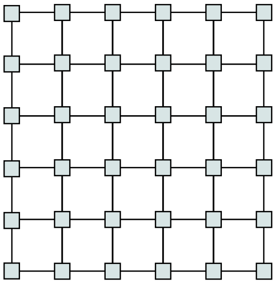
4.2.1 Sink Site Determination Experiments

Sink sites are determined in order to answer *where* to move base station during the network operation. Papers in the literature try to solve this issue by assigning a set of predefined points of the area. Three of them are summarized in Fig. 4.1. P1 and P2 are given as sink sites in the paper of Papadimitriou et al. [34]. In Fig. 4.1a, center points of four grids are chosen as sink sites, whereas the second one takes four corner points and the center of the big square (coordinates are given for 100x100 square). In the third approach, which comes from Basagni et al. [3], area is divided to 3x3 (5x5) grids and corner points, totally 16 ones (36 ones), are taken as candidate migration points (B1 and B2 respectively). We evaluate the performance of our approaches (Neighborhood based set covering heuristic - NB, and Coordinate Based Exhaustive - CB) with these four methods.



(a) Sink Sites - Approach 1 (P1)

(b) Sink Sites - Approach 2 (P2)



(c) Sink Sites - Approach 3 (B1)

Figure 4.1: Different sink site selection approaches.

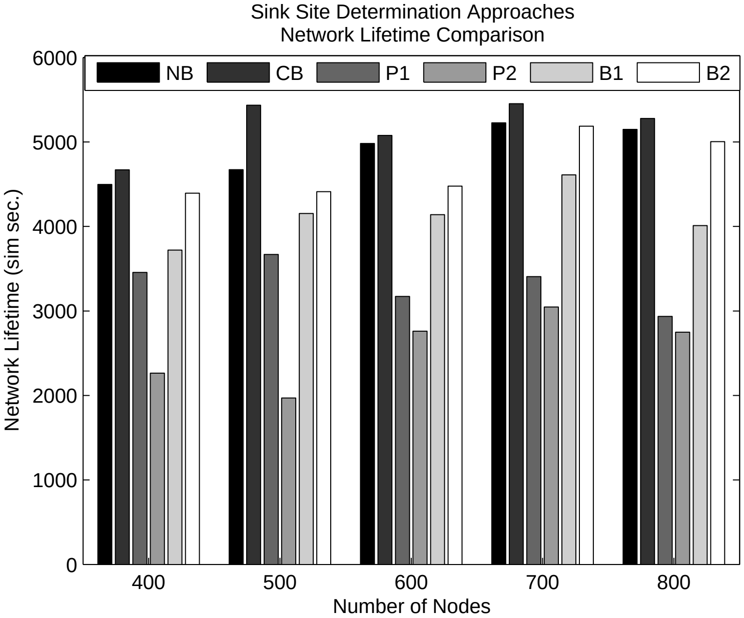


Figure 4.2: Sink site determination approaches: network lifetime comparison.

As can be seen in Fig. 4.2, both Neighbor and Coordinate Based approaches are better than other four in terms of network lifetime. CB approach is three times better than P2 for 500 nodes as well. When we look at Fig. 4.3, it can be seen that, P1 has the best data latency among all others although it has lower network lifetime than the other four approaches. Since, 4 different sites have been optimally placed to the center of the four grids, it has better data latency (average hop count). Although NB and CB approaches have 25% worse data latency than P1, they have better network lifetime, up to 60%. NB and CB have better data latency than the other three in all cases as well.

From now on, we try to investigate about the *when*, and *where* questions of the mobility scheme. Before going into detail, it would be better to give information about the general structure of the experiments. As the first step of our overall scheme, we choose a sink site determination method mentioned in the previous section, either one of the coordinate (area) based sink site determination algorithms or neighborhood based sink site determination algorithms. For the second step, either max-min(MM) approach or visit added max-min approach (VMM) or random movement approach (RM) is chosen as a strategy when moving

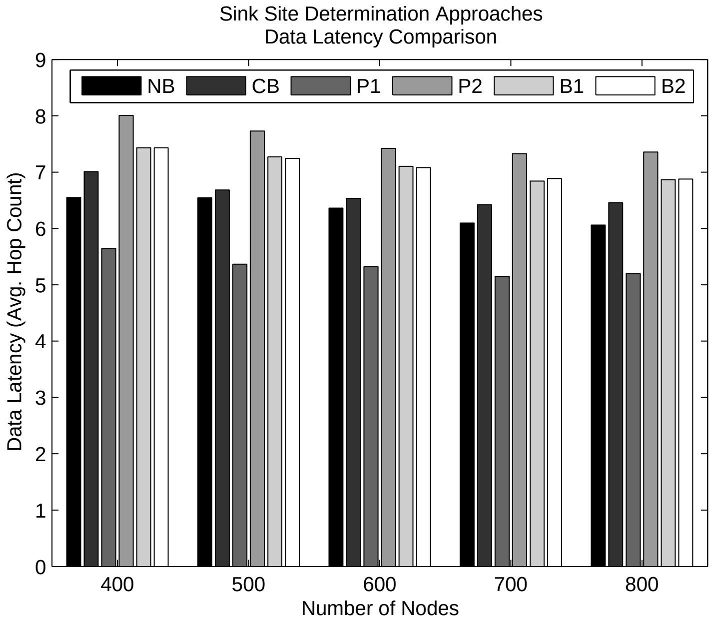


Figure 4.3: Sink site determination approaches: data latency comparison.

through migration points. In RM, when sojourn time (the amount of time that sink spends before leaving that area) expires (same strategy is used as in all other approaches), base station moves to the coordinate of a random sink site in the area. As the fourth approach, Static Sink (STS) approach is used for both of the cases. As its name implies, in this case, the sink does not move to any point in the area, instead it is placed at the center of the area, which is the point that maximizes the network lifetime [31]. In all approaches, if one of the neighbors of the sink loses one ore more levels of energy, then the sink decides to move to another point (sojourn time expires).

4.2.2 Energy Level Experiments

After determining the sink sites, we should decide how often to move the sink node in order to obtain better network lifetime. In previous chapter, it was mentioned that initial energy of each sensor node is divided to L levels, and when one of the sink neighbors looses its $1/L$ amount of energy, then the base station initiates a movement decision process. In order to choose the most efficient L value, we did experiments for 400 nodes which are randomly and uniformly deployed to an area of 300x300 m. with transmission range of 30 m. and initial energy of 10J. Before network started to operate, sink sites were chosen using the neighborhood based algorithm (dominating set heuristic).

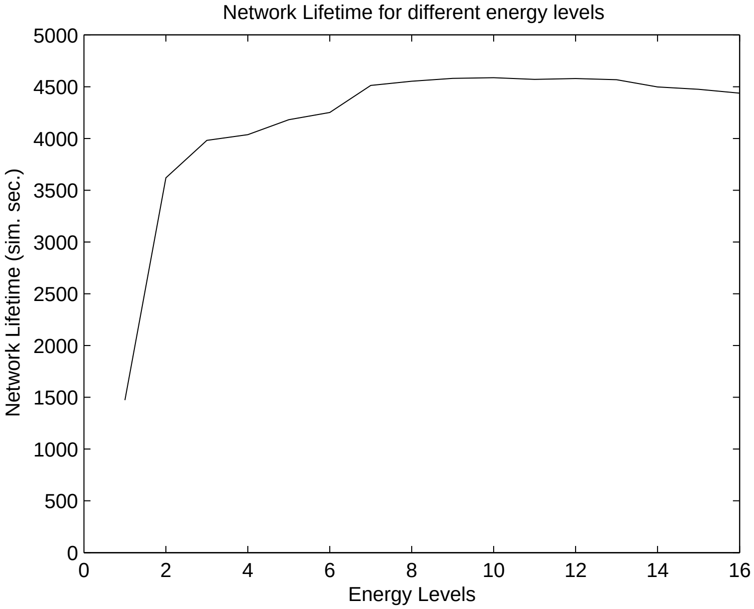


Figure 4.4: Network lifetime for different energy levels. (SSD = DSH and $t_x = 30m$).

When we look at the Fig. 4.4, it can be seen that the network lifetime improves until level becomes 10, and then slightly decreases when level increases to 16. However, the improvement rate between level 3 and level 10 is much more lower than that of between level 1 and level 2. Level 1 means that there is no movement (the sink node is static), when we divide levels to 2, then lifetime becomes 2.5 times better than the previous case while it improves 25% when energy is divided to 10 levels. After 10 levels, network lifetime slightly decreases (for level 16 it is 5% lower than level 10). Since there is a tradeoff between movement frequency and topology reconstruction cost, we expect an improvement until some point and then a decrease should occur because that frequency cannot compensate the topology reconstruction cost. This is the trend that we see in Fig. 4.4.

When we look at for higher energy level values in Fig. 4.5, we can see another

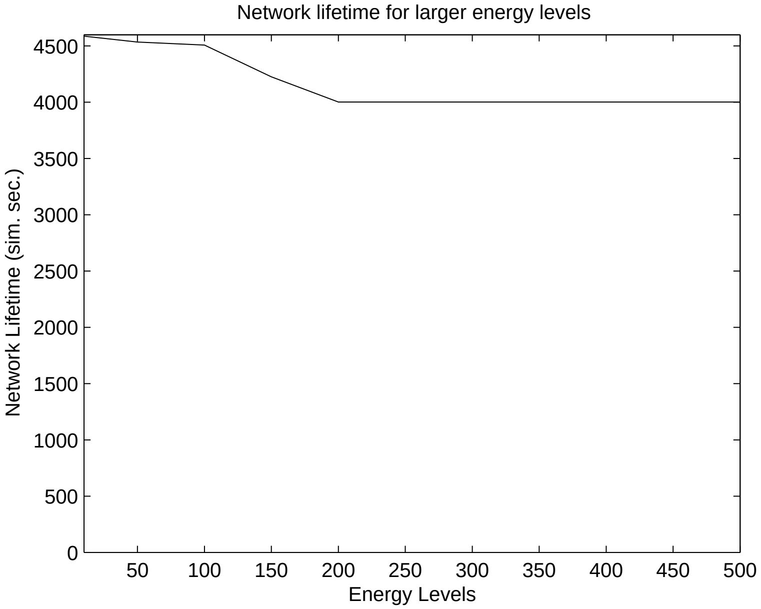


Figure 4.5: Network lifetime for larger energy levels. (SSD = DSH and $t_x = 30m$).

observation. For larger energy level values, network lifetime becomes constant after a l value. This is an expected result, since after a value l, dividing to larger energy levels (and obtaining smaller intervals) do not affect the movement frequency (the sink node starts to move at each round - one message for each node is collected - by dividing to energy to l levels). If base station's neighbors consume u amount of energy on the average in each round, then $l = initialenergy/u$ will give the l value to us. This value is 200 in our experiment. However it can change with the initial energy and number of nodes in the area. For the rest of the experiments we divide the initial energy to 10 levels.

4.2.3 Sink Movement without any Constraint

In the first case, there is not any constraint that delays sink to move to any point like d_{max} (maximum distance to move) and/or t_{min} (minimum staying time). Sink can move to any point in the area without considering possible data loss, or do not need to wait extra time before moving. For this experiment, different number of nodes are uniformly deployed in a 300x300 m area. The transmission range is

35 m. Dominating set heuristic is chosen for sink site determination algorithm (SSD = DSH). Results are shown in Fig. 4.6 for varying number of nodes. As it is seen, VMM perform better than others for most cases. When number of nodes increases, the difference between VMM and others also increases. Data latency values of those approaches can be seen in Fig. 4.7. As it is seen, static sink has lowest latency (since it is placed in the center of the area which is optimal and stays there at the end of the network lifetime), random movement follows it (it tends to move the sink to the center of the area mostly). VMM has lower latency than MM, and this can be seen an achievement, since both latency is decreasing and network lifetime increasing at the same time. RM has lower latency than VMM, since VMM uses more intelligent approach and higher network lifetime (there is a trade-off between latency and network lifetime). However when time goes to infinitive, on the average, RM visits each site for equal number of times and this balances number of hop counts to the sink.

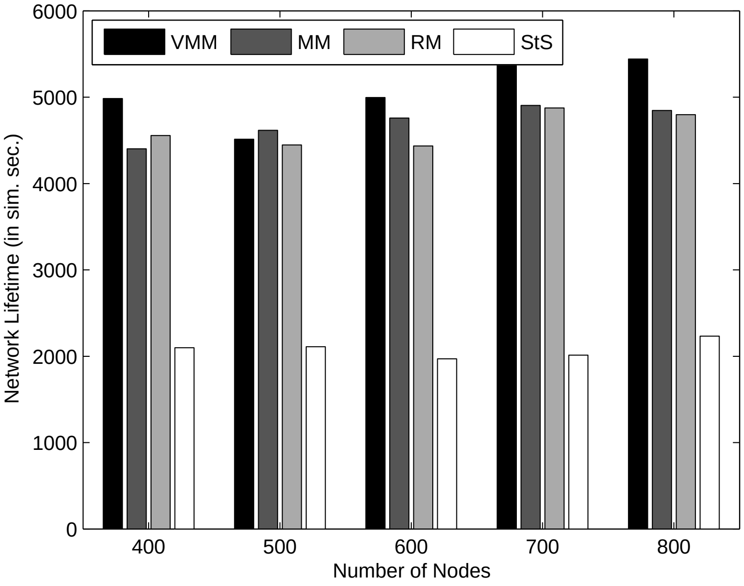


Figure 4.6: Network lifetime without any movement constraint(SSD = DSH and $t_x = 35m$).

When we look at the energy map snapshots of the network (which were taken after the first node died) for different approaches, from Fig. 4.8 to Fig. 4.10,

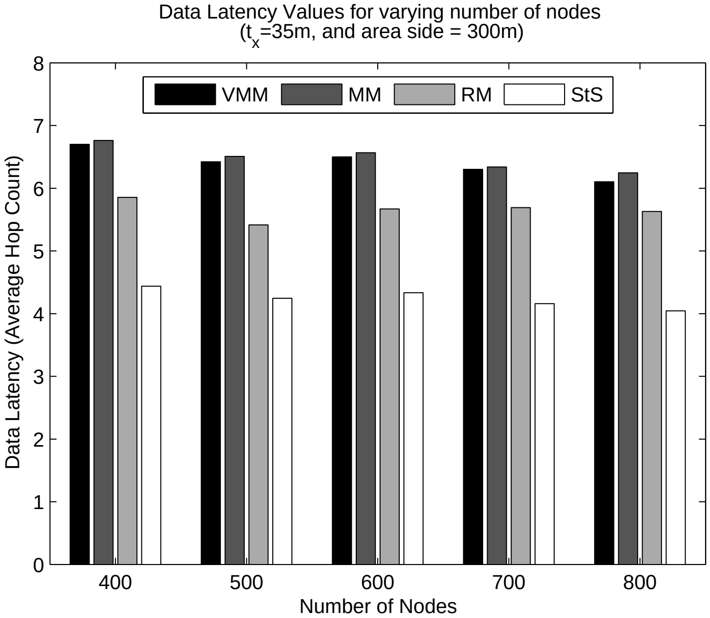


Figure 4.7: Data latency values for varying number of nodes (area side = 300m and $t_x = 35m$).

we can see that VMM approach utilizes the network much more than other approaches. RM generally has lower energy at the center, which is also the case as in static sink approach. For all approaches, the nodes on the corners of the area have much more energy than the others. In static sink and random movement cases, they have 90% of their energies.

4.2.4 Moves around d_{max} Value

In the previous section, it is assumed that sensor nodes have unlimited buffer capacity, which is not the case in practical life. For a moderate packet generation rate and an area with average length, nodes start to loose some of the packets, if sink movement time exceeds a threshold. Using the packet generation rate, buffer capacity and sink movement speed, a d_{max} value can be calculated. d_{max} specifies the maximum distance between sink's two consecutive sites (or movements). In other words, when sink decides to move, it can only go to a sink site that is further away at most d_{max} distance. Fig. 4.12 shows the result of the network

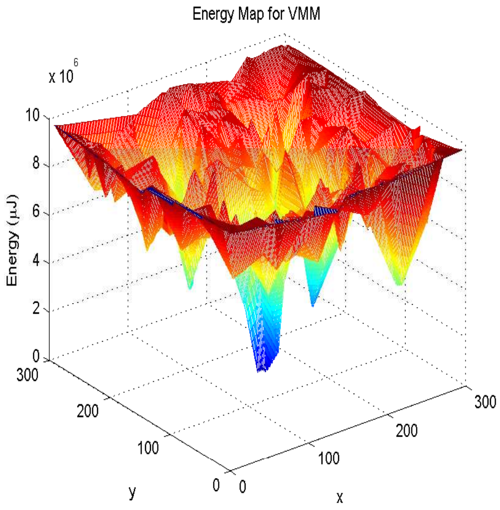


Figure 4.8: Energy map of VMM approach after first node death

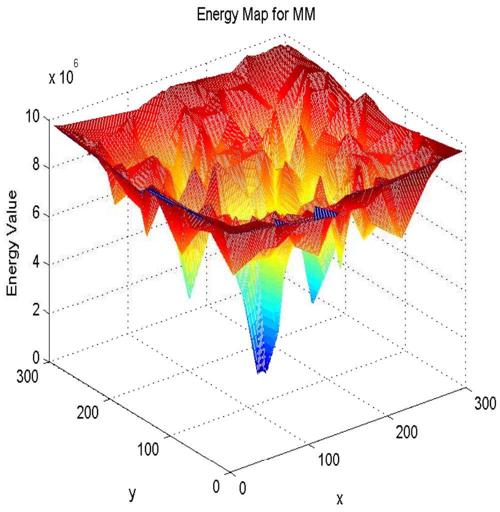


Figure 4.9: Energy map of MM approach after first node death

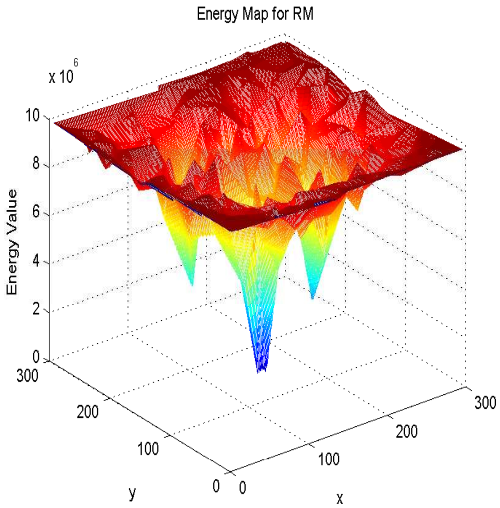


Figure 4.10: Energy map of RM approach after first node death

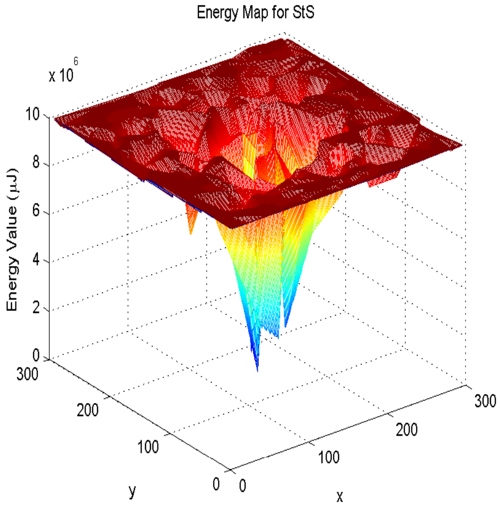


Figure 4.11: Energy map of StS approach after first node death

that has 400 nodes deployed to 300x300 m. area with a transmission range of $t_x = 35$ m. Sink sites are determined using site packing heuristic (SPH). Max-min approach performs better than random movement. When d_{max} value increases, network lifetime of MM consistently increases since there will be more options in that case. However, for RM, mo pattern can be observed. Since RM chooses next site randomly, it would be possible for it to perform better network lifetime with less number of possible sink sites (it performs better when $d_{max} = 200$ than $d_{max} = 300$ case). As expected, there is no any difference for static sink case, since sink does not move to any point in the area.

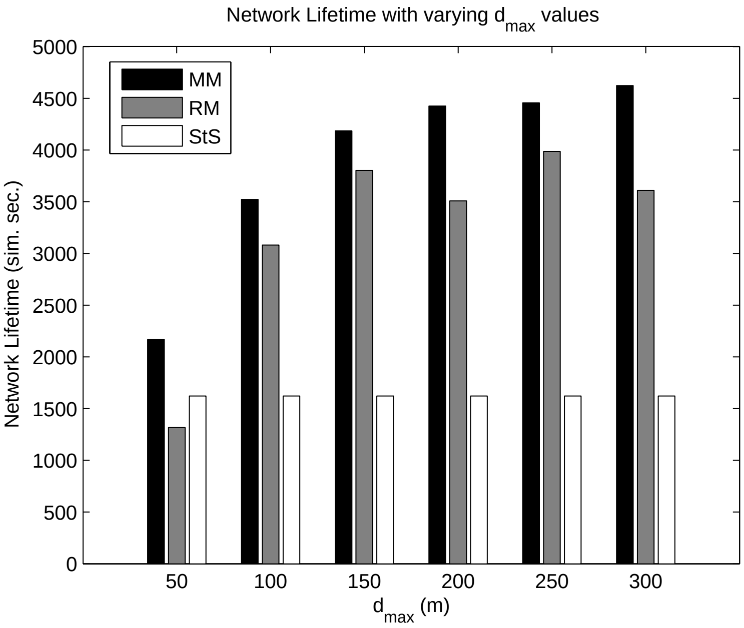


Figure 4.12: Network lifetime with varying d_{max} values (SSD = DSH and $t_x = 35m$).

4.2.5 Sink Stays a Minimum Amount of Time

In the previous parts, sink sojourn times expires when the energy of one of the neighbors of the sink decreases one level among L number of levels. However, it should be necessary for the sink to stay minimum amount of time in the current site for some reasons like ensuring data quality (necessary samples for calculating

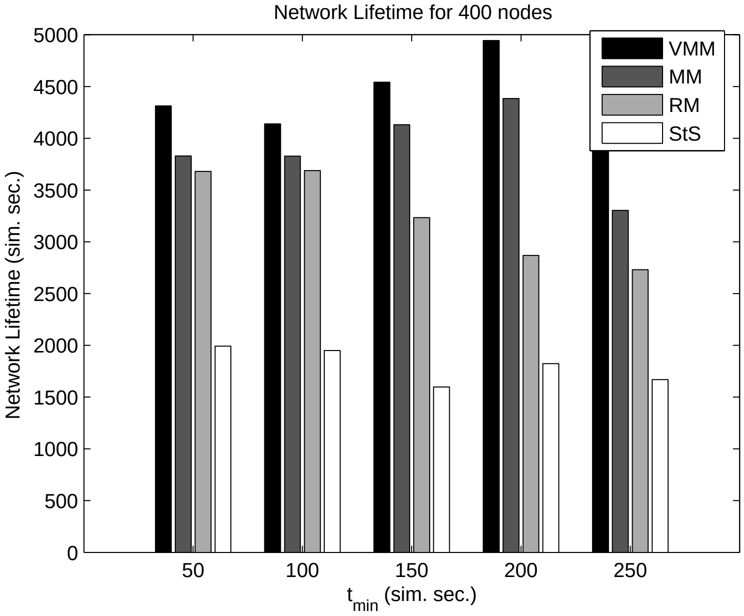


Figure 4.13: Network lifetime for 400 nodes (SSD = DSH and $t_x = 30m$).

average temperature for that local area), gathering some data before movement (no latency is tolerated), and so on. In that case, t_{min} is introduced as the minimum time that the base station have to stay at the current site. After this time expires, sink controls whether one energy level change has occurred or not. If this is so, the sink decides to move, otherwise it stays there for the next decision time arrives. With t_{min} value, it is possible to observe the effect of sink mobility trend in the network. For small values of t_{min}, sink becomes highly mobile, whereas for larger values of t_{min} it tends to stay longer on a site and demonstrates low mobility pattern. Fig. 4.13 shows the results of different approaches under varying t_{min} values from 50 to 250 simulation seconds. 400 nodes are randomly deployed to an area of 300x300 and tx value is 30 m. The figure shows that VMM performs better than all other approaches. Obviously there is a tradeoff between topology reconstruction cost and low sink mobility. For highly mobile sinks, the topology reconstruction cost reduces the advantage of mobility. For low mobility case, topology reconstruction cost reduces, however sink

4.3 Different Network Topology Construction Mechanisms Experiments

In this section, two different topology construction algorithms are compared with each other in terms of network lifetime and data latency. First one uses simple broadcast mechanism, and the second one uses the load balanced approach, the algorithm that is explained in Section 3.4. In Fig. 4.14, different number of nodes are deployed randomly to the area of 100x100 m with a transmission radius of 15m. As can be seen from the figure, when number of nodes increases, load balanced algorithm performs much better (100 %) than simple broadcast mechanism. Fig. 4.16 shows the same comparison for varying transmission range values when 500 nodes are deployed to an area of 300x300 m. It shows the similar behavior to the previous one, that is, difference between two approaches become larger when t_x value increases. Fig. 4.15 shows the data latency comparison. Although it achieves nearly two times bigger network lifetime for some cases, it just only has 2.6% of bigger average hop count value at most (this is intuitive since load balanced topology algorithm aims to distribute the load as uniform as possible instead of using the shortest paths). That means balanced tree topology construction approach significantly improves network lifetime, and causes very low data latency when doing that.

After examining different parts of the scheme, it would be reasonable here to see the overall performance of the proposed algorithms together. In this experiment, we compare two different mobility scheme with different properties. First one uses coordinate based sink site determination algorithm, VMM and balanced tree based construction algorithm for topology generation. Second one uses grid based sink site determination algorithm, random movement (RM) and simple broadcast mechanism for topology construction. In the experiment, varying number of nodes are deployed to an area of 300x300 with a transmission range of 30m. As it can be seen from Fig. 4.17, network lifetime difference between these two approaches increases when number of nodes increases. VMM approach performs up to 3.5 times better than random movement case, even RMM is also a mobility scheme. This brings important improvement to network lifetime when

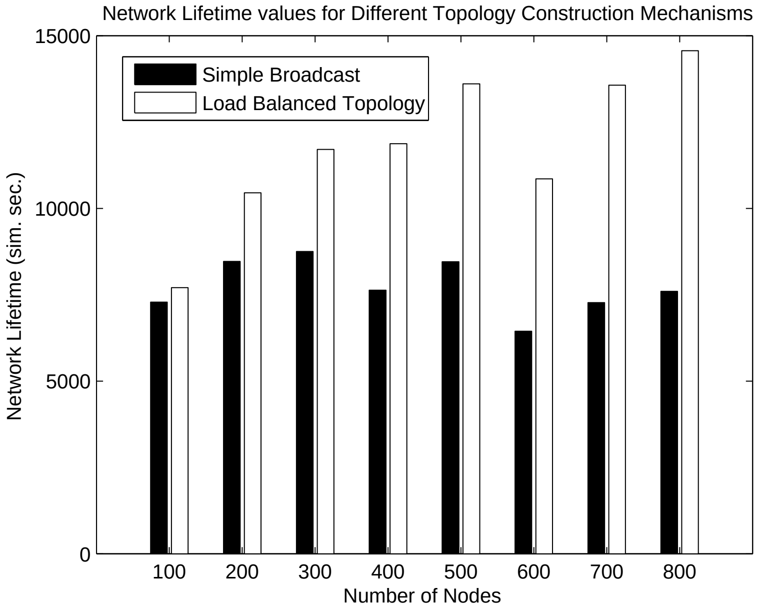


Figure 4.14: Network lifetime for different topology construction mechanisms (area side = 100m and $t_x = 15m$).

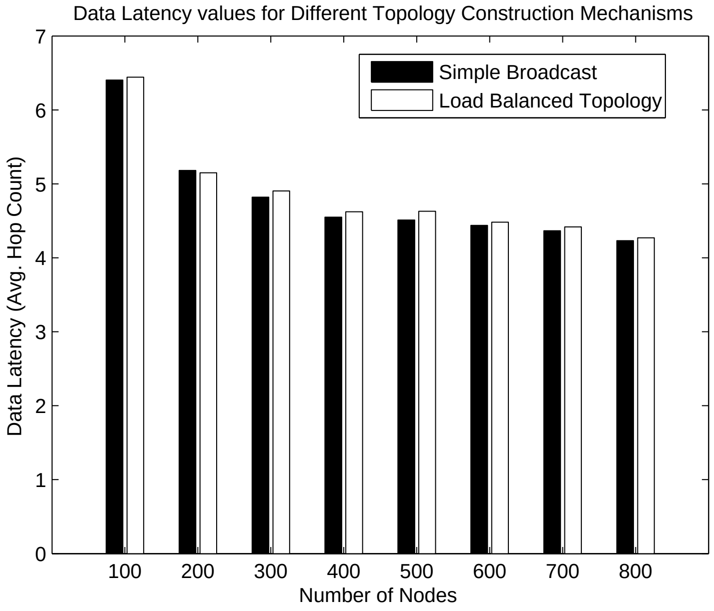


Figure 4.15: Average hop count values for different topology construction mechanisms (area side = 100m and $t_x = 15m$).

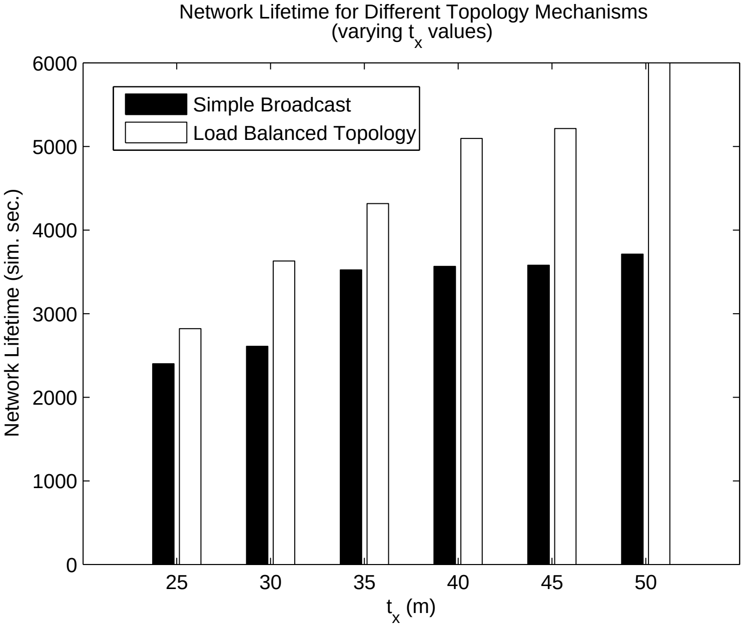


Figure 4.16: network lifetime for different topology construction mechanisms (area side = 300m and number of nodes = 500).

using different components of the scheme together.

4.4 Different Network Topology Update Mechanisms Experiments

The main drawback of sink movement is its topology reconstruction cost. Topology is fully reconstructed from up to bottom starting by the sink nodes's first broadcasting message after each sink movement. Although, all nodes participate this construction event, it should be useless to rebroadcast the message to the medium after a point, if required nodes change their parents. Local tree update mechanism reduces number of message exchanges by providing only a small percentage of nodes participation.

Three different experiments are done in order to see the effect of the local tree update mechanism. Different number of nodes are deployed to area of 300x300 m where transmission range is 25 m. Sink stays one round in each candidate site

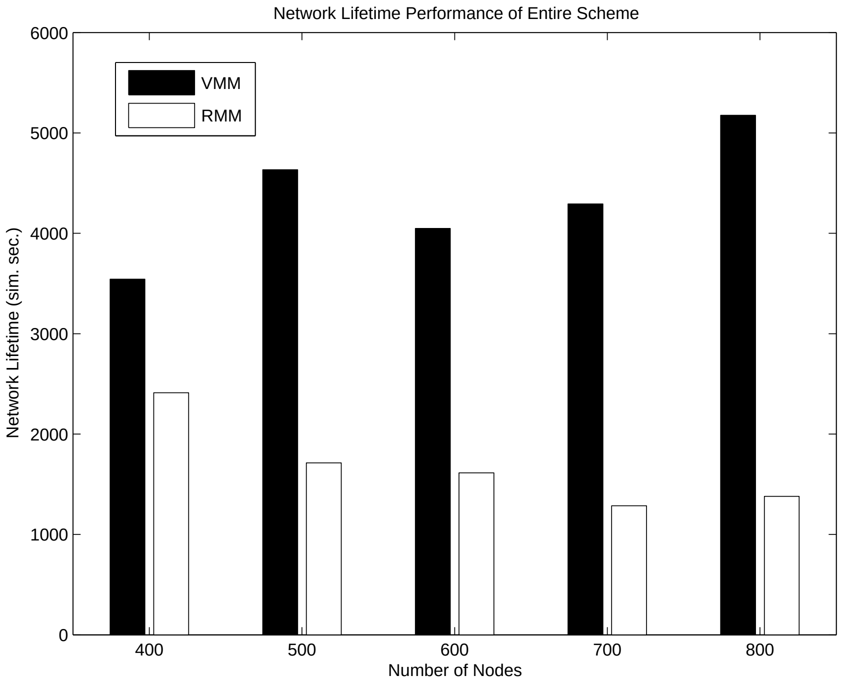


Figure 4.17: Network lifetime performance of entire scheme (area side = 300m and $t_x = 30m$).

and leaves there for the next round.

In Figure 4.18, percentage of participating nodes, i.e. the percent of nodes that broadcast message to the medium for two topology construction approaches, is examined. As it can be seen in the figure, on the average nearly 20% of all nodes participate for topology update in the sink's new site. This means that control packet ratio in the network is reduced around 80%. This energy can be used for data packet transmissions.

There is a trade-off between optimal topology and topology construction cost. When a local update mechanism is used, generally the resulting topology become a non-optimal one according to the topology that comes from full topology reconstruction (has longer paths). This means that average hop count of a packet to the sink increases. This can be seen in the Fig. 4.19. The difference between the two approaches increases when the number of nodes increases (1.07 to 1.35). This situation causes a packet to spend more energy (7% to 35%) in its travel to the sink.

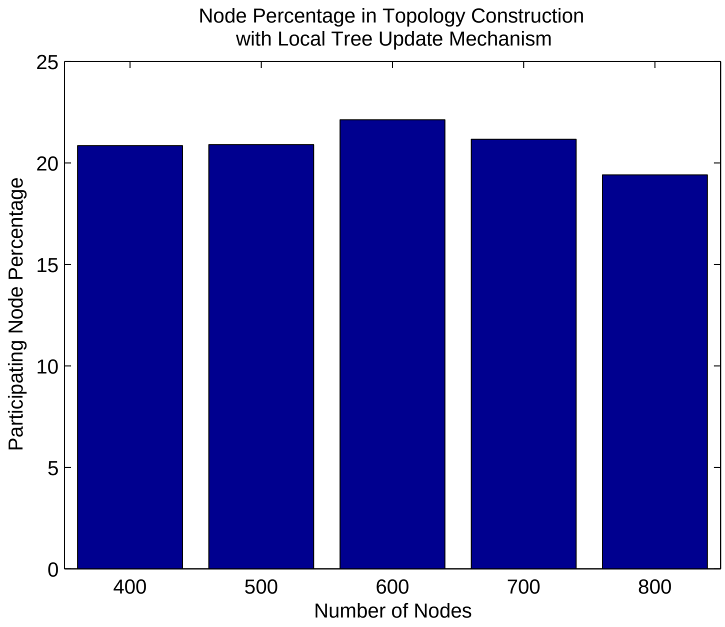


Figure 4.18: Participating Node Percentage in Topology Construction with Local Tree Update Mechanism

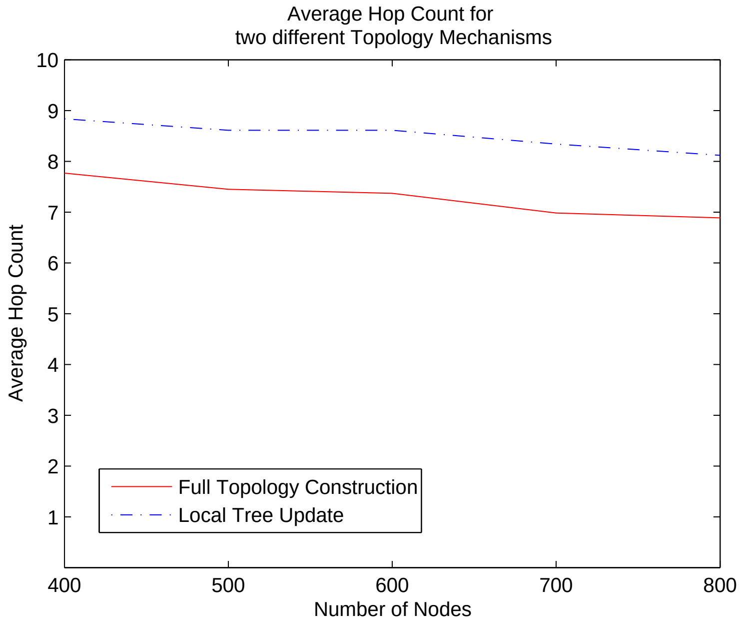


Figure 4.19: Average Hop Count to Sink Values for Different Topology Update Mechanisms

The last figure of this section, Fig. 4.20, shows the energy result of the two approaches. Although local tree update mechanism has higher average hop count and spends more energy for a data packet to be reached to the sink, the total energy in the network when the first node dies is greater than that of full tree reconstruction approach. For a situation where network total energy is important (applications that tolerate node failures to some threshold), this mechanism can contribute to the network operations.

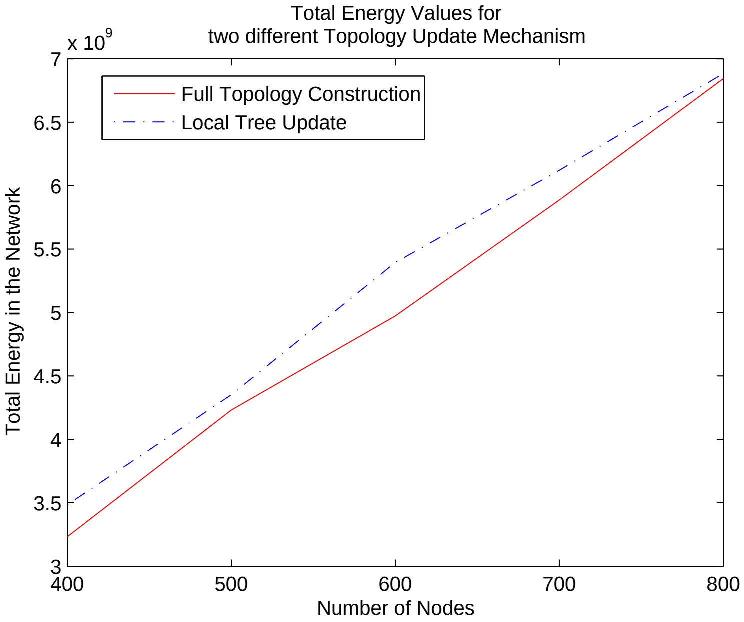


Figure 4.20: Total remaining energy in the network when first node dies

In this chapter, we examine the performance of our algorithms with different approaches. Before giving the performance results, we define the system model and the main simulation parameters that are used in the experiments. Firstly, we examine the performance of our sink site determination approaches with other four approaches in the literature by fixing all the other variables in the system (sojourn time, movement criterion). The results show that, our approaches perform better than the other all four. They have lower data latency values than other three approaches, but higher value than the approach in which center of the four grids are used as sink sites.

Maximum of the minimums (max-min/MM) and visit-added MM (VMM) perform better than RM (random movement) and static sink (StS) approaches for both without and with d_{max} parameter. Performance of the MM increases when d_{max} increases, however RM does not have such a pattern. VMM and MM also perform better, when sink has to stay a minimum amount of time (t_{min}), when it goes to a site.

Our energy efficient topology construction algorithm performs better than simple broadcast mechanism for both network lifetime (varying number of nodes and transmission ranges) and data latency values. This makes it an alternative to be used in cases where sink is mobile.

Finally, our local tree update mechanism reduces the number of message exchanges (80%) by sacrificing the data latency. Although the network lifetime value is worse than the simple broadcast mechanism, the total energy in the network is better than that of its counterpart.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we investigated controlled sink mobility problem in order to improve network lifetime in wireless sensor networks. Since sink's neighbors deplete their energy faster than the other nodes in a network where data aggregation is not used, controlled sink mobility aims to improve lifetime via changing these nodes (its neighbors) periodically and distributing the packet load among those nodes fairly by changing its position using network parameters (energy level of the nodes, node density, so on).

We dealt with different components of the sink mobility problem. First, we proposed two different sink site determination algorithm which use neighborhood relationship and coordinate of the nodes as inputs. To the best of our knowledge, neighborhood based sink site determination algorithm is the first one, in which coordinates of the nodes and/or boundaries of the area are not needed to be known. Unlike using predefined time or round values, sojourn time is also determined using a dynamic approach. Energy of the nodes are divided to L levels and sojourn time of the sink at a site expires when one of its neighbors spends its $\frac{1}{L}$ energy. In order to decide to choose next sink site, sink collects the minimum energy value of the nodes in each sink site, and goes to the site which has the node that has the maximum of this list (max-min/MM). A modified version of this max-min approach (VMM) is proposed, in which sink does not visit a site again, until it will visit all the sites in energy efficient order. Unlike the previous works which used linear programming, there is no scalability problem in our case.

Moreover, balanced tree topology construction algorithm is proposed instead of using simple broadcast mechanism. In this algorithm, current energy levels of the nodes are taken into consideration and packet loads are distributed from bottom to up, using this information. Finally, a local tree update mechanism is presented in order to reduce control packets in the network during topology reconstruction.

We compared the performance of our algorithms with different approaches via simulation experiments. Our sink site determination algorithms perform better than all the other four approaches in the literature. They have also lower data latency values than other three of the four approaches. In sink mobility experiments, MM and VMM give better results than RM (random movement) and static sink (StS) approaches for all cases including without and with d_{max} parameter and sink has to stay a minimum amount of time (t_{min}), when it goes to a site. Our energy efficient topology construction algorithm performs better than simple broadcast mechanism for both network lifetime (varying number of nodes and transmission ranges) and data latency values. Finally, our local tree update mechanism reduces the number of message exchanges (80%) and has higher total energy value in the network than simple broadcast mechanism by sacrificing the data latency.

5.2 Future Work

Although different parts of the sink mobility problem has been investigated in this study, there are still many points to be discussed. More detailed (case specific) and complete solution set can be given by considering these points.

In this thesis, nodes are randomly and uniformly deployed to the area. Different deployment strategies (grid based deployment, custom deployments where some sort of skewness added and so on) should also be tested and evaluated. Algorithms can be modified or extended in order to give better results for those deployment strategies. Network lifetime definition is another point to be diversified. It is defined as the time passed until first node dies, which is also a common definition in the literature. There are also another definitions in the literature (like time until total number of messages received decreases under a threshold value) which can be used and tested in this context. Especially movement criterion can be modified in order to give better results for such a definition. For instance instead of using the maximum of minimums, different functions (like *average*) can be used for getting better performance.

In both neighborhood and coordinate based sink site determination algorithms, the points are determined before network starts to operate and then movement between those sites occurs according to the current energy map of the network. Although it will increase the decision time, the performance of determining the sink sites while network is operating should also be tested and evaluated.

In this thesis, it is always assumed that sensor nodes have equal initial energy values. However, there can be heterogeneity in the network such that some nodes can have better computational and energy values than the others. So the algorithms can be extended to adopt such a case. For instance, in sink site determination algorithms, possible sites can be chosen from the points where the density of those powerful nodes is higher.

Topology reconstruction cost is the most important disadvantage of sink mobility scheme. Reducing this cost as much as possible is going to improve the network lifetime. Although a simple and intuitive algorithm is given about local tree topology update mechanism, it has higher data latency values than its counterpart which increases the cost of transmitting a packet to the sink node. This causes worse network lifetime values than full topology reconstruction case. That's why, more intelligent algorithm can be developed that yields better data latency values and gives better network lifetime values than fully reconstructed topologies especially in highly mobile environments (for instance where sink moves at every round).

The results of the experiments (especially ones regarding to network lifetime) can also be compared with an optimal solution in the literature. Since execution time of linear programming solutions dramatically increases with more number of parameters, unrealistic assumptions can be made in order to reduce this decision time. That's why, it is important to make sure that the assumptions in both works are same when doing such a comparison.

Bibliography

[1] I. F. Akyildiz, M. C. Vuran, O. B. Akan, and W. Su. Wireless sensor networks: A survey revisited. *Computer Networks Journal (Elsevier)*, 2005.

[2] S. Alam and Z. Haas. Topology control and network lifetime in three-dimensional wireless sensor networks. In *CoRR abs/cs/0609047*, 2006.

[3] S. Basagni, A. Carosi, E.Melachrinoudis, C. Petrioli, and M. Wang. Controlled sink mobility for prolonging wireless sensor networks lifetime. *ACM journal on Wireless Networks (WINET)*, 2007.

[4] S. Basagni, A. Carosi, and C. Petrioli. Controlled vs. uncontrolled mobility in wireless sensor networks: Some performance insights. In *Vehicular Technology Conference*, 2007.

[5] R. Bhatia and M. Kodialam. On power efficient communication over multi-hop wireless networks: Joint routing, scheduling and power control. *IEEE INFOCOM*, 2, 2004.

[6] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine*, 7(5), 2000.

[7] E. Bulut and I. Korpeoglu. Zone based data aggregation scheduling scheme for maximizing network lifetime. In *International Workshop on Heterogeneous Wireless Networks*, 2007.

[8] S. Butenko, R. Murphey, and P. Pardalos. *Recent Developments in Cooperative Control and Optimization.* Springer, 2003.

[9] M. Cardei and D. Du. Improving wireless sensor network lifetime through power aware organization. *Wireless Networks, Volume 11, Issue 3*, 2005.

[10] M. Cardei, W. Jie, L. Mingming, and M. Pervaiz. Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. *Wireless And Mobile Computing, Networking And Communications*, 2005.

[11] J. Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(4), 2004.

[12] T. H. Cormen, C. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, 2nd Edition.* The MIT Press, 2001.

[13] Crossbow. Mpr-mib wireless module users manual. *available at http://www.xbow.com/Support/Support_pdf_files/MPR-MIB_Series_Users_Manual.pdf*, July 2008.

[14] T. Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *The First ACM Conference on Embedded Networked Sensor Systems (Sensys 03)*, 2003.

[15] K. Dasgupta, K. Kalpakis, and P. Namjoshi. Improving the lifetime of sensor networks via intelligent selection of data aggregation trees. In *Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2003.

[16] I. Demirkol, C. Ersoy, and F. Alagoz. Mac protocols for wireless sensor networks: A survey. *IEEE Communications Magazine*, 44(4), 2006.

[17] G. Ferrari and M. Martal. Extending the lifetime of sensor networks through adaptive reclustering. *EURASIP Journal on Wireless Communications and Networking*, 2007.

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

[19] Y. Gu, H. Liu, F. Song, and B. Zhao. Joint sink mobility and data diffusion for lifetime optimization in wireless sensor networks. In *The 2nd IEEE Asia-Pacific Service Computing Conference*, 2007.

[20] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. *Proceedings of the 33rd International Conference on System Sciences (HICSS '00)*, 2000.

[21] T. T. Huynh and C. S. Hong. Prolonging network lifetime via intra-cluster routing in wireless sensor networks. In *Second International Conference on Mobile Computing and Ubiquitous Networking*, 2005.

[22] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy. Exploiting mobility for energy efficient data collection in wireless sensor networks. *Mobile Networks and Applications*, 11(3), 2006.

[23] I. Kong, J. Kim, and W. Hwang. Lifetime maximization by cross-layer design considering power control, mac, routing in sensor networks. *International Journal of Computer Science and Society*, 6(9), 2006.

[24] H. Kwon, T. Kim, S. Choi, and B. G. Lee. Cross-layer lifetime maximization under reliability and stability constraints in wireless sensor networks. *International Journal of Computer Science and Society*, 6(9), 2006.

[25] S. Lee, K. Han, K. Lim, and S. An. Cross-layer design based data aggregation for maximizing network lifetime. In *International Conference on parallel and distributed computing and networks*, 2007.

[26] S. Lee, K. Han, K. Lim, J. Lee, and S. An. Zone based data aggregation scheduling scheme for maximizing network lifetime. In *The Fifth International Symposium on Parallel and Distributed Processing and Applications*, 2007.

[27] S. Lee, Y. Jung, W. Park, and S. An. Spatial correlation code based data aggregation scheme for maximizing network lifetime. *Springer Network-Based Information Systems (LNCS)*, 4658, 2007.

[28] W. Liang and Y. Liu. Online data gathering for maximizing network lifetime in sensor networks. *IEEE Transactions on Mobile Computing*, 6(1), 2007.

[29] T. Lim and G. Mohan. Energy aware geographical routing and topology control to improve network lifetime in wireless sensor networks. In *2nd International Conference on Broadband Networks*, 2005.

[30] G. Lu, B. Krishnamachari, and C. S. Raghavendra. An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks. In *18th International Parallel and Distributed Processing Symposium*, 2004.

[31] J. Luo and J. P. Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. *IEEE INFOCOM*, 2005.

[32] R. Madan, C. Shuguang, S. Lail, and A. Goldsmith. Cross-layer design for lifetime maximization in interference-limited wireless sensor networks. *IEEE INFOCOM*, 3, 2005.

[33] Y. Nam, H. Lee, H. Yung, T. Kwon, and Y. Choi. An adaptive mac (a-mac) protocol guaranteeing network lifetime for wireless sensor networks. In *European Wireless Conference*, 2005.

[34] I. Papadimitriou and L. Georgiadis. Energy-aware routing to maximize lifetime in wireless sensor networks with mobile sink. *Journal of Communications Software and Systems*, 2006.

[35] J. Park and S. Sahni. An online heuristic for maximum lifetime routing in wireless sensor networks. *IEEE Transactions on Computers*, 55(8), 2006.

[36] C. Perkins. *Ad Hoc Networks*. Addison-Wesley, Reading, MA, 2000.

[37] Y. Qian, J. Zhou, L. Qian, and K. Chen. Prolonging the lifetime of wireless sensor network via multihop clustering. *Springer Next Generation Teletraffic and Wired/Wireless Advanced Networking (LNCS)*, 4003, 2006.

[38] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. In *ACM SenSys*, 2003.

[39] S. Ray, W. Lai, and I. C. Paschalidis. Statistical location detection with sensor networks. *IEEE/ACM Transactions on Networking*, 14(SI), 2006.

[40] A. Savvides, C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *7th Annual International Conference on Mobile Computing and Networking*, 2001.

[41] C. Schurgers and M. Srivastava. Energy efficient routing in wireless sensor networks. In *MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force*, 2001.

[42] R. Shah and J. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2002.

[43] R. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *First IEEE International Workshop on Sensor Network Protocols and Applications*, 2003.

[44] S. S. Skiena. *The Algorithm Design Manual, 2nd Edition.* Springer, 2008.

[45] R. Subramanian and F. Fekri. Sleep scheduling and lifetime maximization in sensor networks: fundamental limits and optimal solutions. In *The Fifth International Conference on Information Processing in Sensor Networks*, 2006.

[46] L. Tong, Q. Zhao, and S. Adireddy. Sensor networks with mobile agents. In *IEEE MILCOM*, 2003.

[47] D. Vass and A. Vidács. Positioning mobile base station to prolong wireless sensor network lifetime. In *CoNEXT Student Workshop*, 2005.

[48] P. Venkitasubramaniam, S. Adireddy, and L. Tong. Sensor networks with mobile agents: Optimal random access and coding. *IEEE Journal on Selected Areas in Communications*, 22(6), 2004.

[49] A. Verma, H. Sawant, and J. Tan. Selection and navigation of mobile sensor nodes using a sensor network. In *Third IEEE International Conference on Pervasive Computing and Communications*, 2005.

[50] L. Wang and Y. Xiao. A survey of energy-efficient scheduling mechanisms in sensor networks. *Mobile Network Applications*, 11(5), 2006.

[51] N. Wang and C. Chang. Probabilistic-based energy aware routing for network lifetime and source level fairness in wireless sensor networks. In *International Conference on Systems and Networks Communications*, 2006.

[52] S. Wang, M. Yang, J. Wang, Y. Jiang, and J. Jo. Mobile node deployment in hybrid sensor networks. In *Second International Conference on Communications and Networking*, 2007.

[53] L. A. Wolsey. *Integer Programming.* Wiley-Interscience, 1998.

[54] X. Wu, J. Cho, B. J. d'Auriol, S. Lee, and Y. Lee. An integrated sleep-scheduling and routing algorithm in ubiquitous sensor networks based on ahp. *IEICE Transactions on Communications*, E90-B(12), 2007.

[55] Y. Zongkai, Z. Dasheng, C. Wenqing, and H. Jianhua. Energy efficient routing with power management to increase network lifetime in sensor networks. *Computational Science and Its Applications - ICCSA*, 2004.

[56] Y. Zongkai, Z. Dasheng, C. Wenqing, and H. Jianhua. Prolong the lifetime of wireless sensor network with dichotomy power control. In *IEEE TENCON*, 2004.

[57] S. Zou, I. Nikolaidis, and J. Harms. Extending sensor network lifetime via first hop data aggregation. In *IEEE International Performance Computing and Communications Conference*, 2006.