ROUTING AND SCHEDULING APPROACHES FOR ENERGY-EFFICIENT DATA GATHERING IN WIRELESS SENSOR NETWORKS

A DISSERTATION SUBMITTED TO

THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Hüseyin Özgür TAN

September, 2011

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

Assoc. 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 dissertation for the degree of doctor of philosophy.

Prof. Dr. Özgur Ulusoy

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

Prof. Dr. Adnan Yazıcı

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

Assoc. Prof. Dr. Ezhan Karaşan

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

Assoc. Prof. Dr. Uğur Güdükbay

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

ROUTING AND SCHEDULING APPROACHES FOR ENERGY-EFFICIENT DATA GATHERING IN WIRELESS SENSOR NETWORKS

Hüseyin Özgür TAN
Ph.D. in Computer Engineering
Supervisor: Assoc. Prof. Dr. İbrahim Körpeoğlu
September, 2011

A wireless sensor network consists of nodes which are capable of sensing an environment and wirelessly communicating with each other to gather the sensed data to a central location. Besides the advantages for many applications, having very limited irreplaceable energy resources is an important shortcoming of the wireless sensor networks. In this thesis, we present effective routing and node scheduling solutions to improve network lifetime in wireless sensor networks for data gathering applications. Towards this goal, we first investigate the network lifetime problem by developing a theoretical model which assumes perfect data aggregation and power-control capability for the nodes; and we derive an upper-bound on the functional lifetime of a sensor network. Then we propose a routing protocol to improve network lifetime close to this upper-bound on some certain conditions. Our proposed routing protocol, called L-PEDAP, is based on constructing localized, self-organizing, robust and power-aware data aggregation trees. We also propose a node scheduling protocol that can work with our routing protocol together to improve network lifetime further. Our node scheduling protocol, called PENS, keeps an optimal number of nodes active to achieve minimum energy consumption in a round, and puts the remaining nodes into sleep mode for a while. Under some conditions, the optimum number can be greater than the minimum number of nodes required to cover an area. We also derive the conditions under which keeping more nodes alive can be more energy efficient. The extensive simulation experiments we performed to evaluate our PEDAP and PENS protocols show that they can be effective methods to improve wireless sensor network lifetime for data gathering applications where nodes have power-control capability and where perfect data aggregation can be used.

Keywords: Sensor Networks, Data Aggregation, Routing, Node Scheduling.

ÖZET

KABLOSUZ ALGILAYICI AĞLARINDA ENERJİ-VERİMLİ VERİ YIĞIŞIMI İÇİN YOL ATAMA VE ZAMAN PLANLAMA YÖNTEMLERİ

Hüseyin Özgür TAN
Bilgisayar Mühendisliği, Doktora
Tez Yöneticisi: Doç. Dr. İbrahim Körpeoğlu
Eylül, 2011

Kablosuz algılayıcı ağları bir ortamı algılayabilen, ve ölçülen verileri merkezi bir konuma gönderebilmek için birbirleri ile kablosuz şekilde iletişim kurabilen düğümlerden oluşur. Bir çok alandaki uygulamalar için sunduğu avantajlarının yanısıra kısıtlı ve değiştirilemez enerji kaynaklarına sahip olmak kablosuz algılayıcı ağlarının önemli bir yetersizliğidir. Bu tezde, veri toplama uygulamaları çalıştıran kablosuz algılayıcı ağlarının ağ ömrünü iyileştirmek için etkili yol atama ve zaman planlama çözümleri sunulmuştur. Bu amaçla, öncelikle ağ ömrü problemi, tam veri yığışımı ve düğümler için güç ayarlayabilme yeteneğini göz önünde bulunduran teorik bir model oluşturarak incelenmiş; ve bir algılayıcı ağının fonksiyonel ömrü için bir üst sınır türetilmiştir. Daha sonra, ağ ömrünü bazı koşullarda bu teorik üst sınıra kadar iyileştiren bir yol atama protokolü önerilmiştir. L-PEDAP adındaki önerdiğimiz algoritma; yerelleştirilmiş, kendini örgütleyebilen, stabil, ve güç-farkında veri yığışım ağaçlarının oluşturulması esasına dayanmaktadır. Bununla birlikte, ağ ömrünü daha da iyileştirmek için yol atama protokolümüz ile beraber çalışabilen bir zaman planlama protokolü de önerilmiştir. PENS adını verdiğimiz bu zaman planlama protokolü, bir turda en az enerji harcanmasını sağlayacak en uygun sayıda düğümü açık tutar; ve geri kalan düğümleri uyku moduna alır. Bazı koşullarda, en uygun düğüm sayısı, tüm alanı kapsamak için gerekli en az sayıda düğüm miktarından fazla olabilir. Bu kapsamda, daha fazla düğümü açık tutmanın enerji açısından daha verimli olabileceği şartlar türetilmiştir. Önerdiğimiz PEDAP ve PENS protokollerini değerlendirmek için yapmış olduğumuz kapsamlı simulasyonlar, bu yöntemlerin düğümlerin güç ayarlama yeteneğine sahip olduğu ve tam veri yığışımının kullanılabildiği veri toplama uygulamaları için etkili olduğunu göstermiştir.

Anahtar sözcükler: Algılayıcı Ağları, Veri Yığışımı, Yol Atama, Zaman Planlama.

Acknowledgement

I would like to express my gratitude to my supervisor Assoc. Prof. Dr. İbrahim Körpeoğlu for his support, guidance, encouragement, patience and instructive comments in the supervision of the thesis.

I would like to express my thanks and gratitude to Prof. Dr. Özgür Ulusoy, Prof. Dr. Adnan Yazıcı, Assoc. Prof. Dr. Ezhan Karaşan, Assoc. Prof. Dr. Uğur Güdükbay and Asst. Prof. Dr. Ali Aydın Selçuk for evaluating my thesis.

I am also grateful to my managers and colleagues in HAVELSAN for their understanding and tolerance during my research.

I should also express my special thanks to Prof. Dr. Ivan Stojmenovic for his guidance while preparing my papers, which are critical to earn my PhD degree.

My sincere thanks goes to my family for their love, support and motivation.

Above all, I should express my deepest gratitude to my wife for her endless love, patience and support throughout my PhD study. She has been standing by me in the most difficult times of my study, and I should admit that I could not achieve this without her patience and encouragement in these times.

To my beloved wife …

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With recent developments in micro-electro-mechanical-systems (MEMS) it is possible to build low cost, low power, tiny sensor nodes. These sensors can be used to collect information from an area of interest. Each sensor node has a processor, memory, and wireless communication module, besides having various sensors. These tiny sensor nodes are designed to replace their macrosensor counterparts. However, unlike their powerful equivalents, these nodes have very limited capabilities. On their own, they cannot compete with their macrosensor equivalents; but by using hundreds or thousands of them, it is possible to build a low cost, high quality, fault tolerant sensing system. Since these microsensor nodes can communicate with each other by using their wireless modules, they can form a network and the data sensed by individual nodes can be gathered and processed at a center to obtain a high quality signal or highly useful information. A network of these sensor nodes is called a *wireless sensor network*.

There are several advantages of sensor networks over the expensive equivalent systems. First of all, a sensor node is designed to be very inexpensive. The cost of one sensor node is planned to be under 1$. Secondly, the nodes can operate in harsh environments such as deep in the oceans, up in the volcanic mountains or on the battlefields. Finally, since they have wireless modules and can communicate with each other, they can improve the quality of the data by sensing the same event from different viewpoints and combining these data by techniques like data fusion.

Besides the advantages of sensor networks there are some disadvantages. The main problem with the nodes is their limited capabilities. The nodes usually have inadequate resources, such as a low speed microprocessor and a low capacity memory in the order of kilobytes [1]. Fortunately, the nodes are not responsible for tasks that require large amount of processing power and memory. They usually sense simple data and after optionally processing the data, send it to a more powerful base station where complex operations can be performed. However, the main shortcoming of these nodes is their limited power supply. They usually have very small battery and usually their batteries cannot be replaced or recharged because of the harsh environmental conditions and huge number of sensor nodes.

At first glance, wireless sensor networks seem very similar to classical wireless networks. In both of them there are wireless-enabled nodes and the data must be efficiently moved. However, there are some subtle differences between them. Firstly, usually the sensor nodes are stationary, whereas in classical wireless networks mobility of the nodes is common and is a main concern. Secondly, the bulk of the data flow is usually from sensor nodes to a central base station which exhibits all-to-one communication pattern. On the other hand, in classical wireless networks since all the nodes are powerful, they can be both source and the destination of information. Finally, the most important difference is the power supplies of the nodes. In classical wireless networks such as GSM or wireless ad-hoc networks the batteries of the nodes are usually rechargeable or at least replaceable. Therefore, in the design of classical wireless networks, energy consumption is important but is usually not the most critical issue. In wireless sensor networks, however, the main design goal is to effectively and efficiently use and manage the energy resources so that the lifetime of the network is extended as much as possible. Also the design issues such as throughput, latency or quality of service (QoS) requirements are not so important for sensor networks [51]. All these points make the design of sensor networks much different than the design of classical wireless networks and all the unique constraints and features of sensor networks make the design of data communication protocols for sensor networks a challenging task [77].

In a typical sensor network application, nodes are deployed randomly in an area of interest (for instance by dropping from an air-plane). After the deployment, the nodes begin to sense their nearby environment and send the collected information to a central base station using their wireless communication modules. The primary job of a sensor network is to sense/collect and gather data, and it is desirable to be able to do this for a long time. Hence, considering the limited energy resources, the main design issue in wireless sensor networks is to extend the lifetime of the network as much as possible.

A sensor network usually generates too much data for an end-user to process. The transmission of enormous amount of unnecessary data in the system also results in performance degradation. Because of this, methods for combining, filtering, processing data into a small set of meaningful information are required. A simple way of doing that is aggregating (sum, average, min, max, count) the data originating from different nodes. A more complex method is data fusion which can be defined as combining several unreliable data measurements to produce a more accurate signal by enhancing the common signal and reducing the uncorrelated noise [26]. These approaches have been used by different protocols so far, because of the fact that they improve the performance of a sensor network in an order of magnitude by reducing the amount of data transmitted in the system. In all protocols proposed in this thesis, we assume perfect data aggregation, which means that combining n packets of size k results in one packet of size k instead of size nk. Hence our protocols will be useful for applications that allow perfect data aggregation at intermediate nodes.

Since the application areas of sensor networks become very wide from health to military, there exists great amount of work done on this topic [3]. Also with concurrent developments in MEMS technology, the usage of these sensing systems seems to multiply in future. Despite the large amount of work done on the topic so far, however, there are still many open issues and challenges in the design of sensor networks.

In this thesis we focus on improving functional system lifetime of sensor networks for data gathering applications. In the scope of this work, we started with a survey of the methodologies used in the literature for improving lifetime of sensor networks. The approaches proposed in literature can be categorized into five classes: data volume minimization, efficient topology construction, routing, sleep scheduling and mobility. We realized that the majority of the proposed approaches in the literature do not consider sensor nodes with power-control capability - i.e. capability of adjusting transmission power proportional to the desired distance. We also saw that many of the proposed protocols lack mathematical reasoning and solely depend on the simulation results. Another problem with the previous works was that they only focus on a specific approach and try to improve other protocols using the same approach. The results of our survey revealed the need for a theoretical model for evaluating the performance of a data gathering protocol and also the need for an hybrid solution which will incorporate different lifetime improvement approaches together.

In order to determine whether there will be a performance gain of using nodes with power control and using perfect aggregation in terms of functional system lifetime, we first tried to model such a network theoretically. Using this model we investigated the lifetime of the system mathematically and we characterized the maximum achievable lifetime (i.e. upper-bound for the lifetime) of a sensor network. We then worked on a data gathering solution that will get close to this upper-bound. By using the theoretical model, we have seen that a lot of routing and data gathering protocols are far from being close to the optimal lifetime.

To improve network lifetime as much as possible, we propose a new distributed routing protocol to gather data from sensor nodes to the center, which uses the advantage of power control and perfect aggregation. The main idea behind this protocol is to minimize the power consumption in a round, while balancing the load among the nodes. The results of our comprehensive simulations showed that our new protocol outperformed previous proposed methods in the literature.

Our model and simulations, however, shows that increasing the number of nodes in the system does not always help in improving the functional system lifetime regardless of the routing scheme used. Therefore, keeping the right number of sensor nodes active is very important for energy efficient operation. To decide on the right number of sensor nodes to be active, we propose a new sleep scheduling algorithm which also takes the advantage of power control and perfect aggregation. Different from the previous sleep scheduling algorithms, our algorithm tries to keep optimum number of nodes alive, instead of keeping minimum number of nodes alive. This is based on the observation that in some conditions energy can be saved by using more nodes because of the exponential cost of transmitting to far distances. In this part of the thesis, we derived mathematical formulations of such conditions, and we verified these formulations by running several simulations.

The rest of the thesis is organized as follows. In Chapter 2, we first present detailed information about sensor networks and then we present our problem statement in detail. We also specify our system model and assumptions in this chapter. In Chapter 3, we give related work about extending wireless sensor lifetime with a good categorization with respect to used methods. In Chapter 4, we provide a detailed lifetime analysis for wireless sensor network that can apply perfect data aggregation. We present and describe in detail our proposed power-efficient distributed routing solution in Chapter 5. We present our node scheduling solution Chapter 6. Finally, we give our conclusions and future work issues in Chapter 7.

Chapter 2

System Model and Problem Statement

In this chapter we first discuss some common application scenarios of sensor networks and briefly go over the energy consumption models used in sensor network research. We then give our sensor network model and formulate different lifetime definitions. Finally, we formulate the problem that we focus in this thesis and we present the details of the problem.

2.1 Applications of Sensor Networks

As mentioned in Chapter 1, the main idea behind the use of sensor networks is to deploy a large number of sensor nodes in an area of interest and collect useful information from that area. Since each node has wireless communication capability, the collected data can be forwarded hop-by-hop to one of the monitoring base stations. The base stations are usually not energy limited and can be connected to each other using a high performance wired or wireless network. The incoming data to a base station or a control center can be processed with a software and users can issue queries to get some specific information. In this way collecting data from all nodes to a center is converted to a useful information or alarms to the end users of the system.

Because they are inexpensive and can operate even in harsh environments where their macrosensor equivalents cannot be deployed, the sensor networks are preferred for a very wide range of applications, from military to civil [3]. Most of the applications of the sensor networks can be classified into two in terms of data collection strategy: *Event driven* and *demand driven* [13].

In event driven applications, sensor nodes are programmed to detect a specific event. Normally, there is no data flow in the network unless an event is detected. As soon as some of the nodes detect an event, they immediately report this information to the base station. A good example for this kind of application is fire detection systems. In event driven applications, the lifetime of the network can be defined in terms of number of events reported, since only source of energy consumption is detecting events.

In demand driven applications, sensor nodes remain silent until they receive a request from the base station. The base station usually asks the sensors for their data for a specific duration, and consequently all the sensors that receive the request send their collected data for the specified duration. Optionally, the query from the base station can specify the region of interest. In this case, only the sensors in that region are activated and the rest remain silent. Actually, the query must also specify the time period between two reporting events, which can also be specified as data-rate. If the time period is not specified, a predefined value can be used in order to synchronize the nodes. We define this time period as a *round.* That means, in each round, all sensor nodes sense and obtain their readins and these readings are transported to the base station over the sensor network. In demand driven applications, the lifetime can be defined in terms of rounds, which means the number of times the network can provide data to the base station.

A specific type of demand driven application is the one where all the nodes in network are required to report their data to the base station in each round. The data can be aggregated at intermediate nodes. An example application of this type can be an air conditioning system which decides to switch on the conditioners based on the average temperature of the field. In this special application, all the data sensed from the field must be periodically reported to the base station (possibly after aggregation). There is a significant body of work done on these types of applications [26, 30, 33, 39, 64, 74].

As the literature about sensor networks is examined, it can be seen that each application scenario has its own solutions, since the requirements of different applicatios can vary significantly. Therefore, it is very important for a protocol designer to specify application scenario first, with as much details as possible.

2.2 Energy Consumption Models

A sensor node consists of several components such as a processing unit, a wireless communication unit, and a sensing unit. All these components are sources of energy consumption. The rate of energy consumption of a component can vary according to the current activity level of the component. Sometimes a component can be even completely turned off for a while if it is not needed during that time. Managing when components will be on and off is also important for efficient energy consumption.

The key component of a sensor node is the sensing unit. Since the main responsibility of a sensor node is to *sense* the environment, it is usually not turned off. However, the node's role in a specific data gathering round can determine its state. If the node is decided to participate in the data collection operation, the sensing unit must be turned on. Although it is usually meaningless to keep the other units on when there exists no data collection, switching only the sensing unit off can make sense in the case where the node itself does not participate in data collection operation but is responsible for relaying other nodes' data towards the base station.

Moreover, for event driven applications, it is not possible to switch off the sensing unit in a node since it can not be known exactly in advance when an event will occur. On the other hand, in demand driven applications, if the duration and the interval of sensing is specified, the sensing unit can be turned on only at necessary time instants to sense data after which it can be turned off immediately until the beginning of the next time interval.

Hence one source of energy consumption in a sensor node is the sensing unit. It consumes energy when the sensors are on. If sensing is not needed for a while, the sensing unit can be turned off and energy can be saved in this way. We can assume that when the sensing unit is off it consumes no energy; and when it is on, the power consumed is constant E_{sense} (i.e. energy consumed is constant over a unit time interval).

The power dissipated by processing unit is mainly due to post-sense and post-receive operations. These operations may include the analog to digital conversion, aggregation of data, packet parsing, packet assembling, maintenance of in-memory tables, etc. Most of the time, too much processing intensive tasks are not executed at sensor nodes, therefore the energy consumed in the processing unit is usually much less compared to energy consumed in other components like the sensing unit or communication unit. Additionally, the processors used in sensor nodes are designed and selected to be very low power. We can consider the power consumption at the processing unit again to be constant ($E_{process}$). Most work in the sensor network literature ignores the energy consumption at the processing unit and we will do the same in this thesis.

The most significant power consumption happens at the wireless communication unit when it is active. The communication unit can be in one of the following four states: *transmit, receive, idle listening* and *sleep.* The energy costs of these states can easily be understood with the first order radio model presented in [26] (see Figure 2.1). In this model, in order to transmit a k-bit packet to a distance d, the packet must first be processed by the transmit electronics to generate the output signal, and then the output signal must be amplified in order to reach to a distance d. The model expresses the energy consumption per packet in transmit electronics and transmit amplifier as $E_{tr-elec} \times k$ and $E_{amp} \times k \times d^{\alpha}$ respectively, where α is path loss exponent that depends on the environment (it is usually a value between 2 and 6). In order to receive a k-bit packet, the signal is

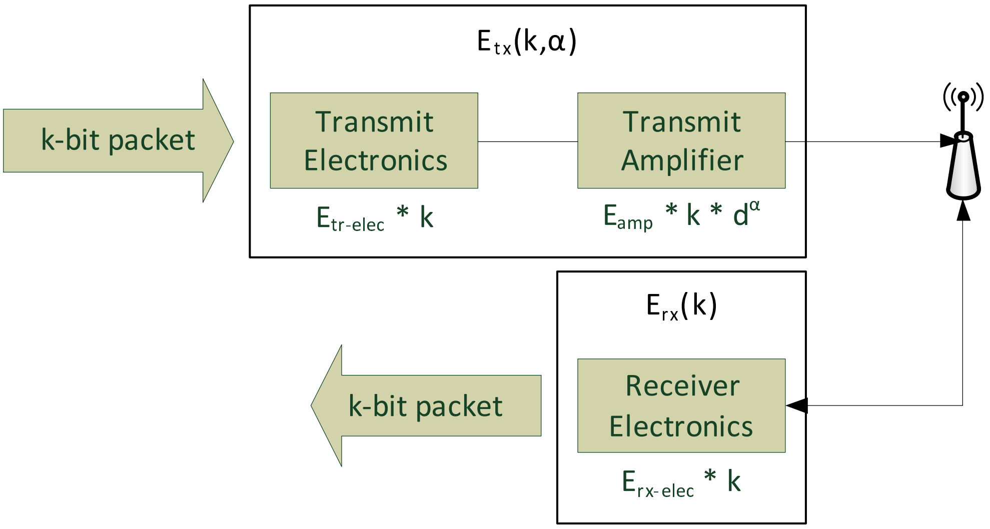


Figure 2.1: First order radio model

captured by the antenna and processed in receiver electronics circuitry to get the digital signal. According to the model, the energy consumed in receiving a k-bit packet is $E_{rc-elec} \times k$. The energy consumption of transmitting a k-bit packet to a distance d, and receiving a k-bit packet according to this radio model can be given as follows:

$$E_{Tx}(k,d) = E_{tr-elec} \times k + E_{amp} \times k \times d^{\alpha} \tag{2.1}$$

$$E_{Rx}(k) = E_{rc-elec} \times k \tag{2.2}$$

In the *idle listening* state, the wireless unit is neither in transmit nor in receive state. Instead it is waiting for possible packets coming from the node's neighbors. Since the unit is still on, a constant power E_{idle} can be assumed to be consumed. In *sleep* state, the whole communication unit is turned off, and no packets can be transmitted or received, and no energy is consumed. The energy consumptions of these four states are summarized in Table 2.1.

The values of the parameters in the energy consumption model described above can vary depending on the wireless communication technology used. Different studies in the literature assumes different values for these parameters. For instance, in [26], the parameters $E_{tr-elec}$ and $E_{rc-elec}$ are assumed to be equal and represented with E_{elec} and has a value of $50nJ/bit$. In the same work, E_{amp}

Table 2.1: Energy expenditure of wireless communication unit

State	Energy	Unit
Transmit	$E_{tr-elec} + E_{amp} \times d^{\alpha}$	Joules/bit
Receive	$E_{rc-elec}$	Joules/bit
Idle Listening	E_{idle}	Joules/sec
Sleep	0	

is taken as $100pJ/bit/m^2$, and propagation model is assumed to be free-space propagation where α is equal to 2. In another work [52], however, the authors take $E_{tr-elec} = 2 \times 10^8$, $E_{amp} = 1$, and $\alpha = 4$.

As a sensor node has different components, it can adjust its energy consumption according to its needs by deactivating the unused components. Therefore a sensor node can be in several energy consumption levels. As given in [57], if the current workload of a node can be determined, by dynamically switching the components off, the lifetime of a node can be prolonged.

It is worth mentioning that almost in every work in the literature the power consumption of components other than communication unit are neglected. In some studies the cost of idle listening is also ignored such as [26, 39, 64], whereas in some of them it is the main concern of the study [82].

Another point in the energy model is that the actual transmit cost of a sensor node is determined by the capabilities of the wireless equipment embedded in it. If the equipment does not support power control, which is adjusting the power in order to reach a distance d, the transmit operation turns to be a broadcast operation to a maximum transmission range R. In this case the energy cost of a send operation is constant. For instance, the energy cost ratios of idle-listening:receive:send operations are shown to be $1 : 2 : 2.5$ in the Digitan 2Mbps Wireless LAN module (IEEE 802.11/2Mbps) specification [82]. For a Mica2 radio (CC1000) the ratio is 1:1:1.5, whereas for a 802.15.4 radio (CC2420) the ratio is approximately 1:1:1 [83]. If the equipment supports dynamically adjusting of transmit power, however, the design of routing protocols for sensor networks gets more interesting and challenging. In this work we also consider the second case

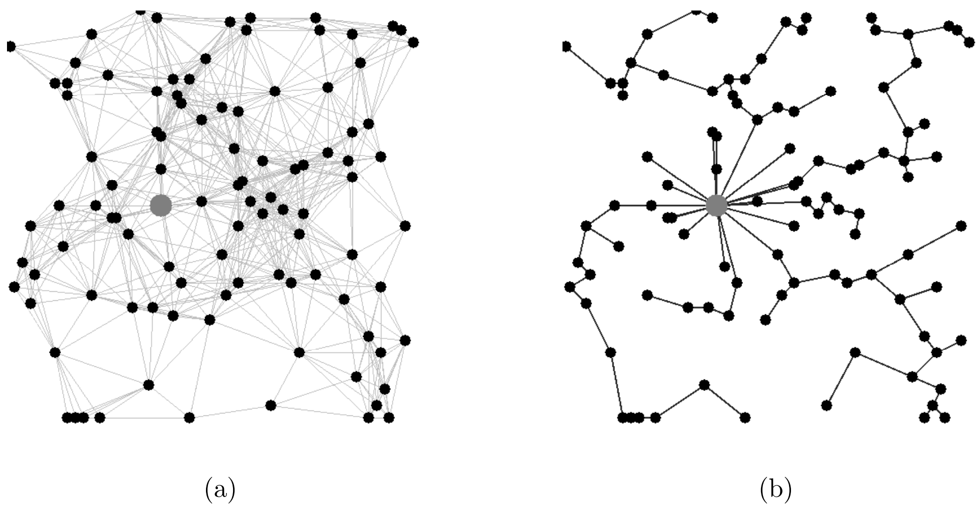


Figure 2.2: A sample network of size 100 nodes (a) and a routing tree for this sample network (b)

where the communication unit is able to control the transmit power.

2.3 Network Model

A sensor network can be modeled as a graph $G = (V, E)$ where vertex set V includes all sensor nodes and base stations, and edge set E includes all edges e_{ij} where node i can transmit a message to node j. If the transmission range of all nodes are equal and is denoted with R, the graph becomes a *unit graph* where $e_{ij} \in E$ if $d_{ij} \leq R$. Each node i has a location denoted by p_i and a sensing radius r_{s_i}. The area node i covers is denoted by D_i which is simply the disk with origin p_i and radius r_{s_i}. The target area to be covered is denoted by A. Figure 2.2(a) shows a sample network in a square-shaped target area.

If the radio channel is symmetric, then e_{ij} is in E if and only if e_{ji} is in E. But this may not be always the case due to reasons such as differing antenna or propagation patterns or sources of interference around the two nodes [32]. However, some MAC protocols such as MACA [34], MACAW [9], or IEEE 802.11 [18] allow unidirectional transmissions only when both source and destination nodes can communicate with each other, due to required RTS and CTS packet exchanges.

i
Etx(k, dij) + Erx(k)
j
i
Etx(k, dib)
b

(a) between two nodes

(b) between a node and base station

Figure 2.3: Energy consumption on a link.

This means although the transmission is in one way, a symmetric channel is required because of the control packets. Therefore, we can assume without loss of generality that all the links in the model are bi-directional.

We can associate a weight w_{ij} with each link $e_{ij} \in E$ representing the energy consumption of the transmission through that link. The weight includes both energy consumption of the transmitting node i and the receiver node j of the link, except when the receiver node is a base station. The weight w_{ij} can be defined as follows:

$$w_{ij} = \begin{cases} E_{tx}(k, d_{ij}) + E_{rx}(k) & \text{,if } j \text{ is sensor node} \\ E_{tx}(k, d_{ij}) & \text{,if } j \text{ is base-station} \end{cases} \tag{2.3}$$

where $E_{tx}(k, d_{ij})$ and $E_{rx}(k)$ are defined in Equations 2.1 and 2.2. As it can be seen, w_{ij} is smaller when destination node j is a base station. Therefore, in order to minimize the total energy consumption in the system the close enough neighbors of base stations should send their data directly to the base station without using multi-hop transmission.

In general, the routing structure in a sensor network can be modeled as rooted trees where the roots are the destination nodes. Since in most of the applications there is only one destination node (base station), we can simplify the model to only one tree T rooted at the base station (Figure 2.2(b)). The tree T does not necessarily span all the nodes in the network, instead it includes only the nodes that must sense and send data to the sink and the nodes that relay the data of the sensing nodes. That means there may be some nodes that should be included in the tree even though they are not sensing and generating data. They may be just responsible for relaying data. Such relay nodes are important since it is proved that multi-hop routing may save significant amount of energy in data transmission [8, 62] compared to single-hop routing, depending on some conditions.

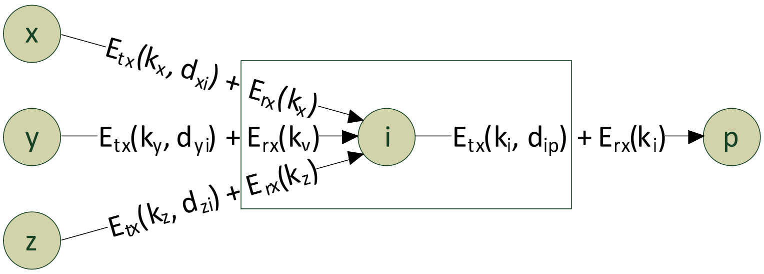


Figure 2.4: Load of a sensor node on a routing tree.

In a tree routing model, we can calculate the total energy consumption load (W_i^T) of a node i on the routing tree T in one round by summing the energy consumption at the node due to receiving data packets from the child nodes and due to sending the aggregated data packet to the parent node:

$$W_i^T = \sum_{\forall j,\, e_{ji} \in T} E_{rx}(k_j) + \ E_{tx}(k_i, {d_{ip_i^T}}^{\alpha}) \tag{2.4}$$

$$= \sum_{\forall j,\, e_{ji} \in T} (E_{rc-elec} \times k_j) + \left[\ E_{tc-elec} \times k_i + E_{amp} \times k_i \times {d_{ip_i^T}}^{\alpha}\right] \tag{2.5}$$

where k_i represents the number of bits that node i should send and p_i^T indicates the id of the node i's parent in the routing tree T. So, $d_{ip_i^T}$ is the distance between node i and its parent. Figure 2.4 illustrates the energy consumption of a node on a routing tree.

Let us introduce a new variable s_i which stands for the number of bits of the data sensed by node i. We can state that if node i is a relay node its s_i value is equal to 0. Now we can define a function $f_k(i)$ which gives the number of bits (k_i) that node i must send to its parent. In case there is no data aggregation or data fusion (see Section 3.1) the function can be defined as follows:

$$f_k(i) = \left(\sum_{\forall j,\, e_{ji} \in T} k_j \right) + s_i = k_i \tag{2.6}$$

If we assume that s_i values for all nodes are equal to s – which is generally the case – and there is a perfect data correlation in which receiving $n \times s$ bits result in only one packet of size s, $f_k(i)$ can be defined simply as:

$$f_k(i) = s = k_i \tag{2.7}$$

In this special case the load of a node i on a routing tree T (W_i^T) can be simplified as follows:

$$W_i^T = s \times \Big[E_{rc-elec} \times \delta_T^-(i) + E_{tc-elec} + E_{amp} \times {d_{ip_i^T}}^{\alpha} \Big] \tag{2.8}$$

In Equation 2.8, $\delta_T^-(i)$ is the *in-degree* of node i on routing tree T. If we further take $E_{tr-elec} = E_{rc-elec} = E_{elec}$ as in [26] we can further simplify the load as in Equation 2.9.

$$W_i^T = s \times \Big[E_{elec} \times \delta_T(i) + E_{amp} \times {d_{ip_i^T}}^{\alpha} \Big] \tag{2.9}$$

where $\delta_T(i)$ is the degree of node i in routing tree T. As seen in the equation for this special case there exists only two parameters that affect the power consumption of a node: degree and distance to the parent. Nodes with high degrees could quickly drain their energies. Since distance has a power of α, the increase in energy load is exponential when the distance is increased. Therefore, to obtain a routing tree that is maximizing the lifetime, we have to try to minimize the degree for a node while minimizing the distance the node will transmit. Additionally, we have to balance the energy load among the nodes (for example, by recomputing the tree from time to time).

The routing tree model can be extended to any kind of application. If there should be more than one routing tree in a round – which is possible if different requests are sent to different sensors – all the above computation can be repeated for all the trees, and by super-positioning them all, we can find the weights of nodes. In this thesis we choose to have only one routing tree in each round of data gathering for the sake of simplicity.

One important point about the routing tree model is that the tree T does not need to be the same in each round. So, the routing tree can be recomputed over time. As we will see in next sections this recomputation can improve the lifetime of the system [26, 64], because it enables balancing of the energy load. In [30] a good analysis is given about when to recompute the routing tree.

2.4 Lifetime Definitions

In the context of sensor networks, the network lifetime can be defined in various ways. The concept of lifetime in sensor networks is highly application dependent. In an intuitive way the lifetime can be defined as the time period from the deployment and initialization of the system until it can not do whatever it is supposed to do. However, it is not so easy to formulate the time when the system can not show its expected behavior. In order to simplify the definition of lifetime we can categorize the needs of the applications into three: *number of alive nodes*, *network partitions* and *coverage*.

In applications where the number of alive nodes directly affects the performance of the system the lifetime is characterized with that number. If for an application it is important to have all the nodes operating together – since the quality of system will be dramatically decreased after first node failure– lifetime can be the time elapsed until the first node failure. However, in applications where receiving information from the area of interest is very important even if there is only one sensor node on the field – e.g. battlefield surveillance – the time in rounds where the last node depletes all of its energy defines the lifetime. In general, we can state that for applications for which the performance is related with the number of alive nodes, the lifetime is the time elapsed until some specified portion of the nodes die.

It is worth mentioning that the first node failure metric is very appropriate to measure the load balancing performance of a routing algorithm. If an algorithm can balance the energy consumption well among the nodes, the time until the first node drains out its energy will be maximized.

Another alternative definition can be the time elapsed until the network is partitioned at which time some of the alive nodes will not be able to transmit their data to the base station. With this metric we can measure how bottleneck nodes are handled by an algorithm. If a network becomes partitioned quickly, that means the energy load of bottleneck nodes are not managed very well.

In applications where sensing coverage is very important, the functionality of the network is not determined directly by how many are alive, but determined by the coverage achieved by the alive nodes. For instance, in event-driven applications like fire detection sensor network systems, what important is to cover the whole area in order to detect a fire instance that can happen at any point in the area. For such systems, the lifetime definition can be given as the time until there is not enough alive nodes to cover a specific portion of the region. A specific instance of such systems is the ones that require the coverage of the whole region.

It is desirable that a routing scheme considers several lifetime definitions and provides reasonably good results for them. In this thesis, we consider all these lifetime definitions in our performance evaluations.

2.5 Problem Statement

This thesis focuses on routing and node activity scheduling (i.e. sleeping node scheduling) problems in wireless sensor networks. The routing and node scheduling solutions to be developed, however, depend on the wireless sensor network application. There are various sensor network application scenarios, and depending on the scenario, the requirements for a routing and scheduling solution are different.

The following are our assumptions about the features of sensor networks and application scenarios we consider in this thesis.

- The sensor nodes are homogeneous and energy constrained.
- Sensor nodes and sink are stationary and located randomly.
- Every node knows the geographic location of itself by means of a GPS device or using some other localization techniques [7, 25, 27, 28].
- Every node senses periodically its nearby environment and has data to send to the sink in each round.

- The nodes have a maximum transmission range denoted by R. Sensor nodes are thus normally not in direct communication range of each other. Therefore applying centralized approaches will have a high communication cost for gathering network information at a node.

- Data fusion or aggregation is used to reduce the data volume. We assume a perfect aggregation or correlation of data which means combining n packets, each packet being of size k, results in only one packet of size k.

- We also assume that the sensing period (the duration of a round) is much larger than the time required for transmitting all the information from all nodes to the sink.

- The nodes are capable of controlling their power. This means the nodes can adjust their power levels to transmit to different distances.

- The nodes can be put into sleep mode if it does not harm network functionality.

In the application scenario we consider for this thesis, sensor nodes periodically sense the environment and generate data in each round of communication. Given a routing plan, each sensor node receives the data from its children, aggregates or fuses them into one single packet, and sends the packet to the next node on its way to the sink. Instances of such an application can be event (fire, intrusion) detection systems or average data (temperature, humidity) extraction systems.

Note all nodes need to be active. Some nodes can be put into sleep provided that the remaining active nodes can cover the region. How many nodes and which nodes will be active affect the coverage and energy consumption performance of the network. One problem we focus in this thesis is determining the optimum number of nodes (which may not be the minimum number of nodes) that need to be active without harming network functionality. Then, over the active nodes a routing plan has be used to carry the data to the sink node.

The problem is to find an energy efficient routing plan which maximizes the network lifetime. The routing plan determines for each round the roles of each node and incoming and outgoing neighbors for data forwarding and aggregation for each alive node. In other words, firstly the nodes which should be alive must be found on each round, and finally a tree spanning the alive nodes must be found for each round as the routing plan. The routing scheme should also include mechanisms to handle node failures and support new node arrivals.

Chapter 3

Related Work

In this chapter, we will discuss the related work done on wireless sensor network routing and node scheduling considering energy efficiency as the most important goal. There are many routing protocols and node scheduling algorithms proposed in the literature that try to use the energy efficiently and improve the sensor network lifetime as much as possible. We will also briefly discuss some other approaches, reducing data traffic volume, mobility and efficient deployment and topology construction which can be used to improve network lifetime. We will start our discussion with those other approaches to reduce the unnecessary energy consumption and prolong network lifetime.

3.1 Minimization of Transmitted Data Volume

One of the most effective techniques to reduce the power consumption in a sensor network is to minimize the transmitted data volume, since the most power consuming component of a sensor node is its wireless communication unit: the less we use that component, the more we save energy. There are different methods to achieve this goal in the literature.

The most common and easily applicable method is *data aggregation*. The idea behind this approach is that since usually the collected data from sensors is too much for an end-user to process, the collected data can be aggregated – eg. with functions like max, min, count, avg – and presented to end-user as a single value. Instead of doing the aggregation after all the data is collected to the base station, if we can do it in the network while the data is gathered we can save a large amount of energy. One disadvantage of this method is that it cannot be used for applications where each individual sensed data need to be collected at the base station.

Another way of reducing the packet size is the *data fusion* technique. By using the data fusion technique the unreliable data measurements can be combined to produce a more accurate and high quality signal by reducing the noise and enhancing the common signal [26]. For instance, the sound signals can be combined by using beamforming algorithms into one single packet that contains all the relevant information from the individual signals. One important disadvantage of this method is being highly application dependent which means that its applicability is related to the type of sensed signal.

In [59] different in-network aggregation algorithms are presented. The paper also gives a comparison of the algorithms with respect to trade-offs between energy efficiency, data accuracy and freshness. We encourage the interested users to read that work.

Another interesting way of minimizing the transmitted data volume is *prediction* based methods [19]. If the application is tolerant to small errors, a precision clause can be added to the query which indicates the permitted error. The main idea behind this technique is to predict the value of the data sensed in children. If it can be correctly predicted within the given precision there is no need to transfer the newly sensed data to the parent. Since the child and parent nodes uses the same prediction function, the child can know what its parent predicts and send the data only when the prediction does not guarantee the precision value given in the query. In this way the energy saving is maximum since the communication only occurs when the source will send an unexpected value.

3.2 Mobility

Another effective way of improving system lifetime is to utilize the mobility. The main idea behind mobility is to reduce the distance between source and the destination dynamically since the most power consuming operation is transmitting to distances. There are two kinds of mobility scenarios in the literature: *mobile base stations* and *mobile relays* where in the former only the base station is mobile, whereas in the latter case there are some mobile gateways that collect information from the fixed sensor nodes and transfer the data to the base station.

One advantage of incorporating mobile elements in the network is that it reduces the redundancy in the number of deployed nodes, since the reason for deploying a dense network is to ensure the connectivity of the network. However in mobile case, sparse or even unconnected networks can also be handled. Another advantage is that it saves the redundant multi-hop routing by having the mobile nodes visited the fixed sensor nodes to collect data. Although this increases the latency as well, it can be used in delay-tolerant applications [60].

One of the earliest application with mobile elements is incorporating the randomly moving mobile 'Data Mules' (Mobile Ubiquitous LAN Extensions) in data gathering [55]. After this work, instead of having random movement, a controllable or predictable movement is considered [24, 60, 71]. These works and many others in the literature propose different algorithms for Mobile-Element-Scheduling (MES) problem which is defined as determining the order and the frequency of node visits of the mobile element in which none of the buffers of the fixed sensors overflows. It is shown that the mobility can improve the lifetime up to four times compared to the static networks [71].

3.3 Efficient Deployment and Topology Construction

In some of the applications, such as biomedical sensor applications, the location of sensor nodes are pre-determined and fixed. We can take the advantage of determining and knowing the locations of the nodes and base station(s) for power-efficient topology construction and routing. In some other systems, sensor nodes cannot be placed manually and therefore their locations may not be decided a priori and where nodes are located may not be known exactly. , A base station, however, is usually placed manually and therefore its location can be pre-determined. The location where base station is placed may also have an impact on the energy performance of the network.

In applications where the locations of all nodes can be predetermined, there are a couple of questions that must be answered in order to get a low energy/cost system: How many sensors should be deployed and how they are deployed [36]. In many works [8, 36, 62] optimal deployment of sensor nodes in 1D is obtained independently. According to all of these works the optimal placement of nodes in 1D can be achieved when the nodes are equally separated from each other. The required number of sensors is also obtained in these works.

Although the 2D or 3D case is not so easy, in different works the effect of different topologies are investigated in terms of power consumption. In [53] the following topologies are examined with the proposed routing protocol DSAP: 2D Mesh with maximum of 3,4,6, and 8 neighbors and 3D Mesh with maximum of 6 neighbors. On the other hand, in [36] the authors proposed that the energy consumption in a two dimensional network is minimized when nodes are evenly spaced inspired from the analysis in 1D. Consequently they investigate even distributions of nodes in triangular, square and hexagonal shapes. They concluded that the triangular arrangement is optimal in many situations.

On the other hand, in systems where the number and the locations of base stations can be determined a priori, it is also important to use this flexibility in order to achieve good lifetimes. [10] showed that the number and locations of the base stations has a great impact on network lifetime. The main goal in that work is to maximize data rate. Therefore, firstly a method for finding the maximum-rate routing is proposed based on maximum flow problem when the number and the locations of the base stations are given. It is also shown that optimizing the number and locations of base stations is NP-complete even in very well structured network topologies. So, they run different search algorithms for finding the optimal layout of the base stations. In another work [43], algorithmic approaches are proposed to locate the base stations optimally which achieve a maximum network lifetime. The main assumption of the work is a two-tier network architecture where there are intermediate application nodes that receive the data from the sensor nodes and send it to the base station after necessary processing.

Another important issue in minimizing the total energy consumption is to find the transmission power for each node in order to maintain a strongly connected network. The issue is called *topology control* in the literature. The topology control affects the system performance in several ways. First of all, it affects network spatial reuse and thus the traffic carrying capacity. Choosing a large power level results in excessive interference, whereas choosing too small power level results in a disconnected network. Collisions can also be avoided by choosing the minimum possible transmission power. And finally and may be the most important effect is on power consumption. There are many works in the literature that tries to find the minimum transmission power for each node where some of them are LMST [38], enclosure-based approach [52], CBTC(α) [37], COMPOW [41] and CONNECT [49]. The idea behind this class of protocols is to compute a topology over the visibility graph and then determining the maximum transmission power for each node as the power required to transmit a signal to the farthest neighbor in the resulting topology.

In [52] a position based distributed algorithm is proposed in order to achieve minimum power consumption. They first define the *relay region* for a transmit-relay node pair as the region where transmitting through the relay node is advantageous in terms of power consumption instead of direct transmission. After that

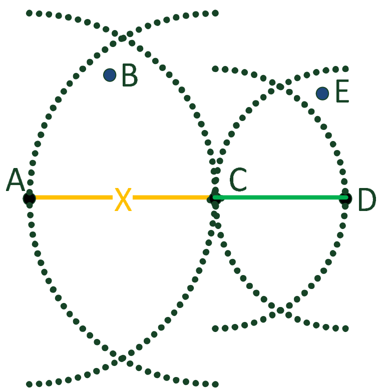


Figure 3.1: Computation of RNG.

they define *enclosure* of node i as the union of the complement of relay regions of all the nodes that node i can reach by using its maximal transmission power. The union of enclosures of all nodes forms the final topology called *enclosure graph*. In other words, an edge e_{ij} is in the enclosure graph if and only if the direct transmission between node i and node j consumes less energy than the total energy of all links of any path between them. It is proved that the enclosure graph includes the minimum cost tree if there is no data aggregation.

However, since we consider only scenarios with perfect data aggregation, the topologies that we focus in this work are supersets of Euclidean MST.

One of them is the relative neighborhood graph (RNG) [69] which is defined as follows. An edge e_{ij} is included in the Euclidean RNG graph if there are no nodes closer to both nodes i and j than the distance between nodes i and j. That is, an edge e_{ij} remains in RNG if it does not have the largest cost in any triangle $\overset{\triangle}{ikj}$, for all common neighbors k. The MST of a graph is a subgraph of its RNG.

Figure 3.1 shows computation of RNG edges for a sample partial network. In this network, the edge between node A and node C is not included in RNG since there exists node B that is closer to both A and C. On the other hand the edge between node C and D is included in RNG since there are no nodes closer to both nodes C and D. Note that node E does not prevent the inclusion of edge CD to the RNG since it is only closer to node D.

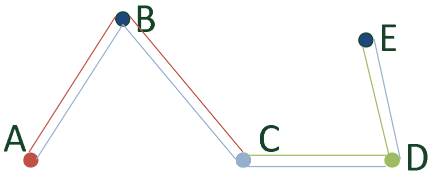


Figure 3.2: Computation of LMST.

As an alternative in [38] a powerful topology control algorithm which is called local minimum spanning tree (LMST) is proposed. The idea of the algorithm is actually very simple. By collecting information about its neighbors each node computes an MST spanning all its neighbors. After computing the MST of the neighbors, each node i selects the edges (e_{ij}) where node j is a direct neighbor of node i in its own MST. So, the direct neighbors of a node in its local MST are called its LMST neighbors. If the LMST neighbors of all nodes are combined together, the final topology called LMST can be generated. The resulting structure is, however, a directed graph. The structure can be converted to an undirected one in two ways [38]. First way is to include edge (e_{ij}) only when both nodes i and j include that edge (LMST$^-$). The second way is to include that edge when either node i or node j include it (LMST$^+$). In this study we choose to use LMST$^-$ in our simulations, but our algorithms can support both.

Figure 3.2 illustrates computation of LMST edges for the same partial network above. In this case, each node separately computes its MST considering the nodes in its communication range. In the figure, the edges of local MSTs for nodes A, C and D are shown with a color corresponding to the nodes. Since the edge between node A and node C is not in both nodes' LMST neighbor set, it is not included in the global LMST. On the other hand, the edge between node C and D is included in local MSTs of both nodes. Therefore, the edge CD is included in the global LMST.

There are some desirable properties of the LMST structure which make using the structure in the context of sensor networks advantageous. First of all, MST of a graph is a subgraph of its LMST and the LMST is a subgraph of its RNG [42].

Therefore it guarantees to preserve connectivity. Moreover, if link costs are defined based on Euclidean distances, the maximum degree of a node is bounded by 6 as it is in Euclidean MST. This is a desirable property since the load of a node is directly related to the degree of the node, as it is shown in Section 2.3.

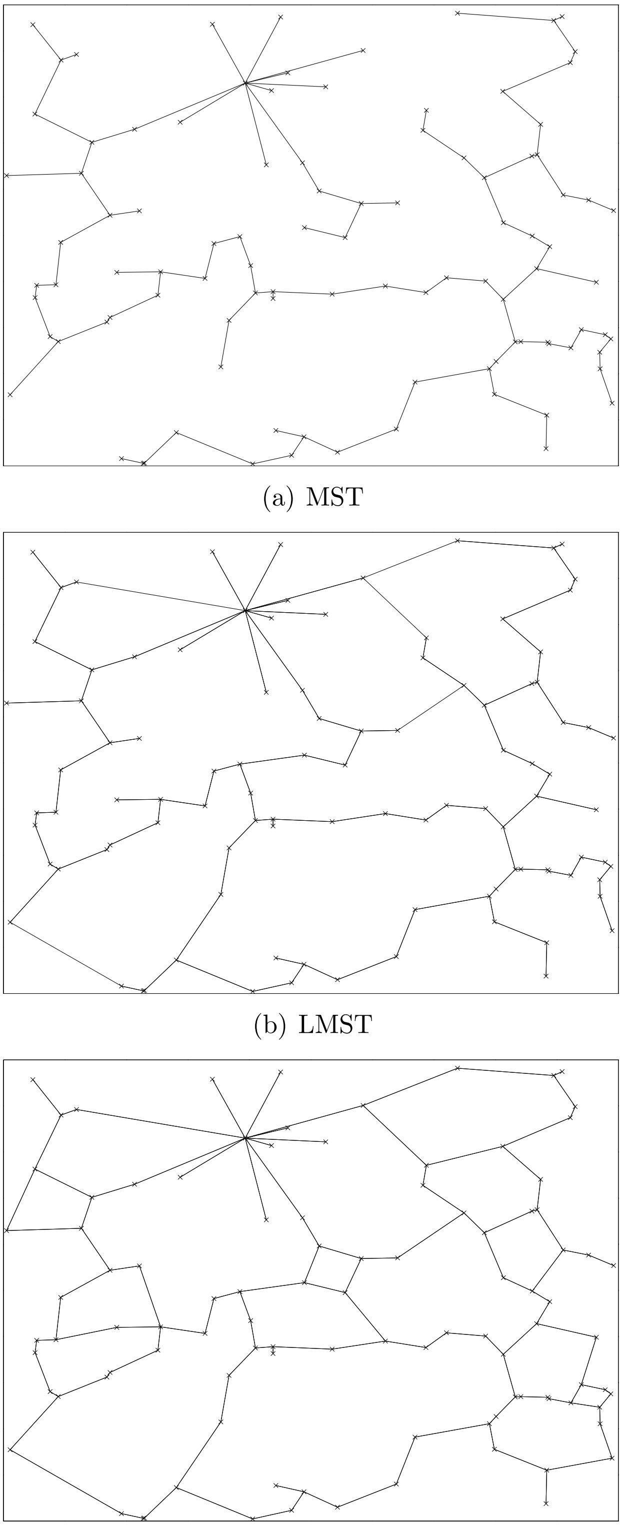
In [38] the authors compare their LMST structure with the enclosure graph and find out that the enclosure graph performs better in terms of energy consumption. However, the comparison did not consider the effect of data aggregation.

It is also worth mentioning that although the RNG and LMST structures are defined based on Euclidean distances, they can be used with other link cost functions as long as the functions are symmetric [20, 46]. We can use for instance, the cost function given in Equation 2.3, while computing the structures. Figure 3.3(b,c) shows this case. For the rest of the study if we mention MST, LMST and RNG, we mean the structures that are computed using the link costs given in Equation 2.3. They resemble the original MST, LMST and RNG structures, except replacing some links by direct links to sink (the effect of adding second part of Equation 2.3). However, the structure may become considerably different in the whole network, if a cost function that depends on nodes' remaining energies is used to define them.

An important advantage of using structures like RNG and LMST is that they can be constructed very efficiently in a localized manner. Node deletions and additions do not globally change the structure. Only local changes in the structure are required and they can be efficiently computed when a node fails or when a new node is introduced to the network.

3.4 Routing

There are many works in the literature that investigate the effect of routing on the network lifetime. It is shown that even in very simple scenarios the routing algorithm individually affects the performance considerably [26, 64, 86].



(a) MST

(b) LMST

(c) RNG

Figure 3.3: Comparison of different topologies.

We will not go over all of the routing protocols in the literature since there exists some surveys about different aspects of routing in sensor networks [2,4]. In this section we will briefly mention about the basics of routing and some of the routing protocols which are related to our work.

There are two classes of routing approaches in the literature: *reactive* and *proactive*. In reactive routing algorithms the routes are set up only when a request is made [31, 32], whereas in proactive routing the routes are determined as soon as possible after the deployment [26, 64]. Proactive routing also makes route management mandatory, whereas in reactive protocols it is not necessary since the routes are found again at each request.

The aim of the routing algorithms can be divided into two also. In one class, the total power consumption in a round is minimized, while in the other the lifetime of the system is maximized. These two goals seem to be the same at first glance, however minimizing the total power consumption in a round does not guarantee the maximum lifetime. Consider a case where there is only one source and one destination in the system. If the minimum cost path is used the total power consumption is minimized. However if the same route is used continuously, the power of the nodes on that path is depleted. Therefore it is a good idea to sacrifice a bit from the minimum cost routes in order to get a good lifetime. [64] experimentally shows this situation.

The characteristics of a routing algorithm is directly related with the environment it will be used. The energy model, the lifetime definitions, and use of data aggregation are some of the parameters that affect the design for a good routing protocol. So each application requires its own specialized routing solution in order to optimize the requirements of that specific application.

In our study we will work on environments where all the nodes are responsible for sending their readings periodically to the base station. We will briefly go over the protocols that are specially designed for this kind of applications.

There exist several routing protocols for data gathering without aggregation. The majority of them uses the shortest weighted path approach using several combinations of transmission power, reluctance, hop count, and energy consumption metrics [14, 15, 56, 62]. The classical routing algorithms such as AODV [47] or Directed Diffusion [31] can be considered also for this case.

There are also algorithms in the literature that take the data growth factor into consideration, where data may not be perfectly aggregated. The purpose of these papers is to provide an optimal routing solution which is adaptive to the data growth factor. Hua and Yum [29] described an algorithm for joint optimization of routing and data aggregation. Row data are sent to downstream neighbors. The receiving neighbor encodes the data using local information, with certain compression rate. Transit data (already compressed by upstream neighbors) are directly forwarded to the next hop neighbors. Therefore data aggregation is done only by neighbors of measuring sensors, and the size of aggregated data varies. This problem statement and the model are different from the ones used in this study. Upadhyayula and Gupta [70] proposed a combination of single source shortest path spanning tree and minimal spanning tree algorithms to construct optimal data aggregation tree which controls latency by limiting the number of children of each node while optimizing energy consumption. Constant data growth factor spans aggregation level from no aggregation to full aggregation at each intermediate node. Although the problem statement is more general than the one in this article, their algorithm is centralized. One important point is that the authors consider MST as optimal solution in perfect correlation case. Park and Sivakumar [44] optimized number of messages sent while aggregating data originating from k of the n sensors, with various data growth factors. Their solution aggregates correlated data from neighboring sources at nodes of minimum dominating set (MDS). It then creates shortest path of MDS nodes tree by basic flooding. In this study we consider perfect correlation with $k = n$. For this case, [44] reduces to a constant number of messages (one per each sensor), and does not consider energy optimization.

There are also a number of protocols for data gathering with aggregation. Most of them are centralized approaches and assume that all the sensor nodes are in direct communication range of each other and the sink.

In [33] a linear programming solution to maximize the lifetime is proposed. The solution provides near optimal results. However, their approach has high computational cost and must be applied in a central location.

One of the first papers on this topic proposes a low energy adaptive clustering hierarchy (LEACH) [26] protocol which is a distributed two-level hierarchy construction algorithm. It is assumed that base station is far away from the field of interest, so directly communicating with it is a very costly operation. In LEACH, the key idea is to reduce the number of nodes communicating directly with the base station. The protocol achieves this by forming a small number of clusters in a self-organizing manner, where each cluster-head collects the data from nodes in its cluster, fuses and sends the result to the base station. In this protocol sensors randomly decide whether or not to become clusterheads. If not, they join the nearest clusterhead and transmit sensed data to it. Clusterheads aggregate collected data and transmit directly to the sink. In order to balance the load among the nodes LEACH uses randomization in cluster-head selection and achieves a significant amount of improvement compared to the direct transmission approach where each node directly transmits its data to the base station. Since LEACH protocol relies on randomization, it is far from being optimal.

In [39] a power efficient data gathering scheme which is called PEGASIS is proposed. PEGASIS is an improvement over LEACH for the same scenario. PEGASIS reduces the number of nodes communicating directly with the base station to one by forming a chain passing through all nodes where each node receives from and transmits to the closest possible neighbor. The data is collected starting from each endpoint of the chain until the randomized head-node is reached. The data is fused each time it moves from node to node. The designated head-node is responsible for transmitting the final data to the base station. There are several disadvantages of PEGASIS protocol. First of all it is a centralized algorithm. Moreover, finding the minimum length chain is actually the same as the traveling salesman problem and therefore it is NP-complete. Also the delay is another problem for PEGASIS.

One of our previous works [64] presents a different centralized data gathering algorithm for the same scenario which is named as PEDAP. The basic idea in PEDAP is to improve the lifetime by forming a minimum transmission cost tree spanning all the nodes. This tree can be efficiently computed in centralized manner using Prim's minimum spanning tree algorithm [48]. With this simple idea it improves the system lifetime dramatically compared to its alternatives. In that work the authors set the cost of a link as the energy consumption of sending data on the link. The main idea behind the success of the scheme is to minimize the total energy consumption in a round, while balancing the load equally among the sensor nodes. Being more specific, PEDAP protocol uses the link costs given in Equation 2.3 and computes the minimum energy cost tree by using Prim's MST algorithm. PEDAP protocol differs from the Euclidean MST with only the degree of the sink. Fortunately, for the nodes the properties of the Euclidean MST are conserved. For example, the degree of the nodes (except the sink) is at most 6. Also, as stated in [45], the longest edge in the Euclidean MST is the minimum common transmission range for network to be connected. So the transmission distances are also optimal for the nodes routing using PEDAP. As shown in Section 2.3, the energy load of a node is directly related with its degree and the distance to its parent, and PEDAP balances these parameters well. Also PEDAP consumes the minimum amount of energy in a single round. Moreover since the transmission cost w_{ij} given in equation 2.3 considers the fact that the base station is not battery limited, the algorithm is also capable of choosing the optimal number of nodes that must be communicate with the base station. However, [64] also shows that this optimal routing does not provide optimum lifetime. In order to achieve a better lifetime a power aware version of the algorithm PEDAP-PA is proposed. This protocol provides near optimal lifetime for the first node failure by sacrificing the lifetime for the last alive node. In PEDAP-PA, the cost of the links are changed so that the remaining energy of the sender is also taken into consideration. Specifically, the cost function is chosen to be:

$$w^*_{ij} = w_{ij}/r_i, \tag{3.1}$$

where r_i is the normalized remaining energy of node i. Note that this cost metric is not symmetric. It is used by a node j when looking for candidate neighbor i on route toward sink. The PEDAP-PA algorithm simply finds the minimum spanning tree with these link costs. Since the link costs vary over time, the authors proposed recomputing the routing tree from time to time. By changing the routing tree over time the load on the nodes is balanced and a longer lifetime compared to the static version is achieved. In this way the lifetime in terms of the first node failure is almost doubled.

In HMRP [74] a multi-path routing approach is used for the same problem. By broadcasting a special packet from the base station each node determines its potential parents on its minimum-hop path. Instead of selecting one alternative parent and fix it, the protocol uses the potential parents in a round-robin fashion.

Another work which is worth to discuss is energy efficient spanning tree protocol (EESR) given in [30]. EESR is similar to PEDAP-PA but has some advantages over PEDAP-PA algorithm. For instance, edge weight assignment used in EESR considers both transmitters and receivers remaining energy levels. The key observation is that if the residual energy of receiving node is not considered in weight assignments, the receiving node can have a higher load. With the edge weights they use, the algorithm prevents transmitters and receivers from being overloaded. Another advantage of it is dynamic determination of the duration of recomputation period. it chooses the number of routing trees intelligently and tries to recompute the tree accordingly. The algorithm is however centralized.

Both PEDAP and EESR showed that MST based structure is suitable for environments where all the nodes have data to send and data can be aggregated (fused) in the relay nodes. The drawback of PEDAP and EESR protocols is the centralized nature of MST and the lack of quick response to node failures.

In [78] authors studied the construction of a data gathering tree to maximize the network lifetime, which is defined as the time until the first node depletes its energy. Nodes do not adjust their transmission radius to the distance to neighbors (different from our model). Even in this form, the problem is shown to be NP-complete. They design a centralized algorithm which aims at finding a spanning tree whose maximal degree is the minimum among all spanning trees, since energy consumption at each node only depends on the number of messages received from children nodes, that is, on the number of children. Such tree then reduces the load on bottleneck nodes.

3.5 Sleep scheduling

Another class of algorithms that reduces the energy expenditure is to put the nodes into different levels of sleep states. As pointed out in Section 2.2 a sensor node has different components and switching off the unnecessary components results in large energy savings.

The roles can be represented by the components of a sensor node that must be turned on in order to accomplish the job. In this work we consider three roles: sensing, relay, and sleep. In sensing state all the components of a sensor node must be turned on since in this role the node is responsible for sensing its nearby environment and reporting its data to the sink. In relay state the sensing component can be turned off since this role only requires the relay of the data of the sensing nodes to the sink. The sleep state means just turning off all the components since the node is not required for data transmission. Since usually the sensor nodes are densely deployed, many nodes are not needed in a round and thus can be put in sleep state and this results in an enormous amount of energy saving.

In [57] *dynamic power management* method is proposed. The key observation is that switching of node states takes some finite time and resource. Therefore if the energy saving achieved in the sleep mode cannot compensate the energy consumed to get to that state because of early wake-ups, there is no point to switch to that sleep state. However, we generally cannot predict when a component should be waken up and hence we need stochastic analysis in order to predict when a component is needed. In [57] the authors assume a multilevel sleep state model where at deeper levels the power consumption is less, while getting to that state takes more time and energy. They also proposed a workload prediction strategy based on the adaptive filtering of the past workloads. A node decides being in which state based on this prediction. If the probability of occurring an event is low, the node switches to deeper sleep states.

In another study called sensor-MAC (SMAC) [82] a new medium-access control (MAC) protocol that is designed for the sensor networks. Generally reservation and scheduling based MAC protocols such as TDMA-based protocols have a natural advantage of knowing when a node is needed. However dynamic management of the time slots to the wireless nodes is not an easy task. On the other hand contention based protocols such as IEEE 802.11 [18] consumes much energy mainly due to idle listening. Therefore, SMAC tries to minimize the power consumption by using a combination of scheduling and contention based routing. In that work four sources of energy waste is given: *collision*, *overhearing*, *control packet overhead*, and *idle listening*. SMAC tries to avoid all these problems by incorporating a periodic listen-sleep schedule. In a unit of time each sensor node sleeps half of the time and listens in the other half. Intuitively this reduces the power consumption close to 50%. They also proposed methods for avoiding other sources of energy wastes and as a result they provide good energy gains compared to its alternatives.

GAF, SPAN, and STEM are different node scheduling algorithms which are worth to mention. In GAF [80], the key observation is that the nearby nodes can perfectly replace each other in a routing topology. So by finding nodes that can be replaced by others and putting those into sleep state energy is saved. In order to find such nodes GAF uses a grid virtually defined on the field and it keeps only one node working in a grid cell. The role of being active in a cell is rotated in order to balance the workload. As a disadvantage it requires the nodes to be deployed very densely. On the other hand, SPAN [16] constructs a routing backbone where the nodes in the backbone are responsible for forwarding the data packets. The other nodes only sense and generate data. Whenever other nodes must send their data to the base station, the data must be firstly sent to a backbone node. So, the nodes that are not in backbone can switch to sleep state more frequently. Again the workload is balanced by rotating the role of being in backbone. STEM [54] takes the problem to extreme with the observation that the sensor nodes are generally in monitor state instead of being in transfer state especially for event driven applications. Therefore, turning on the radio when it is not needed is a great waste of energy. The authors assume that initially all nodes are in sleep state. Whenever a data is detected and must be sent to the base station, by using a separate channel, which uses very low power, a wake-up signal is sent to the node on the way to base station. A node which receives wake-up signal turns on its primary radio in order to receive the data. The procedure repeats itself until the data reaches to the base station. The disadvantage of this method is the high latency in relaying data to the base station.

Another approach to sleep scheduling is proposed in [81] which is a probing based node scheduling protocol (PEAS). In PEAS, a node can be in three states: *sleep*, *probe*, and *working*. The protocol keeps only a subset of the nodes in working state and allows the others to sleep. The working nodes continue working until they deplete all of their energy. The sleeping nodes occasionally wakeup and probe their environment in other to find a working node. If there is a working node in the close neighborhood of the probing node, the node again falls asleep. Otherwise it begins working. Although PEAS provides good energy saving it does not guarantee coverage.

The protocols so far have no guarantee to cover the whole area. They only try to get a connected network to relay the packets to sink. The distributed node scheduling protocol given in [67] guarantees that the original sensing area is covered after turning off redundant nodes. In this protocol, the nodes advertise their location to their neighbors in their sensing ranges at the beginning of each round. After receiving the location information each node decides whether it is turned off or not by using the coverage information. In other words, if the sensing range of a node is fully covered by its neighbors, the node decides to switch off itself. However, if all the nodes make decisions at the same time there will be some blind points in the area which is not covered by any of the nodes. In other to prevent this, the protocol applies a back-off based self-scheduling step. In this step the nodes wait for a random back-off time and then broadcast a status message informing their neighbors about the decision they have made if the decision is turning off. So the nodes that receive this status message remove the sending node from their neighbor lists and remake their decision. However, this protocol cannot provide the optimal number of alive nodes since it does not consider the nodes whose distance is greater than the sensing range. Another disadvantage of this protocol is that it does not guarantee the connectivity.

There are some protocols in the literature which guarantee both connectivity and coverage [11, 72, 84]. In [84], it is proven that the coverage implies connectivity if transmission range is greater than two times the sensing range. The authors propose Optimal Geographic Density Control (OGDC) algorithm which tries to minimize the number of alive nodes while maintaining coverage. The idea behind OGDC protocol is to keep a node alive only if it covers an intersection point of two sensors and minimizes the overlap with other alive nodes. In [72], the authors propose a protocol called CCP, which provides k-coverage and k-connectivity. CCP uses SPAN [16] protocol to provide connectivity if the transmission range is not greater than two times the sensing range. Another protocol proposed in [11], also guarantees both connectivity and coverage. Different from the previous ones, [11] considers the neighbors that are more than one hop away. In this way it can provide full coverage with a smaller number of nodes. It also generalizes the provided solution for varying transmission and sensing ranges.

With the same idea of preserving the coverage in mind [58] introduced a heuristic that selects mutually exclusive sets of sensor nodes, where the members of each of those sets together completely cover the monitored area. After finding the sets, only the nodes in a set are kept working in a round while the rest are switched to sleep state. The sets are used in a round robin fashion and since each set individually covers the whole area, the lifetime of the system can be improved while preserving the coverage. In [12] a different heuristic called MC-MIP is proposed which provides better results. [73] further utilizes the nodes that are not included in any of the cover sets (subject sensors) by assigning them to the appropriate cover set considering the routing protocol.

As it can be seen it is not so easy to achieve an optimal node scheduling. The key questions that must be answered carefully in deciding the node schedule are: which nodes must be in sensing, relay, and sleep state, and when these roles must be exchanged among the sensors.

Chapter 4

Lifetime Analysis

In order to propose a good routing scheme and evaluate it, it is important to have a theoretical model that can provide the optimum achievable targets. As mentioned in the previous chapters, it is very important to optimize lifetime in wireless sensor networks and therefore to know the optimum achievable lifetime. Towards this goal, in this chapter, we first formulate the lifetime of a network with respect to the first node failure time. After that we propose a new upper bound for first node failure time (i.e. the maximum achievable lifetime). We then investigate this upper bound further and try to relate it with number of nodes in the routing tree. This analysis is for both routing and node scheduling problems to compare the proposed solutions against optimally achievable values.

4.1 Lifetime Formulation

We will provide a lifetime formulation for sensor networks that use a rooted tree as the routing structure. We assume that a routing tree T spanning active nodes is established using which nodes forward their data towards the sink node. We assume the tree is fixed throughout the lifetime. Then, the lifetime of a node i in T, denoted with L_i^T, can be defined as follows: depending on the residual energy of the node i (R_i):

$$L_i^T = \frac{R_i}{W_i^T} \tag{4.1}$$

Here, R_i is the residual energy of the node, and W_i^T is the energy load of the node, which is defined in Section 2.3:

By using this lifetime definition for a node, the lifetime of an edge $e_{ij} \in E(T)$ can be formulated as:

$$L^T(e_{ij}) = \min \{\ L^T(i), L^T(j)\ \} \tag{4.2}$$

And finally the lifetime of the routing tree T (L^T) which is defined as the maximum number of rounds that this routing tree can be used is modeled as:

$$L^T = \min_{e_{ij} \in E(T)} \{\ L^T(e_{ij})\ \} = \min_{i \in V(T)} \{\ L^T(i)\ \} \tag{4.3}$$

Equation 4.3 is meaningful since a routing tree becomes unconnected if one of its nodes dies [85].

Now, suppose that we have given a set of nodes of cardinality n and its visibility graph (G). The problem is to find an efficient routing scheme to maximize the functional lifetime. Let us assume that we have only one chance to calculate a routing tree (*static routing scheme*). In this case we have two options: minimizing the total energy consumption and maximizing the lifetime.

With given definitions we can state that in order to yield the minimum energy tree once the set of nodes and hence the visibility graph (G) is given we should use the following optimization function:

$$T_{min}^G = \arg\min_{T \subset G} \left\{ \sum_{i \in T} W_i^T \right\} \tag{4.4}$$

And the optimal tree for maximum lifetime once the visibility graph G is given can be defined as:

$$T^G_{opt} = \arg\max_{T \subset G} \{ \ L^T \ \} \tag{4.5}$$

Computing T_{min} is actually finding the minimum spanning tree and hence it is polynomial. However finding T_{opt} is known to be NP [78]. However, we can improve the lifetime of a set of nodes if we use a dynamic routing scheme instead of using the same aggregation tree for the set. Finding optimum lifetime for a set of nodes is defined as maximum lifetime data aggregation (MLDA) in previous researches [33], and there is an extensive amount of work done on this topic as summarized in Section 3.4.

4.2 Upper Bound on First Node Failure Time

The lifetime definition in terms of first node failure - given in Equation 4.3 - is an important metric as mentioned in Section 2.4. First of all, it gives directly the maximum amount of time for which an application will gather data reliably. It also shows how balanced a rooting scheme handles the bottleneck nodes. It will also provide good basis for node scheduling algorithms. Since the node scheduling algorithms tend to put as many nodes as possible to sleep state so that the whole area of interest will be covered, usually they end up with a routing tree where every node plays an important role (sensing or relaying). In this case the routing tree cannot be used even when a node in the tree fails. Therefore, first node failure time is also important for coverage based lifetime definitions.

It is obvious that it is important to have an upper bound on first node failure time, in order to measure how well protocols designed for sensor networks perform. In this section we derive a theoretical upper-bound for the lifetime of the first failing node in a sensor network using tree-based routing. This upper bound will be used to test the performance of our protocols against a theoretical limit.

Theorem 4.2.1. *The lifetime of a sensor network with n nodes in terms of first failing node is upper bounded by*

$$\bar{L^G} = nE_0/E^G_{min}, \tag{4.6}$$

where E_0 is the initial energy of the nodes, E^G_{min} is the minimum possible total energy consumption for a round that can be achieved, and G is the visibility graph of given n nodes.

Proof. Let T^G_{min} be the tree that gives the minimum total energy consumption for any round. That is, it is a fixed tree which minimizes the total energy consumption for the network. If all the nodes will be used, this tree can be derived from the minimum spanning tree algorithm [48] on graph G by using the cost function given in Equation 2.3. It is also possible to use a subset of given nodes so that the energy consumption is minimum while satisfying the conditions for routing. We will investigate this condition in Section 4.3.

Let E^G_{min} be the total energy expenditure in using T^G_{min} as the routing tree. We can state that in any round the total energy consumption is $\geq E^G_{min}$.

Although the routing trees may change in each round, the total energy consumption in L rounds is always $\geq LE^G_{min}$. This implies that there exist at least one node whose energy consumption in L rounds is $\geq LE^G_{min}/n$. Since energy of each node is limited by E_0:

$$LE^G_{min}/n \leq E_0 \tag{4.7}$$

Consequently

$$L \leq nE_0/E^G_{min} \tag{4.8}$$

□

This upper bound is actually very intuitive, which states that the maximum achievable lifetime is equal to total initial energy in the system over minimum achievable energy consumption in a round. From this result we can conclude that for achieving maximum lifetime, the routing algorithm should try to minimize the energy consumption in a round while balancing the load among the nodes.

Theorem 4.2.1 states that, if all the nodes are used for routing, we can easily compute the upper bound $\bar{L^G}$ for any set of nodes, where we know the locations of the nodes, by just computing T^G_{min} which is the minimum spanning tree with edge weights as in Equation 2.3. The total cost of the minimum spanning tree gives us E_{min}^G, and since we know n and E_0 we can find the upper bound $\bar{L^G}$.

For static routing tree approach, it seems to be very difficult to achieve this upper bound, since the load on the nodes cannot be balanced in a static tree. As we will see in our simulations, the lifetime of a static method will be far from being optimal. Even the tree computed by the Equation 4.5 cannot give this upper bound.

In this work, however, instead of using a static routing tree, we propose dynamically changing the routing tree repeatedly in order to balance the load among the nodes over the time. Although optimal load balance (E_{min}^G/n) cannot be achieved for a single round, if we can use a good randomization scheme, we expect maximum average value for w_i to become closer to E_{min}^G/n and consequently lifetime becomes closer to $\bar{L^G}$.

4.3 Mathematical Analysis for Minimum Energy Expenditure

In previous section we derived an upper bound for maximum lifetime for data aggregation applications. Although the upper bound is very intuitive, E_{min}^G does not give any direct information. Therefore, in this section we will try to further investigate E_{min}^G to relate it to the number of nodes deployed, and try to yield an closed formula for optimum number of nodes to achieve minimum energy expenditure.

We already know that if all the nodes in G are used, in order to find E_{min}^G, we just need to compute the minimum spanning tree of G rooted at sink. However, for some applications the routing solution does not need us to use all the nodes, where only a subset of the graph can satisfy the requirements of the application (e.g. coverage conditions). In such scenarios, in order to find E_{min}^G we need to enumerate all subtrees, which satisfy the requirements, and choose the one with minimum energy consumption. We can restate this problem as enumerating all subsets of nodes, which satisfy the requirements, and for each subset compute the minimum spanning tree, and finally choose the one with minimum energy expenditure. So, again we end up with minimum spanning trees, and this leads us to investigate the characteristics of total cost in minimum spanning trees.

4.3.1 Total Length of Euclidean MST

In the literature, there are a small number of works which investigate the total length of the minimum spanning trees. Fortunately, there are two of them which are very related to our problem.

In [61] Steele examines the total cost of an Euclidean MST for points in a given hypercube in p dimensions, and proved the following lemma, which gives a closed formula for the total cost:

Lemma 4.3.1. *The total cost of an Euclidean minimum spanning tree with n nodes where link costs are powered to α is*

$$M_n^\alpha = \kappa_p^\alpha \times n^{(p-\alpha)/p} \textit{ for } 0 < \alpha < p \tag{4.9}$$

where p is dimension and κ_p^α is a constant proportional to α, p and the side-length of the hypercube (l).

This is an important lemma since it states that the asymptotic cost of an MST is equal to $\Theta(n^{(p-\alpha)/p})$. Interestingly this means that the total length of the MST only depends on the number of nodes n when the hypercube is fixed. So, for $p = 2$ and $\alpha = 1$, which is Euclidean MST in two dimensions, we get $M_n^2 = \kappa_2^1 \times \sqrt{n}$ which can be rewritten as $\Theta(\sqrt{n})$, which means the total cost of a n node minimum spanning tree with link costs as their Euclidean distances is directly proportional to $\sqrt{n}$.

Aldous and Steele later showed in [5] that when $\alpha = p$, then M_n^p tends to a constant as $n \to \infty$. This result is very important for us, since it states that the total cost M_n^2 (where $\alpha = p = 2$) remains constant regardless of the number of nodes.

Although the results given in [61] and [5] are important, we cannot conclude with any results since we are interested in conditions also where $2 < \alpha \leq 6$. Because of this, we conducted an experiment to see the characteristics of total cost of Euclidean MST where $1 \leq \alpha \leq 6$. In this experiment, we generated different networks deployed in a square area with side length of 1000m for different number of nodes ($10 \leq n \leq 1000$). For each n, we computed the cost of minimum spanning trees, where the link costs are powered to all α values ($1 \leq \alpha \leq 6$), and repeated the experiment 1000 times in order to achieve good approximations.

Our empirical results showed that for $p = 2$ and $\alpha > 2$ the total cost M_n^α is again $\Theta(n^{(2-\alpha)/2})$. This result can be explained by the following intuition: If for $p = 2$ and $\alpha = 1$ we get the total cost as $\Theta(\sqrt{n})$, we can state that the average length of an edge on Euclidean MST is approximately $\Theta(\sqrt{n}/n)$ for sufficiently big n values. If the link costs are powered to α, the cost of an edge becomes $\Theta((\frac{1}{\sqrt{n}})^\alpha)$. In this case M_n^α becomes approximately:

$$\begin{aligned} M_n^\alpha &\approx \Theta(n(\frac{1}{\sqrt{n}})^\alpha) && (4.10)\\ &= \Theta(\frac{n}{n^{\alpha/2}}) && (4.11)\\ &= \Theta(n^{1-\alpha/2}) && (4.12)\\ &= \Theta(n^{(2-\alpha)/2}) && (4.13) \end{aligned}$$

This result is actually interesting, since it states that the total cost of MST is *decreasing* exponentially with increasing number of nodes when $\alpha > 2$.

4.3.2 Total Energy Expenditure of MST

After deriving an approximate value for the total cost on Euclidean MST with link costs powered to α, we can calculate the total energy expenditure of the MST for the link costs given in 2.3. Let the total cost of a tree be defined as E_T.

E_T can be computed with the following formula:

$$E_T = \sum_{e_{ij} \in T} w_{ij}^T \tag{4.14}$$

If we substitute w_{ij} with the definition given in Equation 2.3 and use the assumptions used in yielding Equation 2.9 we get:

$$E_T = \sum_{e_{ij} \in T} 2sE_{elec} + sE_{amp} {d_{ip_i^T}}^{\alpha} \tag{4.15}$$

If we let $c = 2sE_{elec}$, and $a = sE_{amp}$, we can yield the following formula for E_T for large values of n:

$$\begin{aligned} E_T &\approx a \sum_{e_{ij} \in T} {d_{ip_i^T}}^{\alpha} + nc & (4.16) \\ &= aM_n^{\alpha} + nc & (4.17) \\ &= a\kappa n^{(2-\alpha)/2} + nc & (4.18) \end{aligned}$$

Equation 4.18 shows that the total energy expenditure of MST can be formulated as a function of the number of nodes (n). Considering that all the other factors of the formula are constants that we cannot change for a specific application, this formula is very useful for protocol developers.

4.3.3 Optimum Number of Nodes for Minimum Energy Expenditure

As we found in Equation 4.13, for $\alpha > 2$, M_n^{α} is a decreasing function of n. In Equation 4.18 we can see that second part of the equation is an increasing function of n. So we can expect an optimum number of nodes n_{min} which minimizes E_T and gives E_{min}^G. In order to find this value we should take the derivative of E_T and equate to 0. After a few mathematical operations we yield

$$n_{min} = \left(\frac{(\frac{\alpha}{2} - 1)a\kappa}{c} \right)^{2/\alpha} \tag{4.19}$$

This value implies that regardless of how many nodes we have, the optimum energy expenditure will be achieved if we can use approximately n_{min} nodes.

It is also worth to discuss about the special case where $\alpha = 2$. Substituting α with 2 in Equation 4.19 gives $n_{min} = 0$. This means that using minimum possible number of nodes while satisfying the application specific requirements (connectivity, coverage etc.) gives the optimum energy consumption when $\alpha = 2$.

4.4 Summary

In this chapter we developed useful mathematical models for efficient protocol development and evaluation for wireless sensor networks where advantages of perfect aggregation and power control are considered. This theoretical analysis forms the basis of our protocols proposed in this thesis. It is also used in the evaluation of our proposals.

Chapter 5

Power Efficient Routing

As mentioned in Section 2.1 the design of wireless sensor networks depends on the application requirements. In this chapter we will focus for a specific type of demand-driven application where all the nodes are informed to report their data to the sink in each round. The problem studied for this scenario is finding an energy efficient routing scheme for gathering all data at the sink periodically so that the lifetime of the network is prolonged as much a possible. The lifetime can be expressed in terms of rounds where a round is the time period between two sensing activities of sensor nodes. This problem is defined as maximum lifetime data aggregation problem (MLDA) in literature and presented in Section 4.1 in detail. In this problem all the nodes available are used in order to achieve maximum lifetime. So, we will not consider any type of sleep or node scheduling approaches in this part.

There are several requirements for a routing scheme to be designed for this scenario. First, the algorithm should be distributed since it is energy consuming to calculate the optimum paths in a dynamic network and inform others about the computed paths in a centralized manner. The algorithm must also be scalable. The message and time complexity of computing the routing paths must scale well with increasing number of nodes. Another desirable property is robustness, which means that the routing scheme should be resilient to node and link failures. The scheme should also support new node additions to the network, since not all nodes fail at the same time, and some nodes may need to be replaced. In other words, the routing scheme should be self-healing. The final and possibly the most important requirement for a routing scheme for wireless sensor networks is energy efficiency.

Our previous study PEDAP [64] mentioned in Section 3.4 showed that using MST based approach can improve lifetime dramatically for this specific scenario. The most important disadvantage of that work, however, is its centralized nature. In this work we inspired from that study and we propose a localized version of PEDAP, which tries to combine the desired features of MST and shortest weighted path based gathering algorithms. We also expand the idea and propose a new family of localized protocols for the power efficient data aggregation problem. Our main concern is to satisfy all the requirements stated above. We name our new approach *Localized Power Efficient Data Aggregation Protocol* (L-PEDAP).

Our proposed routing approach consists of two phases. In the first phase, it computes a sparse topology over the visibility graph of the network in a localized manner. The topology needs to be efficiently computed by using only the one-hop neighborhood information. In the second phase, it computes a data gathering tree over the edges of the computed sparse topology.

For the first phase, we propose the use of two different sparse topologies in a distributed manner, namely local minimum spanning tree (LMST) [38] and relative neighborhood graph (RNG) [69]. These structures are already available in the literature. They are supersets of MST and can be efficiently computed in a localized manner. Section 3.3 gives a detailed information about the desired properties of these structures and how they are computed.

After computing the structure, in the second phase, we need to find a power efficient routing tree to gather the data from nodes to the sink. For the second phase, we propose three different methods and provide performance results of them. All of the methods are based on flooding a special packet using only the edges of the computed structure. According to the decisions made during this flooding process, the tree is yielded. These three methods that can be executed at a node for choosing the parent node toward the sink are to choose: 1) the first node from which the special packet is received, 2) the node that minimizes the number of hops to the sink, and 3) the node that minimizes the total energy consumed over the path to the sink.

Our solution can also handle new node arrivals and departures of existing nodes. Hence it is adaptive. The routing path is maintained when those dynamic conditions occur.

We also propose power-aware versions of our protocols that consider the dynamic changes in the remaining energy levels of nodes while constructing the sparse topologies and routing trees. For this, we actually needed only to take the idea of re-constructing the tree over time from the PEDAP-PA protocol [64] and make it work in a distributed manner.

For each simulation we made we used the upper-bound derived in Section 4.2 to see how close our protocols are to the theoretical limit. The simulation results showed that our protocols can achieve up to 90% of the upper bound.

To sum up, in this chapter, we present a localized, distributed, self organizing, robust and energy efficient data aggregation approach, which we call Localized Power Efficient Data Aggregation Protocol (L-PEDAP). Our approach is based on topologies, such as LMST and RNG, that can approximate minimum spanning tree and can be efficiently computed using only position or distance information of one-hop neighbors. The actual routing tree is constructed over these topologies. We also consider different parent selection strategies while constructing a routing tree. We compare each topology and parent selection strategy and conclude that the best among them is shortest path strategy over LMST structure. Our solution also involves route maintenance procedures that will be executed when a sensor node fails or a new node is added to the network. The proposed solution is also adapted to consider the remaining power levels of nodes in order to increase the network lifetime. Our simulation results show that by using our power-aware localized approach, we can almost have the same performance of a centralized solution in terms of network lifetime, and close to 90% of an upper bound derived in Section 4.2.

Preliminary conference version of this work appeared in [65] and the journal version appeared in [66].

5.1 Proposed Solution

As mentioned in Section 3.4, the network lifetime can be considerably extended by gathering packets on the MST structure. However, computing MST requires a global knowledge about the network and has a high cost when it is attempted to be computed in a distributed manner. A well known distributed MST computation algorithm given in [22] requires $5N \log N + 2E$ messages exchanged during the execution where N is the number of nodes and E is the number of edges. The worst case time complexity of this algorithm is $O(N \log N)$.

On the other hand, the cost of route computation is lower if we use shortest weighted path based approaches. However, the shortest weighted path based approaches can not always provide a good lifetime since they can not balance the load among the nodes [14, 15, 31, 47, 56, 62]. Our aim is to combine the energy efficient features of the MST with the distributed nature of shortest weighted path based routing schemes, in order to efficiently and locally compute the routing paths that can also provide a superior network lifetime.

5.1.1 Our Approach

Our approach for solving the power-efficient routing problem in wireless sensor network for data gathering applications in a distributed manner consists of two phases: 1) distributed and localized sparse topology construction; and 2) distributed aggregation/routing tree computation.

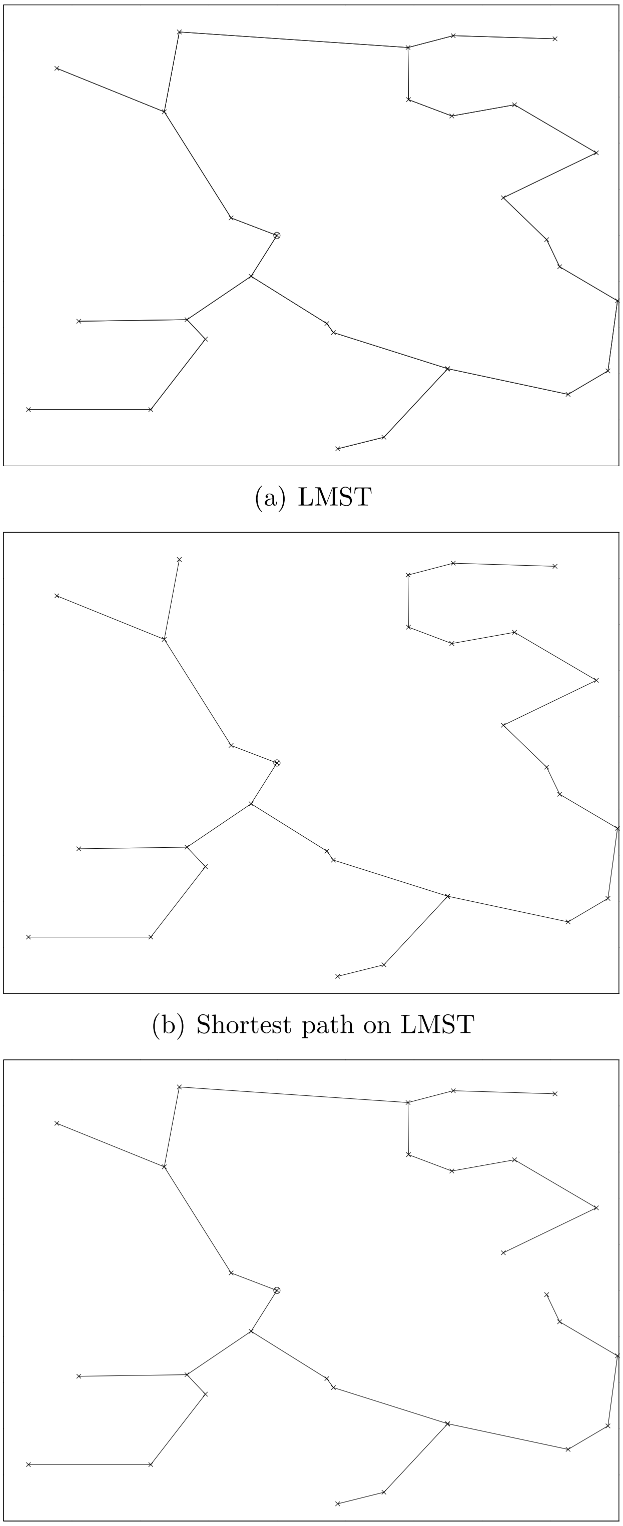
5.1.1.1 Sparse Topology construction

In this phase, we aim to construct a sparse and efficient topology over the visibility graph of the network in a distributed manner. We have different alternatives for sparse topologies that can be efficient for energy-aware routing. In this work, we choose to investigate the use of RNG and LMST and we compare their relative performance. We expect that LMST performs better than RNG because it is sparser. However there are some aspects that make RNG and LMST comparable. First, the computation of RNG is more efficient than of LMST. RNG needs only the location information of 1-hop neighbors, whereas LMST needs a second message for informing about the LMST neighbors. This second message contains the local MST neighbors of the nodes and hence it is larger in size compared to the first message which contains only the location information. On the other hand, one advantage of LMST is that it can approximate MST well especially when the density is high.

As mentioned in Section 3.3, in both topologies, we can use arbitrary cost functions. For instance both can use the same cost functions (Equation 2.3) used in PEDAP. Figure 3.3(a-c) shows MST, LMST and RNG structures with these costs. As seen in the figure, LMST is sparser than the RNG structure. Also we can use power-aware cost functions and consequently we can efficiently approximate PEDAP-PA. As mentioned in Section 3.4, PEDAP-PA algorithm recomputes the routing tree in every 100 rounds by using an asymmetric cost function given in Equation 3.1, by applying Prim's minimum spanning tree algorithm [48]. However, our algorithms need a symmetric cost function to compute LMST and RNG. Consequently, we changed the power-aware cost function to the one given below for our dynamic case:

$$w^*_{ij} = w_{ij}/(r_i \times r_j). \tag{5.1}$$

For the rest of this work, whenever we refer to PEDAP-D or LMST-D, we mean the structures that are computed using the link costs given in Equation 5.1. Our simulation results with these cost functions showed that our dynamic approach is a good randomization scheme.



(a) LMST

(b) Shortest path on LMST

(c) Minimum Hop path on LMST

Figure 5.1: Comparison of different route computation techniques.

5.1.1.2 Routing tree computation

There are several methods for obtaining a tree structure (spanning all the nodes) given a graph. In this work we use a flooding based tree construction algorithm. A special route discovery packet is broadcasted by the sink and when a node receives that packet it decides its parent according to the information in the packet. After selecting the parent it rebroadcasts the packet. The details will be given in Section 5.1.2. Here we investigate the efficiency of three different methods: first parent path method (FP), nearest minimum hop path method (MH), and shortest weighted path (i.e. least cost) method (SWP).

The FP method is the simplest among the three. In this method, a node will set its parent as the first neighboring node (among neighbors in selected sparse structure) from which the special route discovery packet was received.

In the MH method, the node chooses its nearest neighbor among those with minimum hops to reach to the sink. So, the node updates its parent only if the sender node has a smaller hop count or has the same hop count as the current parent but it is closer than the current parent (among neighbors in selected sparse structure). Otherwise, the packet is ignored.

The SWP method tries to yield a tree that minimizes the cost of reaching the sink for each node. The details will be given in the next section.

At first glance, we expect these three algorithms to give almost the same performance for approximating the minimum spanning tree, since the topologies are sparse enough. However, this is not always the case. Since we use a cost function that uses a power of the distance, minimum hop method cannot give always the most efficient tree. Intermediate nodes at closer distances can make the packet transmission more efficient [62]. Consider the LMST of a sample network given in Figure 5.1(a). Note that the sink is at the center. Only one edge removal yield a tree. As seen in the Figure 5.1(c), the longest edge is kept by the minimum hop method since choosing that edge reduces the hop count of children nodes toward the sink. However, the shortest path algorithm yields the

Algorithm 1 Topology and Route Computation

```
Send HELLO message
Collect HELLO messages for $t_{hello}$
Reset Parent ($\pi \leftarrow null$)
Compute neighbors on the sparse topology
while ROUTE-DISCOVERY packet $RD$ received
in $t_{discovery}$ do
    if update required for $RD$ then
        Update parent ($\pi \leftarrow source(RD)$)
        Broadcast ROUTE-DISCOVERY
    end if
end while
Inform $\pi$ to construct its child-list
```

same tree as MST since having closer nodes reduces the total transmission cost especially when the power of the distance is high and consequently longer paths in terms of hop count can be more efficient than the shorter ones.

5.1.2 Algorithm Details

In our proposed routing scheme, at any time each sensor node has to know its all one-hop neighbors and their locations, the neighbors on the computed topology, the parent node that it will send the data to in order to reach the sink, and the child nodes that it will receive the data from before it sends the fused or aggregated packet to its parent node. Our solution consists of three parts: *Route Computation*, *Data Gathering* and *Route Maintenance*.

5.1.2.1 Topology and Route Computation

The main goal in this phase is to find a sparse topology and setup the routes over it, which means determining the children and parent nodes for each node. At the end of this phase a data aggregation tree rooted at sink is constructed. The pseudo code for this phase is given in Algorithm 1.

Initially, the nodes and the sink are not aware about the environment. In the setup phase, all nodes and the sink broadcast HELLO messages, which include their location and remaining energy, using their maximum allowed transmit power. The remaining energy level is advertised only when dynamic (power-aware) protocols are used. We give a time threshold t_{hello} for waiting advertisements, which must be long enough to hear all possible advertisements. After receiving HELLO messages, all nodes are informed about their one-hop neighbors and their locations and energy levels. Each node can then locally compute its neighbors in the desired sparse topology (static and dynamic versions of RNG and LMST). After finding its neighbors in the sparse topology, a node can join the distributed route computation process in order to find its parent and children on the aggregation tree.

In this step, if dynamic method is used, we compute the LMST-D structure instead of LMST. This means we use the cost functions given in Equation 5.1. Similarly we compute RNG-D structure. If static case is used, then LMST and RNG are computed using the cost functions given in Equation 2.3.

The route computation is done via a broadcasting process which starts at the sink node. The sink initiates a ROUTE-DISCOVERY packet in order to find and setup the routes from all sensor nodes toward itself. When a sensor node receives a ROUTE-DISCOVERY packet, it broadcasts the packet to all its neighbors on the computed topology if it updates its routing table. By this way the routing tree rooted at the sink is established over the sparse topology. An important energy conserving feature of our algorithm is that the packet is sent with a power just enough for reaching all the neighbors on the sparse topology instead of using the maximum power.

Each ROUTE-DISCOVERY packet has three fields: a *sequence ID*, which is increased when a new discovery is initiated by the sink, an optional *distance* field which shows the cost of reaching the sink, and an optional *neighbor list* field which is the list of the neighbors of the sending node in the chosen topology. The *distance* field is not required if FP algorithm is chosen. It holds the minimum number of hops or minimum energy cost to reach the sink, respectively, if MH or SWP algorithm is chosen. The *neighbor list* field must only be used if the LMST topology is chosen. So if we use the FP on RNG topology we can decrease the message overhead. On the other hand if we use SWP on LMST, which gives the best performance in all cases according to our experiments, we have to have some overhead. But an important point to mention is that in our approach since the LMST computation is combined with the route computation, no extra messages are used for negotiation among LMST neighbors. Only overhead is the size of the `ROUTE-DISCOVERY` packet.

Upon receiving a new `ROUTE-DISCOVERY` packet, the sensor node ignores the packet if it is not coming from a direct neighbor, in order to ensure using only the edges in the computed topology.

After that, according to the routing strategy chosen, the node decides whether or not to update its parent. If FP strategy is used, the node updates the parent information only if it has not a parent yet. In MH strategy, the node compares its current parent with the sending node and chooses the sender as its new parent if it has a smaller hop count to the sink or has the same number of hops but is closer to the node. And finally, if the SWP is chosen, the node updates its parent only if the path using the sender node is advantageous in terms of total energy consumption. Regardless of the chosen strategy if the packet has a higher sequence ID the node directly updates its parent. If the node decides to update its parent it rebroadcasts the `ROUTE-DISCOVERY` packet with updated fields. If in the time threshold $t_{discovery}$, no other route discovery packets are received we can conclude that the route setup converged.

At this step each node can inform its parent, in order to construct the children-list which will be used in data gathering phase. After this final step, the data aggregation tree is setup and stabilized. This means that each node knows from which neighbors it will receive data and to which node it will send the received data after aggregation.

5.1.2.2 Data Gathering

After the parent and children nodes for an individual sensor node are determined, the node can join the data gathering process. In data gathering phase each sensor node periodically senses its nearby environment and generates the data to be sent to the sink. However, before sending it directly to the parent node, it will wait all the data from its child nodes and aggregate the data coming from them together with its own data, and then send the aggregated data to the parent node. Thus, at the beginning of data gathering step, only leaf nodes can transmit their data to their corresponding parent nodes. At each step the data is gathered upwards in the tree and reaches the sink after h steps, where h is the height of the aggregation tree. The reason for waiting to receive data from child nodes is to use the advantage of the aggregation. In this way each sensor only transmits once in a round, and as a result saves its energy.

5.1.2.3 Route Maintenance

After setting up the routes, three events can cause a change in the routing plan: route recomputation, node failure and node addition. We will discuss them separately.

Recomputation of the aggregation tree is required when power-aware (dynamic) cost functions are used given in Equation 5.1. In power-aware methods, the tree must be recomputed at specified intervals. Since the computation depends on the remaining energy of nodes, each time the computation takes place, a different and more power efficient plan is yielded. In our case, we handle this requirement by broadcasting a new `ROUTE-DISCOVERY` packet with a new sequence ID. Apparently, in order to utilize the power aware methods, each node must know the remaining energy levels of its neighbors. In order to exchange the remaining energy levels, we use `HELLO` messages. So at the beginning of each recomputation phase, the nodes advertise their remaining energy levels. After that, `ROUTE-DISCOVERY` packet with a new sequence ID can be broadcasted by the sink. It is worth to mention that in order to achieve recomputation, each node

Algorithm 2 Route Recovery

```
$\pi_{old} \leftarrow \pi$
if BYE message $B$ received then
    remove $source(B)$ from neighbor list
    compute the sparse topology
    if $source(B) = \pi$ then
        Reset parent ($\pi \leftarrow null$)
        Reset child list
        Broadcast PARENT-DISCOVERY message
        Enter route discovery phase
    end if
end if
if PARENT-DISCOVERY message $PD$ received then
    if $source(PD) = \pi$ then
        Reset parent ($\pi \leftarrow null$)
        Reset child list
        Broadcast PARENT-DISCOVERY message
        Enter route discovery phase
    else
        if $\pi \neq null$ then
            Send ROUTE-DISCOVERY
        end if
    end if
end if
if $\pi \neq \pi_{old}$ then
    Inform $\pi_{old}$ and $\pi$ to construct their child-list
end if
```

must know the predefined time (in terms of rounds) to send HELLO messages.

Node failures can be due to various reasons. However, the most critical reason is depletion of energy of a node. Previous approaches (e.g. [26, 39, 64]) did not discuss the node failure problem. In these approaches, however, a node failure in communication phase will cause a routing problem in which the descendants of the failed node cannot send their data until next setup phase. In order to prevent this, failures must be handled as soon as possible. In our solution, we handle the case where failures are due to energy depletion. However, the idea behind the solution can be applied to other failure causes as well.

Failure of a node due to energy depletion can be handled gracefully, since the node can predict that it will die soon due to energy limitation. Algorithm 2 presents the route recovery algorithm. In our solution, when a node's energy reduces below a threshold value, which can be set to a very small value, the node broadcasts a BYE message using maximum allowed transmit power. All nodes receiving the BYE message will immediately update their local structure. This message is not required to be retransmitted since the node failures do not affect the structure globally. However, in this case the nodes that cannot reach the sink because of the energy depletion of their ancestor must find a new cost-efficient path to send their packets. In our solution, this is handled in a localized manner as follows. The child nodes of the failed node that receive the BYE message reset their routing tables and enter the parent-discovery phase by broadcasting a special message PARENT-DISCOVERY to its neighbors on the structure. According to the receiver of that special message, if the sender is its own parent on the way to the sink, the receiver also reset its routing table and broadcasts the packet to its neighbors. In this way, all the nodes that should enter the parent discovery phase will be reached. If the PARENT-DISCOVERY packet is received by a neighboring node of the sender and if it has a valid parent, the receiver constructs a new ROUTE-DISCOVERY packet as mentioned above and broadcasts it to the sender. This ROUTE-DISCOVERY packet is handled as mentioned in Section 5.1.2.1. It is worth to mention that the sequence ID in this new packet is not incremented, therefore the update of the routing table takes place only when the newly received cost is smaller. After the route discovery phase converges, the new routes are set up and data gathering can continue.

Consider now the case of node additions. When a new node is deployed, it broadcasts a HELLO message. Its neighbors update their local structure upon receiving this message, and also inform the new node about their existence and locations by replying a HELLO message so that the newly deployed node can also determine its neighbors. Nodes that update their local structure send back a ROUTE-DISCOVERY packet including their costs to the newly deployed node. The new node selects the most efficient node as its parent, and broadcasts this information by a new ROUTE-DISCOVERY packet. Since the sequence ID is again not incremented, the new packet is broadcasted throughout the network only

Table 5.1: Summary of the messages used in PEDAP

Message	Purpose
HELLO	Advertisement of existence
ROUTE-DISCOVERY	Discovery of routing paths
BYE	Advertisement of failing nodes
PARENT-DISCOVERY	Advertisement for nodes with no parent

when using the new node is advantageous. This final step can be avoided if FP method is used. So the newly added node just chooses its closest neighbor as its parent and starts sending data.

If node failures and node additions occur frequently, within a small time period, a recomputation of whole aggregation tree will be more efficient, instead of handling each event separately. In order to realize such a solution only action that a node must take is to piggyback a *REROUTE* bit to the data packet sent to its parent if it has still one parent. Upon receiving a data packet with a *REROUTE* bit piggybacked to it, the sink constructs a new `ROUTE-DISCOVERY` packet and initiates a new route computation phase. Table 5.1 gives summary of the messages used in PEDAP and their purposes.

5.2 Simulation Results

In this section we will first try to choose the best parent selection strategy, and then continue the experiments with that strategy, since running the experiments with all topologies and strategies will become too complicated.

For our scenario there are three parameters that we can change to see their effect: number of nodes N, maximum transmission radius R and side-length of the square area l. One other parameter that depends on these three parameters and that gives direct intuition about the scenario is the density ρ, which is defined as the average number of neighbors per node. For the sake of completeness we will give the value of ρ for each scenario since it is immediately very informative.

Table 5.2: Comparison of algorithms - Normalized lifetime N:100, R:20, l:100, ρ:10

	RNG			LMST		
α	SWP	MH	FP	SWP	MH	FP
2	0.917	0.910	0.910	0.991	0.990	0.989
4	0.808	0.401	0.384	0.907	0.744	0.737

We generated a network with parameters $N = 100, R = 20m, l = 100m \Rightarrow \rho = 10$. On this network we repeated the experiments on LMST and RNG topologies with three alternative parent selection strategies. We compared the methods in terms of the lifetime they provide for the first node (normalized lifetime) and how well they approximate the PEDAP tree (approximation percentage). Normalized lifetime is the ratio of the lifetime to the lifetime provided by PEDAP for the same network whereas approximation percentage is the ratio of the number of common edges with PEDAP tree to the total number of edges.

Tables 5.2 and 5.3 provide the results of experiments that compare the efficiency of three parent selection methods. Here α denotes the power of distance in the cost function. We can conclude with these results that if the propagation is free space ($\alpha = 2$), using first parent (FP) algorithm on RNG can be advantageous because the setup cost is minimal in that case and the performance is almost the same as in the best solution SWP on LMST ($< 9\%$ lesser lifetime). If $\alpha = 4$, however, choosing shortest weighted path on LMST gives considerably better performance in terms of lifetime. We can also see that the difference among parent selection strategies is more striking when $\alpha = 4$. These results also show

Table 5.3: Comparison of algorithms - Approximation percentage N:100, R:20, l:100, ρ:10

	RNG			LMST		
α	SWP	MH	FP	SWP	MH	FP
2	0.686	0.682	0.666	0.897	0.895	0.894
4	0.825	0.629	0.613	0.927	0.869	0.869

Table 5.4: Upper bound for FNF - R:20, l:100

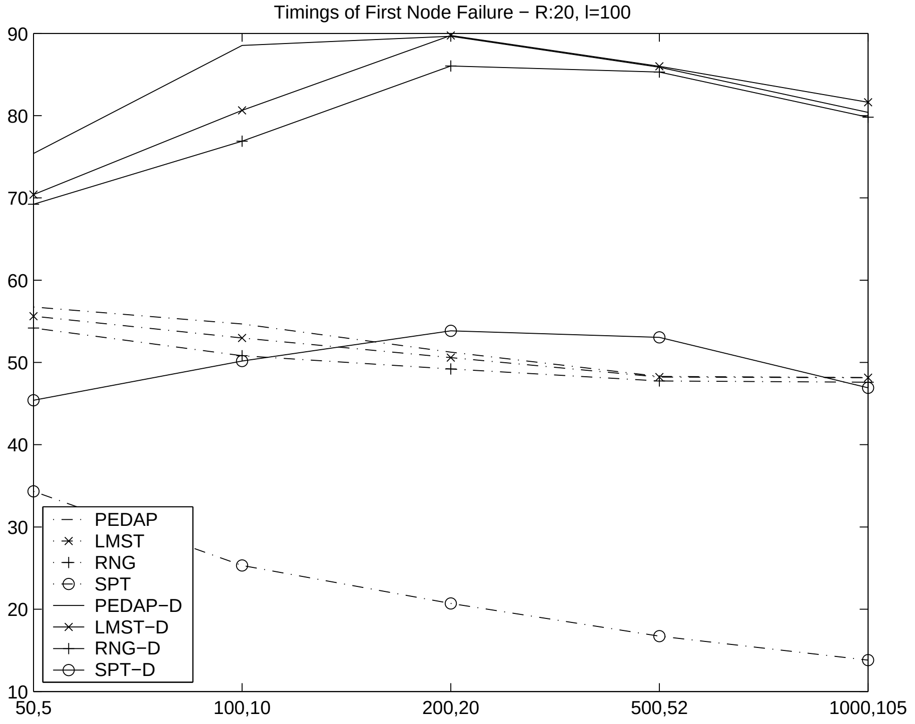
N, ρ	50, 5	100, 10	200, 20	500, 52	1000, 105
OPT	4748	4949	5063	5125	5165

that the SWP strategy outperforms its alternatives in each case. Therefore for the rest of the simulations we always use SWP approach to compare performances of different topologies.

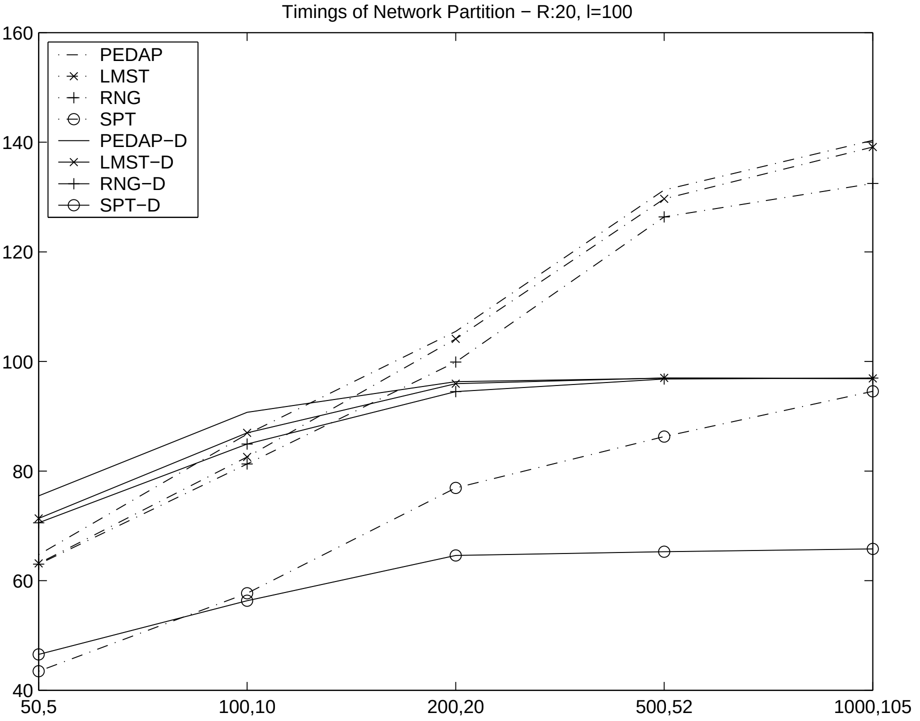
The rest of the simulations evaluates the performance of our routing scheme. We conducted experiments with different values of N, R and l. For each parameter we ran the experiments 100 times and obtained an average value for the two evaluation metrics: First Node Failure Time (FNF) and Network Partition Time (NPT). The initial energies of the nodes were given as 0.5 J. For dynamic algorithms (e.g., PEDAP-D, LMST-D), we used the power-aware cost functions given in Equation 5.1 and we recomputed the routing paths every 100 rounds. For transmission costs, we used the parameters of the first order radio model given in [26]. In all of the routing schemes, data aggregation is used at every step for a fair comparison. Also, for all methods, the setup and maintenance costs are not included in energy expenditures, which means that only the cost of data packets is considered. We used a fixed value of 1000 bits for data packet size k.

In order to compare our algorithms based on LMST and RNG, we also implemented the centralized PEDAP algorithm and the shortest weighted path tree (SPT) as other alternatives. With the dynamic versions it adds up to 8 different methods to compare (PEDAP, LMST, RNG, SPT, PEDAP-D, LMST-D, RNG-D, SPT-D).

Since the most informative parameter for our scenarios is ρ, we try to investigate the performances on different values of ρ. However there are three ways of changing ρ: for each of the parameters N, R and l we can keep two of them fixed and change the third one. One important point is that in the rest of simulations we give the results normalized to upper bound $\bar{L}$ which is computed as given in Section 4.2. In all cases we provide the actual values of $\bar{L}$.



(a) Normalized FNF Timings vs (N, d)



(b) Normalized NPT Timings vs (N, d)

Figure 5.2: Effect of number of nodes on network lifetime for various data gathering schemes.

Reprinted with permission from IEEE Transactions on Parallel and Distributed Systems Vol. 22, No.3, March 2011 pp. 489-500.

Consider the impact of the number of nodes N on the lifetime. In Figure 5.2(a) we can see the normalized lifetimes for various values of N in terms of first node failure (FNF). Since the upper bound for each case is different, we give the exact values of the upper bound in Table 5.4. We can see that the upper bound slightly increases with increasing N. We observe that the effect of N is not much significant in static MST based approaches. For the static SPT approach, however, increasing ρ decreases the lifetime of the system. This is mainly because of the fact that the SPT approach can not balance the degree of a node. So if N is increased, the maximum node degree will also increase. However, in MST based approaches, since the maximum node degree is bounded by 6, the decrease is not much significant. On the other hand, in all MST based approaches, the maximum node degree is increasing (from 3-4 to 4-5) with increasing ρ and thus the network lifetime is slightly decreased when N is increased.

Next, we study the impact of N on the dynamic versions of algorithms (PEDAP-D, LMST-D, RNG-D, SPT-D). When N increases, the lifetimes increase until reaching a maximum and decreases afterwards. Since the dynamic versions of the algorithms almost balance the degree among the nodes, this behavior is due to the distances between the neighbors. In low density case, the distances are long, and since the overhead because of the distance is exponential, the lifetime is far from being optimal. As ρ increases, the average distance among the nodes become closer to the optimal distance - which may be the same as given in [8, 62]. After some point, however the decrease in distances have a negative effect due to the constant cost of wireless transmission. So, we can conclude that using too many nodes is not always very effective in providing longer network lifetime. If we compare RNG and LMST based approaches, RNG gives very close results with LMST, but LMST performs always slightly better than RNG. At their best, both PEDAP-D and LMST-D achieve 90% of the upper bound.

In Figure 5.2(b), the lifetimes in terms of network partition time are given, normalized to the values given in Table 5.4. Again, as expected, the lifetime improves with increasing ρ in static versions of the protocols. However, for the power-aware (dynamic) methods, the increase is smaller. This is explained by the fact that, in order to provide longer lifetime in terms of FNF, the system uses

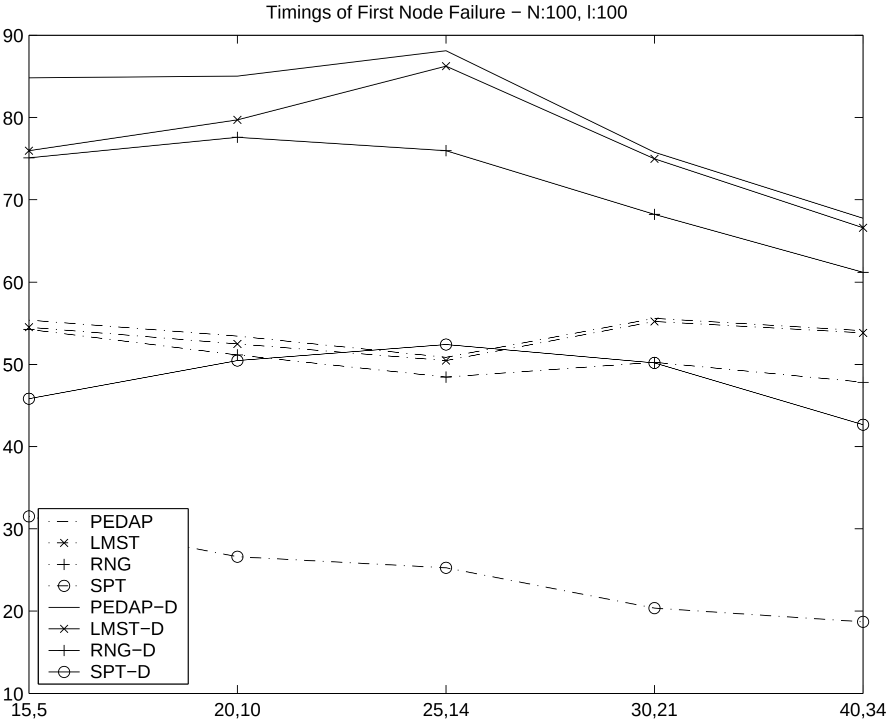
Table 5.5: Upper bound for FNF - N:100, R:20

l, d	50, 33	75, 17	100, 10	125, 6	150, 4
OPT	5871	5256	4983	4776	4576

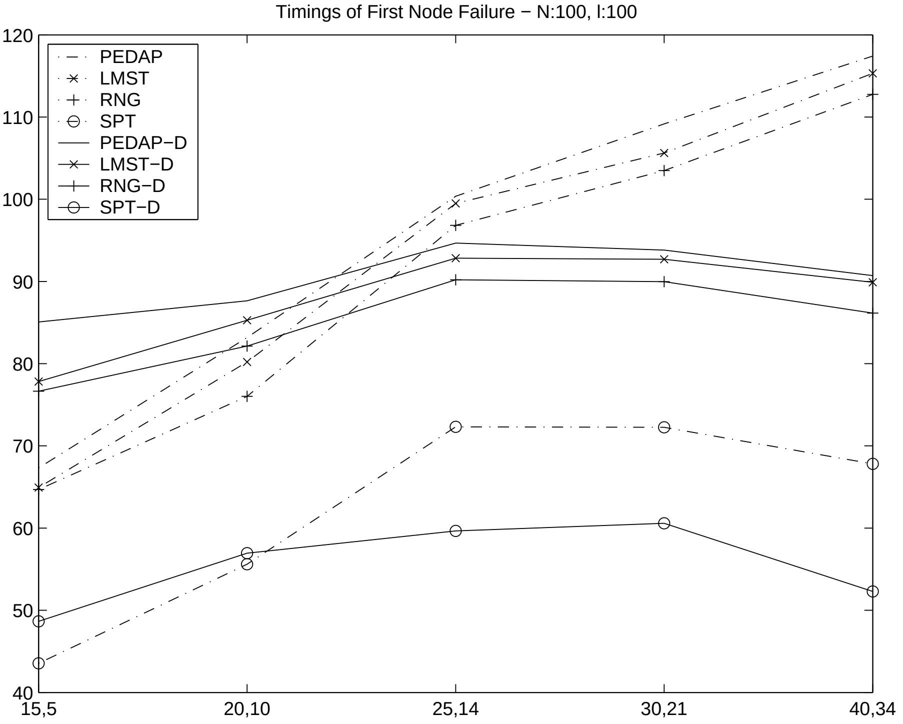
more resources.

The lifetimes for different R values are given in images 5.3(a) and 5.3(b). As it can be seen in Figure 5.3(a), increased transmission radius dramatically reduces the lifetime of the dynamic methods after some point. The maximum value is achieved when $R = 25m$. This can be explained by the effect of the distance to parent. With increasing R, although there exist more alternative nodes to choose, the average distance of the alternatives also increases. So, the nodes will tend to send to long distances as the residual energy of the neighbor nodes decrease, and this will cause a decrease in FNF. So we can say that increasing the radius above some point has an inverse effect on lifetime for our dynamic approach. The dynamic versions may give the best performance when R is chosen equal to the same optimal distance mentioned above. One important point here is that the upper bound for the lifetime is always the same in this scenario, since increasing R does not effect the MST topology. So, the E_{min} is constant with increasing radius. On the other hand, similar results are observed for the network partition times as in the previous case (Figure 5.3(b)).

Another scenario that changes the density is increasing the area size while keeping graph with parameters N and R the same. The Figure 5.4(a) shows the normalized simulations results for this case. The upper bounds of each specific case is given in Table 5.5. The optimal value of the same graph is decreasing with increasing area size, since the average distance among the nodes is increasing. However, if we normalize the lifetime, we observe that for the static methods the normalized lifetime is slightly increasing with decreasing density. If the dynamic versions are examined, above some density the PEDAP-D and LMST-D methods can achieve 90% of the upper bound. With decreasing density, after some point the lifetime decreases dramatically. This is expected since when there are more alternative neighbors to choose, our dynamic version can balance the load among

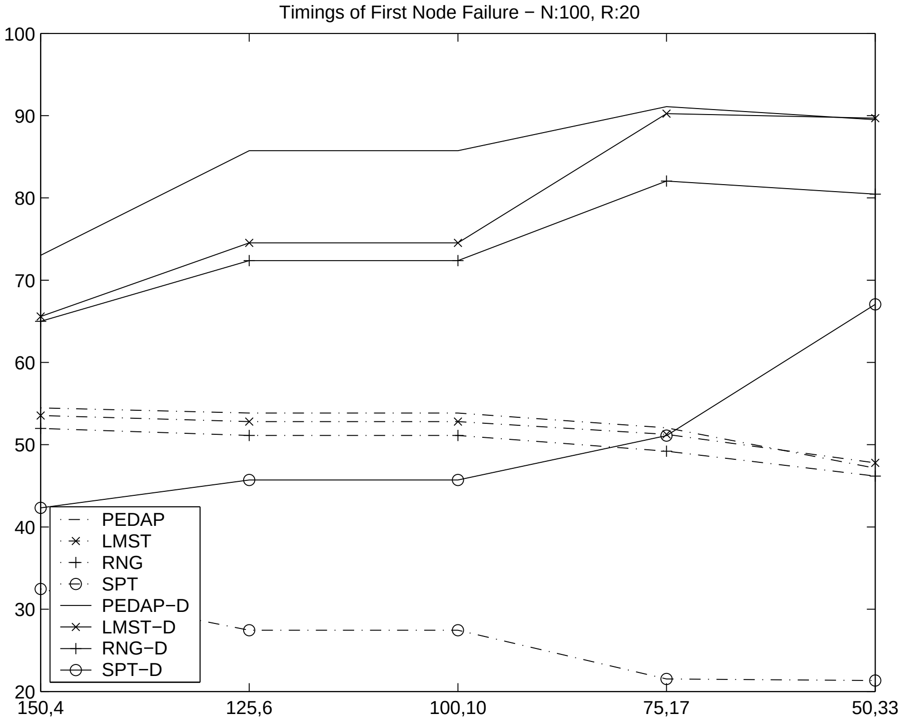


(a) Normalized FNF Timings vs (R, d)

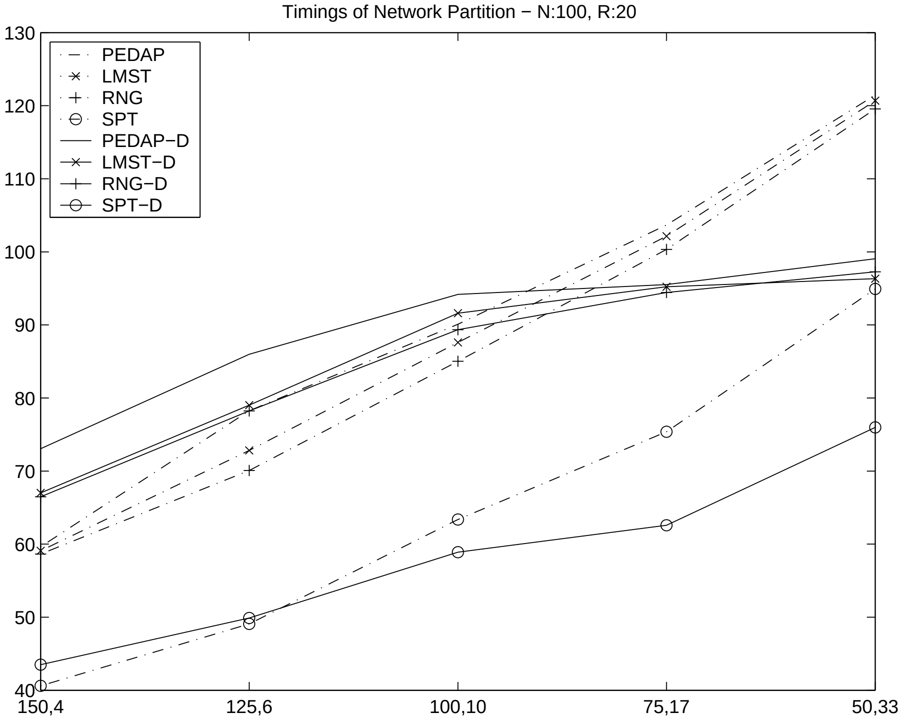


(b) Normalized NPT Timings vs (R, d)

Figure 5.3: Effect of transmission radius on network lifetime for various data gathering schemes.



(a) Normalized FNF Timings vs (l, d)



(b) Normalized NPT Timings vs (l, d)

Figure 5.4: Effect of area size on network lifetime for various data gathering schemes.

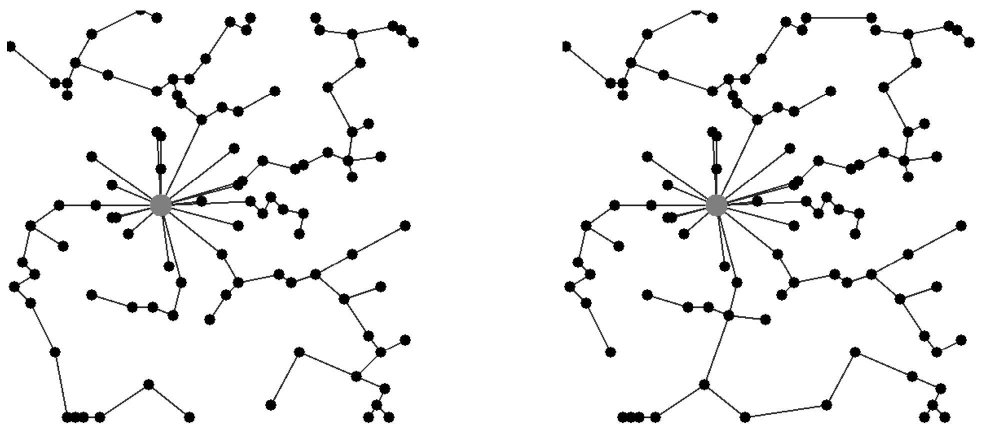
the nodes. If the density is low, the number of alternative routing trees becomes also small. This fact combined with the distance effect reduces the system lifetime considerably on wide networks. The reason of the decrease in lifetime on high density networks is that as the area size becomes smaller, the effect of the distance gets smaller. Similar to the first scenario, the degree plays more important role to determine the lifetime. So as in the first case, the maximum degree in increased slightly and the overall lifetime decreases.

We can observe similar result also for the NPT timings (Figure 5.4(b)). As the area enlarges, connectivity decreases, and distances get longer. This leads to a decrease in NPT timings.

In summary, the behavior of the NPT timings is similar in all three scenarios. This is because the NPT timing is directly related with the connectivity. With increasing density the network becomes more connected and the network partition time gets longer. However, we cannot see a single pattern for FNF timings in different scenarios. In the first scenario, the effect of the constant in wireless communication takes place and after some point the lifetime starts decreasing. In the second case, as the density increases the resulting trees become the same, thus the lifetime is the same after some point. However, since the average distance of neighbors increases, the lifetime becomes worse with increasing density. In the last scenario, the increase in density reduces the effect of the distance, and maximum degree determines the lifetime. The only common pattern is that with increasing density the FNF lifetime starts decreasing after some point.

Finally, we will present a performance comparison of our algorithm with previous routing schemes. In order to evaluate the performance of our algorithm we implemented six different routing schemes: MST, LMST, minimum energy route obtained by constructing shortest weighted path tree with edge weight accounting for transmission and reception powers (SPT), minimum hop route, minimum battery cost routing (MBCR) [68], and maximum residual energy route (MREP) [15].

For this part of our simulations we generated networks of diameter 100 m. We repeated the experiments for sensor networks having 50 and 100 nodes. We fixed the maximum transmission range of the sensors to 20 m. The initial energies of



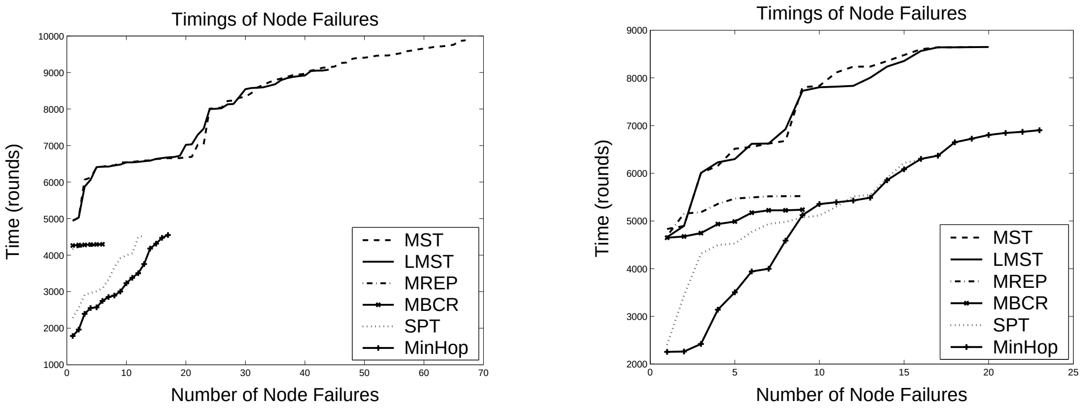
(a) MST based routing scheme. (b) LMST based routing scheme.

Figure 5.5: Sample aggregation trees for MST and LMST based routing.

the nodes were given as 1 J. For MBCR and MREP we recomputed the routing paths every 100 rounds as we do for our power aware algorithms.

Figure 5.5 presents the computed routing paths for MST and LMST based routing schemes for a sample network. One important point in the figures is that the nodes closer to the base station tend to send directly to the sink instead of choosing a closer neighbor. This is mainly because we include the receive cost in the link cost. As given in Equation 2.3 there is no receive cost for the base station. If a node closer to the sink transmits to its closest neighbor, the transmitted node should also spend the energy required to receive one more packet. So if the MST's are computed with Equation 2.3, the nodes for which it is advantageous to send directly to the sink will choose the sink as their parent. As seen in the figures, our proposed algorithm approximates the original MST quite well. This is as expected since as stated in [42] the LMST structure has only 5% more edges compared to MST.

Figure 5.6(a) and Figure 5.6(b) show the timings of all node failures until the network is partitioned for networks of sizes 100 nodes and 50 nodes respectively. As depicted in the figures, MST gives a very good lifetime in terms of both first node failure and network partition time. Our LMST based approach gives



(a) 100 nodes

(b) 50 nodes

Figure 5.6: Timings of node failures for various data gathering schemes.

Copyright 2011 © IEEE.

Reprinted with permission from IEEE AINA'07, May 2007 pp. 220-227.

almost the same results with MST based routing whereas in dense networks MST performs quite better. The power aware methods (MREP and MBCR) give better lifetime for first node compared with other shortest path based methods whereas minimum hop and minimum energy routing provides a connected network for a longer time compared to power aware methods.

Table 5.6 summarize the results for two different networks of sizes 50 and 100. In the tables AVG stands for the average node lifetime, FNF is the first node failure time, NPT shows the network partition time, and CNT stands for

Table 5.6: Statistics on lifetimes for different network sizes.

	100 nodes				50 nodes			
	AVG	FNF	NPT	CNT	AVG	FNF	NPT	CNT
MST	8191	4942	9881	67	7361	4832	8648	19
LMST	7497	4942	9077	44	7359	4662	8648	20
MREP	4240	4163	4282	6	5323	4685	5522	9
MBCR	4283	4262	4297	6	4983	4648	5236	9
SPT	3447	2283	4532	13	4926	2403	6278	16
MinHop	3185	1787	4550	17	5144	2256	6903	23

number of failed nodes when the network is partitioned. The simulation results show that our LMST based routing scheme improves the lifetime of the first node by 100% when compared with non-power aware shortest-path based algorithms, and provides significantly better lifetime for the first node when compared with power-aware algorithms.

The simulations also show that the LMST based approach improves the network partition time drastically. It can also be seen that the improvement of the LMST structure on the lifetime of the system is considerably large when the number of nodes in the same area is increased. We can conclude that LMST approach is more effective in environments where the nodes are densely deployed.

5.3 Summary

In this chapter we presented a new energy efficient distributed routing approach that combines the desired properties of minimum spanning tree and shortest path tree based routing schemes. The proposed scheme uses the advantages of the powerful localized structures such as RNG and LMST and provides simple solutions to the known problems in route setup and maintenance because of its distributed nature. The proposed algorithm is robust, scalable and self organizing. The algorithm is appropriate for systems where all the nodes are not in direct communication range of each other. We show through simulations that our algorithm outperforms shortest weighted path based approaches, and can achieve 90% of the upper bound on lifetime.

The simulation results showed that the SWP over LMST approach is the best among our new family of protocols and by using this approach one can achieve almost the same performance with the best known centralized solution PEDAP.

Another important result is that dynamic methods tend to increase both FNF and NPT timings especially in reasonable densities for sensor networks ($\rho < 15$). This means dynamic methods can balance the energy expenditure among the nodes well while providing good lifetimes for bottleneck nodes.

As a result of the experiments we also conclude that increasing the node density up to some point results in higher system lifetime. However after this point, high density leads into poor network performance. With this result we can see that there should be an optimal density which gives the maximum possible performance.

Although in this work we have used 100 rounds to recompute the aggregation tree as in PEDAP-PA, it is worth to mention that the period of the recomputation is an important factor for achieving long lifetimes. With a small period we can achieve a good balance among the nodes, whereas we have larger overhead due to control packets. Determination of the optimal recomputation period needs complex mathematical analysis and it is beyond the scope of this work. An example of changing recomputation period dynamically in a centralized solution can be found in [30].

The area size and the maximum transmission range are usually set by the application itself. It is an interesting open problem to theoretically derive the optimum number of nodes for given R and l. Also, based on this result, one could combine our method with some sort of sleep scheduling algorithm to get a performance increase on high density networks. So if a sleep scheduling algorithm [23] recomputes the roles of the nodes periodically, the same period can be used to recompute the routing tree spanning only the active nodes with our protocols. Moreover with the advantage of using periodic recomputations, our dynamic methods can be used efficiently in such a scenario. One can also investigate the application of connected dominating sets (CDS) [63] to limit internal tree nodes to such a set, and rotating periodically these sets. Tree computation via broadcasting is possible only via nodes in CDS, and leaves can even sleep temporarily while data are being gathered.

We did not measure the cost of set up and maintenance. However, our motivation is exactly to address this set up cost and maintenance cost by proposing localized solutions. Almost all existing papers do ignore these costs by describing centralized solutions, without even mentioning the communication overhead involved in gathering needed information. In our study, measuring this cost would be even counterproductive. This cost in our approach is negligible compared to the same cost in existing algorithms which are centralized. By ignoring this cost, we were able to conclude that our localized solutions perform almost as well as centralized, and with over 90 percent of ideal number of rounds, when unfair advantage (for not measuring the cost) is given to these alternatives. This clearly shows the effectiveness of our solutions. If we added mentioned cost in simulations, our localized algorithms would be winning, but we would not be able to conclude that we actually cannot really improve our solution further significantly.

Chapter 6

Power Efficient Node Scheduling

As mentioned in Section 3.5, determining a schedule for putting nodes into sleep state, which is called node scheduling, is proven to be a good approach to reduce the total power consumption, thus increase the operational lifetime in wireless sensor networks.

A common scenario where using node scheduling is advantageous is event detection systems. Suppose that there is a field of interest that must be monitored all the time for a certain event such as fire or intrusion. Usually the data to be transmitted in these systems is only a true/false data or the information about the status. Since the data transmitted is usually small, especially when the events to be detected are very rare, the most of the energy consumption is due to sensing and relaying part. So the idea is keeping minimum number of sensor nodes that cover the whole area active while being connected.

In this chapter we will again focus on the same scenario which we work on the previous section, however, in this chapter we add node scheduling to the scene. The simulation results of the previous chapter showed that by increasing the number of active nodes we cannot always achieve an increase in the functional system lifetime. The conclusion was that it is not necessary to keep all the nodes active, and there must be an optimal value for the number of active nodes in a round. In this part of the study we examine this optimal value theoretically.

Main constraint used here is to have full coverage of the area for the data to be collected. Therefore, for each round we have to keep a set of nodes that can cover the whole area as active. For the rest of this chapter, when we consider lifetime, we mean the time elapsed until no full coverage can be achieved with the remaining nodes.

Almost all existing node scheduling algorithms mentioned in Section 3.5 try to minimize energy consumption by minimizing the number of connected active nodes which is required to cover the whole area. In this chapter we define a problem called Maximum Lifetime Node Scheduling (MLNS) and claim that keeping minimum number of nodes active does not result in minimum energy consumption in some certain conditions when the nodes are capable of power control. Although it is very intuitive to minimize the number of active nodes in a round to achieve maximum lifetime, this may cause an increase in the average distance and if nodes are capable of power control, transmitting to far distances requires much more energy. So, if the energy required for transmission is much more than the constants in sensing and relay states, keeping more nodes active and using them as relay nodes can be more appropriate than leaving them in sleep state.

In this chapter, we will investigate the conditions under which keeping more nodes active results in lower energy consumption. And also we will propose a new node scheduling algorithm called Power Efficient Node Scheduling (PENS) which can adapt to the conditions to provide maximum lifetime in any case. We will present our simulation results comparing PENS with a naive algorithm which uses minimum number of nodes approach.

6.1 Maximum Lifetime Node Scheduling Problem

In this section we formally define the Maximum Lifetime Node Scheduling (MLNS) problem by first giving all the definitions necessary to state it formally.

Definition 6.1.1. A subset of nodes $C \subset V$ is called a *cover set* if the target area A is totally covered by the nodes in C.

$$A \subset \bigcup_{i \in C} D_i, \tag{6.1}$$

where D_i is the area covered by node i.

Definition 6.1.2. A subset of nodes $C \subset V$ is called a *connected cover set* if C is a cover set and the induced subgraph of C is connected.

It is obvious that we must find a connected cover set in order to preserve both connectivity and coverage in network. However, there can be many alternative sets of nodes that satisfy this condition. We must choose the one which is optimal in terms of energy consumption and lifetime. Let us define two alternative connected cover sets.

Definition 6.1.3. Among all connected cover sets of a network, *minimum connected cover set* is the one with the smallest number of nodes.

Definition 6.1.4. Among all connected cover sets of a network, *minimum energy connected cover set* is the one with the optimum number of nodes which gives the minimum possible energy consumption in a round.

In order to compute the energy consumption of a cover set, we need to find the energy consumption distribution of the nodes. The actual energy consumption distribution of the nodes is related with the routing strategy chosen. If we choose the right nodes for being active but cannot find a good routing plan, we cannot achieve the desired result. In this work, we define a routing plan as a sequence of aggregation trees which is defined as follows:

Definition 6.1.5. An *aggregation tree* for a set of nodes is a tree rooted at the sink which spans all the nodes in the set.

Definition 6.1.6. An *optimum aggregation tree* for a set of nodes is the minimum energy spanning tree of the nodes in the set.

In this chapter we assume that the optimum aggregation tree is used in order to compute the energy consumption of connected cover set.

Definition 6.1.7. *Maximum Lifetime Node Scheduling Problem* (MLNS) is defined as finding a sequence of aggregation trees - with their cover sets - and their frequencies which provides maximum overall functional lifetime in terms of coverage among all alternatives.

By this definition, in order to solve the MLNS problem firstly we have to enumerate all the possible aggregation trees for all possible connected cover sets and compute their frequencies. After that we should try to find a schedule of these aggregation trees to yield the maximum possible lifetime. Considering that the subproblem of finding the optimum aggregation tree when the set is fixed is NP-Complete [78], we can conclude that MLNS problem is also NP.

On the other hand we can simplify the problem by thinking it as finding a series of disjoint connected cover sets $\widetilde{C} = C_1, C_2, ...C_m$ where the sum of the lifetimes provided by each set (L_{C_i}) is maximum among all alternatives. In this case we have to calculate the lifetime of each possible connected cover set (L_{C_i}). Fortunately, this subproblem is the same as MLDA problem since we have to find the optimum routing tree which uses all the nodes in a specific cover set. Since the MLDA problem is well-known and we have good mathematical models for that case, we can concentrate on finding the disjoint connected cover sets which gives optimum overall lifetime.

6.2 Analysis

As mentioned in Section 6.1, at first glance it seems reasonable to find maximum number of disjoint minimum connected cover sets to achieve a high performance. This idea actually follows the intuition that with minimum number of active nodes per round we can save more energy since keeping a node active is very expensive in terms of energy consumption. However, as mentioned above, if the nodes are capable of power control, energy required for transmission is increasing exponentially with increasing transmission distance. So, by choosing minimum number of nodes to cover whole area, we also increase the distance between them.

Table 6.1: System Parameters for Evaluating Node Scheduling.

r_s	Sensing radius
r_t	Transmission radius
N	Total number of nodes
l	Length of one side of the area (assuming square area)
n_c	Minimum number of nodes for full coverage
n_e	Optimum number of nodes for minimum energy consumption
ω_i	Energy consumption of node i
B	Initial battery energy of the nodes
E_n	Minimum energy that can be achieved with n nodes
T_n	Overall lifetime when n nodes is used in each round

Therefore, it is possible in some conditions that keeping some other relay nodes active can lead into a better lifetime.

From the Section 4.2, we know that we have to minimize the energy consumption in a round in order to yield a better upper bound on lifetime. In this section we will show that the minimum number of nodes approach does not always result in minimum energy consumption by investigating the circumstances which give the minimum possible energy consumption. In order to make it more understandable we will first derive formulations in 1D, and then try to generalize it in 2D. Table 6.1 gives definitions of our system parameters.

6.2.1 1D case

Consider a scenario in which we want to monitor all events on a line of length l. We have N nodes and would like to get maximum system lifetime. As given in Section 6.1 we can think of MLNS problem as finding a set of connected cover sets whose total lifetime is maximum.

For the sake of simplicity in this work we assume that maximum transmission range r_t is big enough to neglect the effect of it. And for the rest of this chapter we introduce an energy model constant c which is equal to $2E_{elec}/E_{amp}$ in order to simplify the formulations given in this chapter.

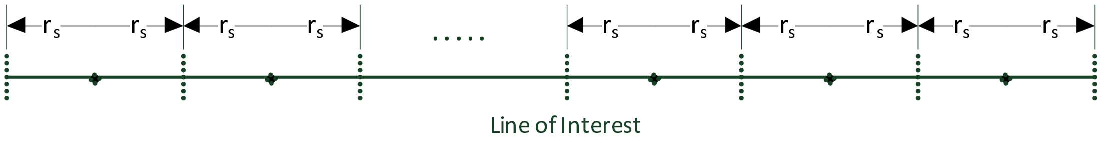


Figure 6.1: Minimum connected cover set in 1D.

We will present the cases for the following two approaches:

- Finding maximum number of disjoint minimum connected cover sets
- Finding maximum number of disjoint minimum energy connected cover sets (our approach)

6.2.1.1 Minimum Connected Cover Set

The following lemma gives the minimum number of nodes required to cover whole line (n_c).

Lemma 6.2.1. *If $2r_s < l$, the minimum number of nodes (n_c) required to cover all the line is $n_c = \lceil l/2r_s \rceil$.*

This result is obvious since each node covers a line segment of length $2r_s$ on the line. In order to minimize the number of nodes covering the line, we have to minimize the overlaps. If we assume that we can place the nodes manually we will have no overlaps by separating nodes by $2r_s$. Since the total length of the line is l, we need at least $\lceil l/2r_s \rceil$ nodes to cover the line. Figure 6.1 illustrates the positioning of nodes to achieve full coverage with minimum number of nodes.

6.2.1.2 Minimum Energy Connected Cover Set

Let us present some proven lemmas for data gathering in 1D.

Lemma 6.2.2. *If $l > (c/(1 - 2^{1-\alpha}))^{1/\alpha}$, then using at least one relay node is more advantageous than the direct transmission.*

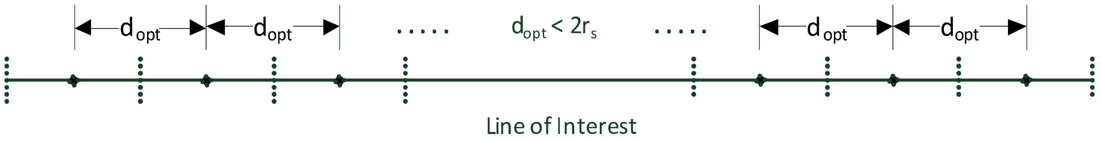


Figure 6.2: Minimum energy connected cover Set in 1D.

This lemma states that if the distance between two nodes is greater than a threshold value, using at least one relay node is more appropriate. This lemma is proved in different papers in literature [8, 62] together with the following two lemmas.

Lemma 6.2.3. *If $l > (c/(1-2^{1-\alpha}))^{1/\alpha}$, then the optimum number of nodes needed to minimize the energy consumption is $n_e = l((\alpha-1)/c)^{1/\alpha}$, where the nodes should be separated with equal distance.*

Lemma 6.2.4. *If $l > (c/(1-2^{1-\alpha}))^{1/\alpha}$, then the minimum possible energy consumption $E_{n_e} = lc((\alpha-1)/c)^{1/\alpha} + l(-1/c)^{(1-\alpha)/\alpha}$.*

The first lemma exactly finds our parameter n_e which is the number of nodes required to minimize the total energy consumption in a round. The second lemma gives our parameter E_{n_e} which is the minimum energy consumption when using n_e nodes. It is proven that this energy value is a lower bound on the energy consumption in a round on a line. So increasing or decreasing the number of nodes will increase the energy expenditure. Figure 6.2 presents the optimum placement of the nodes to achieve minimum energy consumption in one dimension. As it is shown in the figure, it is possible to cover whole line while achieving minimum energy consumption only when the distance between the nodes d_{opt} is smaller than $2r_s$.

6.2.1.3 Comparison

We can state that if $n_c >= n_e$, we have no choice but using minimum number of nodes approach, since we cannot cover whole line with nodes smaller than n_c. However if $n_c < n_e$ we can use minimum energy approach since we know that if $n_c < n_e$ then $E_{n_c} > E_{n_e}$.

The question is that which approach will provide more lifetime. At one side we will have more cover sets which consume more energy. On the other hand we will have less number of cover sets with optimum energy expenditure. In order to get it theoretically let us remember the upper bound on lifetime in terms of first node failure, given in Equation 4.6, which is proven in Section 4.2.

Lemma 6.2.5. *The upper bound for lifetime of the network with N nodes is $\bar{L_N} = NE_0/E_{min}$.*

E_{min} in this formula means the minimum possible energy consumption in a round. In this case we have $E_{n_e} = E_{min}$. As seen the overall lifetime is independent of the number of cover sets. Therefore we can state that if $E_{n_1} < E_{n_2}$ then potentially the lifetime provided by using n_1 nodes is greater than its alternative ($T_{n_1} > T_{n_2}$). With this results we can easily conclude that the upper bound of overall lifetime of the minimum energy case T_{n_e} is greater than the other alternative T_{n_c}.

If we simply put the results found in above lemmas, we can find the conditions when using minimum energy approach is better than the other:

$$
\begin{aligned}
n_c &< n_e && (6.2)\\
l/2r_s &< l((\alpha-1)/c)^{1/\alpha} && (6.3)\\
r_s &> 0.5(c/(\alpha-1))^{1/\alpha} && (6.4)
\end{aligned}
$$

Corollary 6.2.6. *If $r_s > 0.5(c/(\alpha-1))^{1/\alpha}$ and $l > (c/(1-2^{1-\alpha}))^{1/\alpha} \Rightarrow T_{n_e} > T_{n_c}$. For $\alpha = 2$, if $r_s > 0.5\sqrt{c}$ and $l > \sqrt{2c} \Rightarrow T_{n_e} > T_{n_c}$.*

6.2.2 2D case

In two dimensions we can also investigate the results of using two approaches. However in 2D, we cannot find exact formulations. Because of this fact we will analyze the problem asymptotically.

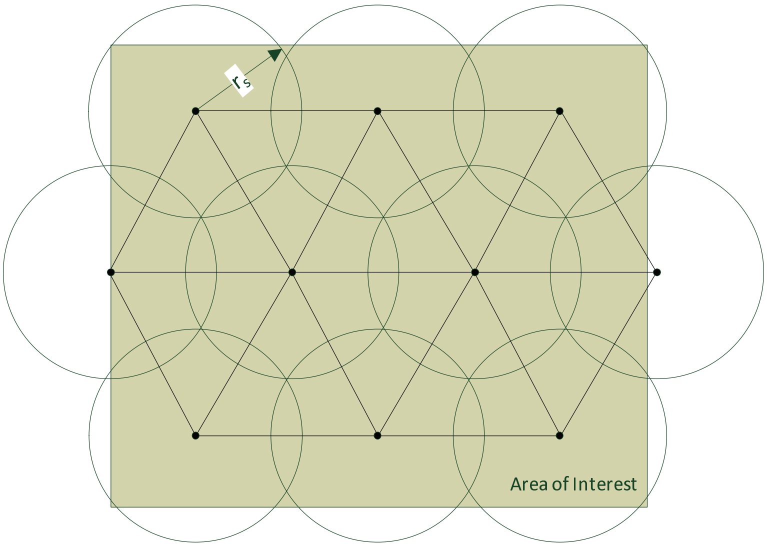


Figure 6.3: Minimum connected cover set in 2D.

6.2.2.1 Minimum Connected Cover Set

First of all let us try to get the minimum number of nodes to cover whole area in 2D. Fortunately it is proven that if $r_t > \sqrt{3}r_s$ placing the nodes on the vertices of a triangular lattice is optimal in terms of number of nodes required to cover the whole area [76]. Figure 6.3 illustrates the positioning of the nodes to achieve full coverage with minimum possible nodes. Since we assumed no maximum transmission range in the work, we can use this lower bound safely. According to this result the minimum number of nodes required for full coverage is

$$n_c = \frac{2\sqrt{3}l^2}{9r_s^2} \tag{6.5}$$

6.2.2.2 Minimum Energy Connected Cover Set

On the other hand, computing the aggregation tree with minimum energy consumption is another problem to solve. It is obvious that this tree will be minimum spanning tree where link costs are the transmission costs. This means if the nodes are given we can find the desired tree in polynomial time. However, the question is how many nodes should we select and how should we select them in order to yield the minimum possible energy tree.

In Section 4.3 we have analyzed the optimum number of nodes for minimum energy expenditure and yielded a formula for this:

$$n_e = \left(\frac{(\frac{\alpha}{2} - 1)\kappa}{c} \right)^{2/\alpha} \tag{6.6}$$

6.2.2.3 Comparison

As in previous section, we can conclude with the following corollary since $n_c < n_e \Rightarrow E_{n_c} > E_{n_e} \Rightarrow T_{n_c} < T_{n_e}$.

Corollary 6.2.7. *$T_{n_e} > T_{n_c}$ if the following condition holds*

$$r_s > \sqrt{\frac{2\sqrt{3}l^2}{9n_e}} \tag{6.7}$$

6.3 Proposed Algorithm

In this section we present our new algorithm called Power Efficient Node Scheduling (PENS) for data aggregation. The main idea behind our algorithm is to use minimum possible energy in a round in order to achieve better functional system lifetime. Motivated from the analysis done in Section 6.2 we concentrate on finding minimum energy connected cover sets instead of finding sets with minimum number of nodes. Since the routing strategy is also as important, as the node selection strategy we will use PEDAP [64] as the routing algorithm which is known as one of the best routing algorithms to solve MLDA problem. So, with a combination of a good node selection strategy and a good routing strategy we achieve longer lifetimes.

Before going into detail of our algorithm, let us give some definitions and theorems.

6.3.1 Power Efficient Neighbor Set

Definition 6.3.1. An edge e_{ij} is defined as *power efficient* if there is no other relay node $k \in N$ where power required to transmit a packet from i to j over k is smaller than the power required for direct transmission from i to j ($E_{ik} + E_{kj} < E_{ij}$).

Definition 6.3.2. Power efficient neighbor set of a node i, *PEN(i)*, is defined as

$$\text{PEN}(i) = \{j \mid e_{ij} \text{ is power efficient}\} \tag{6.8}$$

For a node i transmitting to other nodes outside PEN(i) is not power efficient, since there is at least one node in PEN(i) where using it as a relay reduces the total transmission cost. So usage of only power efficient edges should result in significant energy saving.

6.3.2 Coverage Calculation

Determining whether a selected node set covers the whole area or not is another problem that we have to deal with in node scheduling problem. The naive approach is to check all points in the target area A whether they are covered by at least one node or not. But this approach is not efficient in terms of computation. Xing et al. give an efficient way of determining full coverage in [79] with the Theorem 6.3.3.

Theorem 6.3.3. *A convex region A is covered by a set of sensors S if*

- *there exist in region A intersection points between sensors or between sensors and A's boundary*
- *all intersection points between sensors and between sensors and A's boundary are covered by S*

In this theorem the intersection points are the ones between all sensors' sensing regions (D_i) and area boundary A. It states that if all intersection points are covered by a given set of nodes, then the given set covers the whole area. Please note that a node cannot cover its own intersection points.

Let I be the set of all intersection points of sensing regions of the nodes and boundary of A. Let us denote the subset of intersection points covered by node i as I_i. With the above theorem the problem of selecting nodes become set cover problem, in which we have to find a cover C for each round where

$$I = \bigcup_{i \in C} I_i \tag{6.9}$$

Since in this work we assume all nodes are in transmission range of each other, we don't have to deal with connectivity, which means every cover set C will also be a connected cover set.

6.3.3 Finding Minimum Connected Cover Set

Computing the set cover, which is defined as finding the smallest number of sets whose union contains all elements, is known as NP-Complete and it is one of Karp's 21 NP-complete problems [35]. A greedy algorithm is given in [17] which states to choose the set that contains the largest number of uncovered elements at each iteration of the algorithm. It is shown that the approximation ratio achieved by the greedy algorithm is $H(s)$, where $H(n)$ is the n-th harmonic number and s is the size of the largest set.

$$\begin{aligned} H(n) &= \sum_{k=1}^{n} \frac{1}{k} && (6.10) \\ &\leq \ln n + 1 && (6.11) \end{aligned}$$

The studies on inapproximability of the set cover problem showed that the problem cannot be approximated in polynomial time to within a factor of $c \ln n$ under the weaker assumption that P $\neq$ NP, where c is a constant less than 1 [6, 21, 40, 50]. The results of these studies showed that the greedy algorithm given in [17] is the best possible polynomial approximation algorithm [75].

Algorithm 3 Node selection strategy to find the minimum connected cover set.

```
$S \leftarrow \emptyset$
$UI_i \leftarrow I_i$
while S does not cover whole area do
  $t \leftarrow \arg\max_{i \in \{N \setminus S\}}(|UI_i|)$
  $S \leftarrow S \cup t$
  for all $i \in \{N \setminus S\}$ do
    $UI_i \leftarrow UI_i \setminus I_t$
  end for
end while
```

Since each node defines a set of intersection points I_i, finding the minimum number of nodes that cover whole area, is the same as finding the minimum number of sets that cover all the intersection points on the area.

Algorithm 3 presents the approximation algorithm for finding the set cover with minimum number of nodes used for comparison purposes in our simulations. This algorithm is based on the greedy algorithm given in [17].

This algorithm just tries to select the node with maximum number of uncovered intersection points in each iteration until all the area is covered by the selected nodes. Initially selected node set is empty and set of uncovered intersection points that node i covers (UI_i) is equal to I_i for all nodes $i \in N$. Every time a node is selected the intersection points covered by that node are removed from the uncovered point sets of the other nodes. The idea behind using this algorithm is to find maximum number of disjoint cover sets as it is proposed in [12, 58].

6.3.4 PENS Algorithm

Instead of using the minimum number of nodes required to cover whole area, PENS algorithm aims to utilize the optimum number of nodes required for minimum possible energy consumption in a data gathering round. To achieve this PENS algorithm uses a different node selection strategy.

Algorithm 4 Node selection strategy in PENS.

```
$S \leftarrow \emptyset$
$UI_i \leftarrow I_i$
calculate PEN($i$) for all nodes
$CN \leftarrow$ PEN($s$)  // sink
while S does not cover whole area do
  $t \leftarrow \arg\max_{i \in CN}(|UI_i|)$
  $S \leftarrow S \cup t$
  $CN \leftarrow \{CN \cup (\text{PEN}(t) \setminus S)\} \setminus t$
  for all $i \in \{N \setminus S\}$ do
    $UI_i \leftarrow UI_i \setminus I_t$
  end for
end while
```

The selection strategy of PENS, which is the heart of it, is given in Algorithm 4. As seen the algorithm resembles the previous one. Only difference actually is the introduction of candidate node set (CN). CN holds the set of candidate nodes to be selected in the next iteration. Initially it contains only PEN of the sink s. In each iteration the node in CN with maximum number of uncovered intersection points in CN is selected. After that the nodes in PEN of the newly selected node, which are not selected yet, are added to the CN, whereas the node itself is removed from CN. In this way the algorithm ensures usage of only power efficient edges and thus provides energy saving. Note that it is obvious that the number of nodes selected in this way may be more than the minimum number of nodes approach under conditions mentioned in Section 6.2. This algorithm is again an approximation algorithm which tries to determine the optimum number of nodes to be kept active in a round.

After the nodes are selected, they are kept active while the other nodes are put into sleep state. The active nodes can use PEDAP to route the packets to sink node until no coverage can be provided by active nodes. After that, the idea is to select new cover sets iteratively using the same selection algorithm using the remaining nodes until no coverage can be achieved. Figure 6.4 illustrates the working principle of PENS. It divides the network into m cover sets and uses one cover set at a time. For routing packets in a cover set it uses PEDAP to achieve longer lifetime.

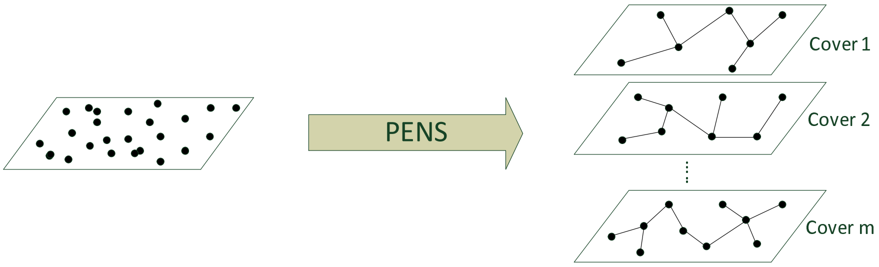


Figure 6.4: PENS Protocol.

6.4 Simulation Results

In order to measure the performance of our algorithm, we run experiments to see its effects on number of active nodes in a round, energy consumption in a round and finally functional system lifetime. First of all we conduct experiments in 1-dimension to examine the theory given in Section 6.2. After that we move on the 2-dimensions and compared the performance of our algorithm (PENS) with the minimum number of nodes approach in next section.

6.4.1 1D Simulation Results

For 1D case we have run simple simulations. We assumed a line of length $l = 1000m$ to be covered all the time. We also assumed that all nodes can communicate with each other, which means there is no maximum transmission radius. We assumed $\alpha = 2$ and $c = 500$ for experiments in 1D as in [26]. We used 1000 nodes to cover that area. We repeated experiments for varying r_s values ranging from 4 to 200 meters. In each case we evaluated minimum size connected cover set and PENS approaches. For each case, we measure the number of nodes required, the average power consumption in one round, and total system lifetime.

Figure 6.5 shows the number of nodes required for each method with varying r_s. As seen on the figure after $r_s > \sqrt{c/4} \cong 12$, keeping the number of nodes constant results in minimum energy consumption. The small decrease in graph is as a result of the boundary conditions. In order to cover whole area we choose nodes from the centered line of length $l - 2r_s$. So with increasing r_s value, the

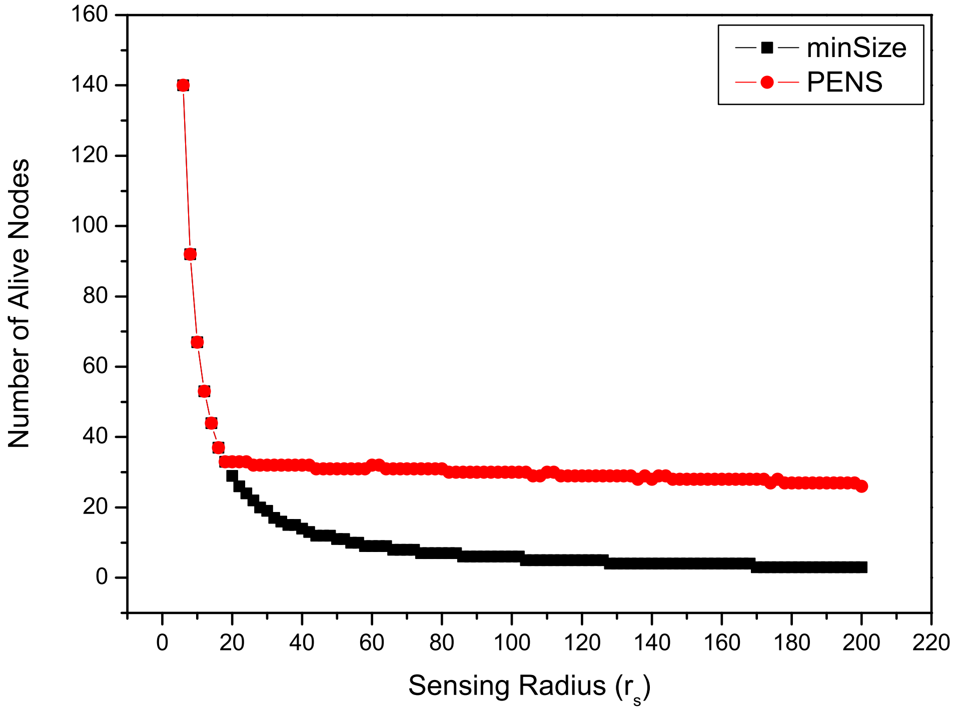


Figure 6.5: Effect of sensing radius on number of active nodes for (1D)

number of nodes required will slightly decrease.

In Figure 6.6 the power consumptions for one round is presented. We expect an increase in power consumption after $r_s > 12$. And actually we can see that increase after that point in our experiments. Although we expect no increase for the PENS approach, we see a small increase which is mainly because of the positions of the nodes. If we look at the power consumption of minimum number of nodes approach, we can see the huge difference. With increasing r_s values the difference in power consumptions of two methods reaches almost 10 fold.

However it is not enough to examine only one round. If the number of nodes for one round decreases, the number of disjoint connected cover sets increases. It may compensate the increase in power consumption. But this is not what we have observed in our experiments. As we examined theoretically, the number of nodes in a round has almost no effect on the overall lifetime. The overall lifetime is directly related with total initial energy and energy consumption in a round.

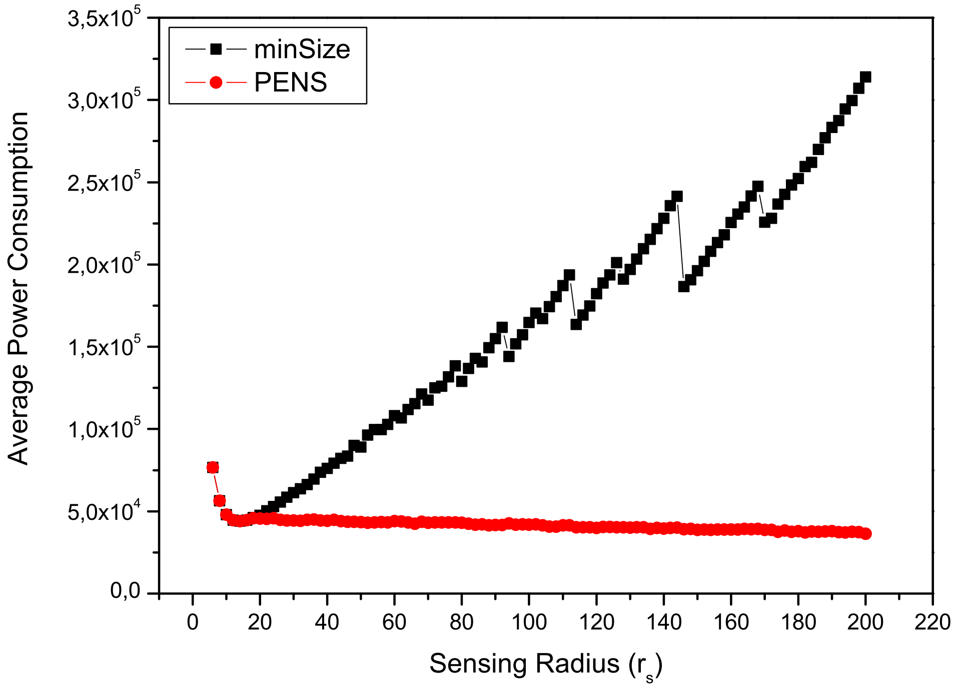


Figure 6.6: Effect of sensing radius on average power consumption (1D).

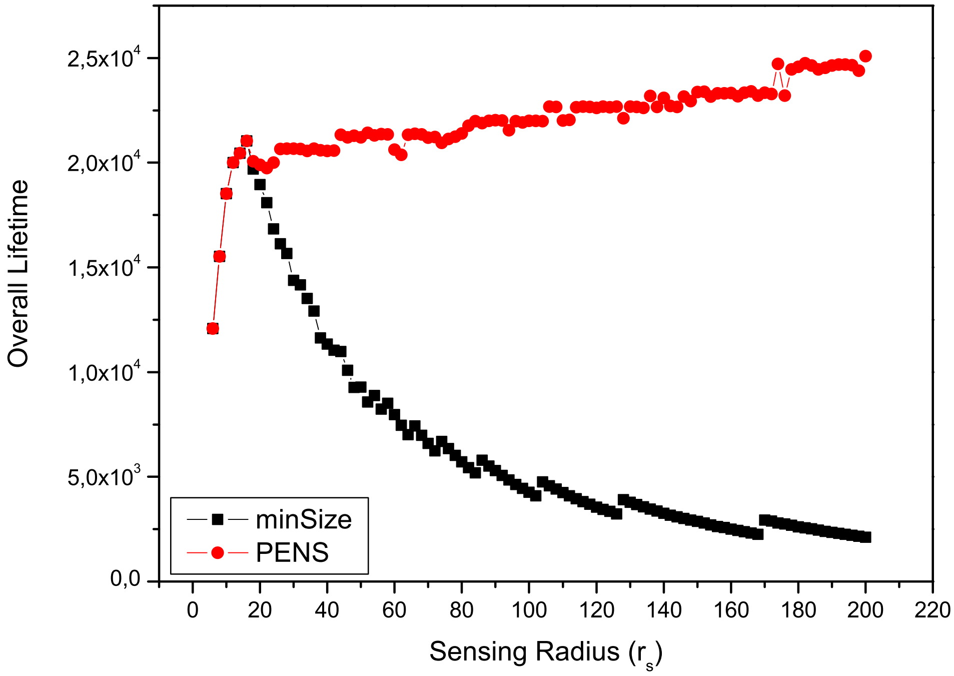


Figure 6.7: Effect of sensing radius on overall lifetime (1D).

Figure 6.7 shows the overall system lifetime with increasing r_s. As seen on the figure the overall lifetime is almost constant for minimum energy approach (our PENS algorithm). This is meaningful since the energy consumption in one round is almost the same. On the other hand the overall lifetime in minimum connected cover set approach rapidly decreases after the threshold. All these experiments verify our theoretical results.

6.4.2 2D Simulation Results

For 2D case we assumed a square area with side length of $l = 1000m$ to be covered. We again used 1000 nodes. We assumed $\alpha = 4$ and $c = 2 \times 10^8$ for experiments in 2D as in [62]. The reason we do not assume $\alpha = 2$ in 2D is that we expect no benefit to use more active nodes as we show in Section 4.3.3.

Before running the simulations we tried to find κ_p^α for $\alpha = 4$ and $p = 2$. In order to find κ_2^4 we compute the cost of the minimum spanning tree (M_n^α) in a square area with side length of 1000m for different number of nodes (n). For each n we repeated the computation 1000 times to achieve a good approximation. As a result of this computation κ_2^4 become approximately 5×10^{11}. According to Equation 6.6, n_e for our case is ≈ 50. With these values the threshold for r_s should be ≈ 87.5 according to Equation 6.7.

In order to see the performance of PENS algorithm we repeated experiments for varying r_s values ranging from 60 to 200 meters since smaller sensing ranges provide no coverage. In each simulation, we compared minimum size connected cover set and PENS approaches. For each case again we try to find the number of nodes required, the average power consumption in one round, and total system lifetime.

Figure 6.8 shows number of nodes required for each method. As expected, in two dimensional case we see a similar pattern with 1D case. So after $r_s > 87.5$ the number of active nodes for PENS approach is more compared with the minimum size approach. Although we expect the number of nodes become constant

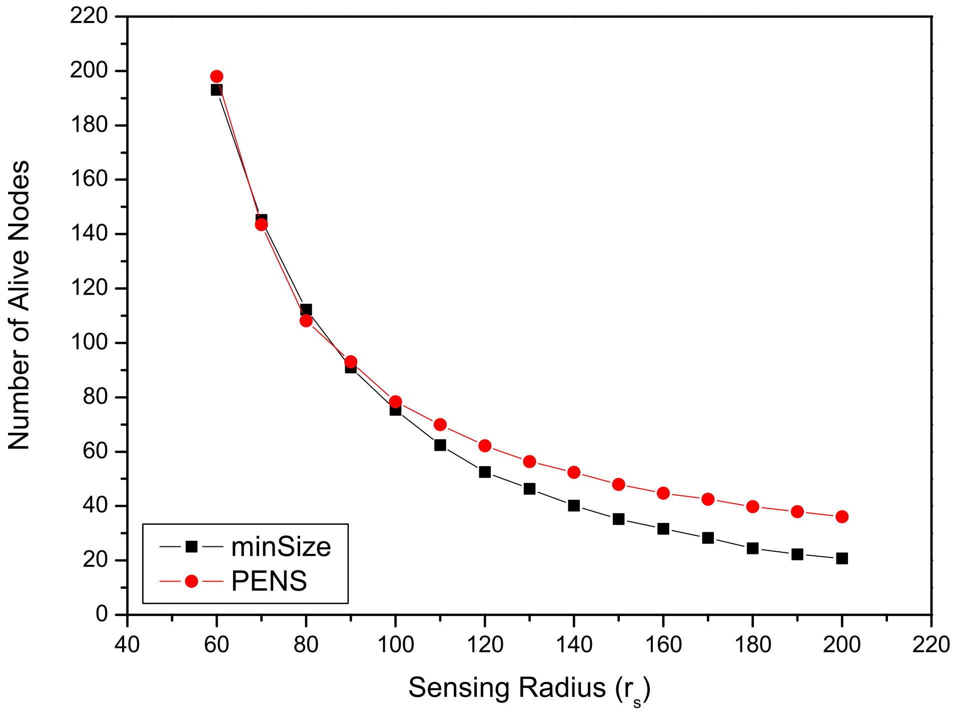


Figure 6.8: Effect of sensing radius on number of active nodes (2D).

for PENS approach, due to boundary and coverage conditions it continues to decrease. It is interesting to see that the difference between number of nodes required to cover whole area for both methods is not so significant. So in PENS approach we tend to use ≈ 15 more nodes compared to minimum size approach.

Figure 6.9 presents the average power consumptions per round for each method. For minimum size approach, as we found theoretically, the average power consumption is increased dramatically after the threshold value. By using smaller number of nodes, we yield greater distances among the nodes; and thus we get enormous amount of energy consumption for each round. The difference of two methods become almost 4 fold when $r_s = 200$.

In theory we can find N/n distinct cover sets for each approach. However in practice the location of the nodes and the coverage condition puts an upper bound to the maximum number of disjoint cover sets. Fortunately, with the intersection points method we use in our algorithms we can find the upper bound on the

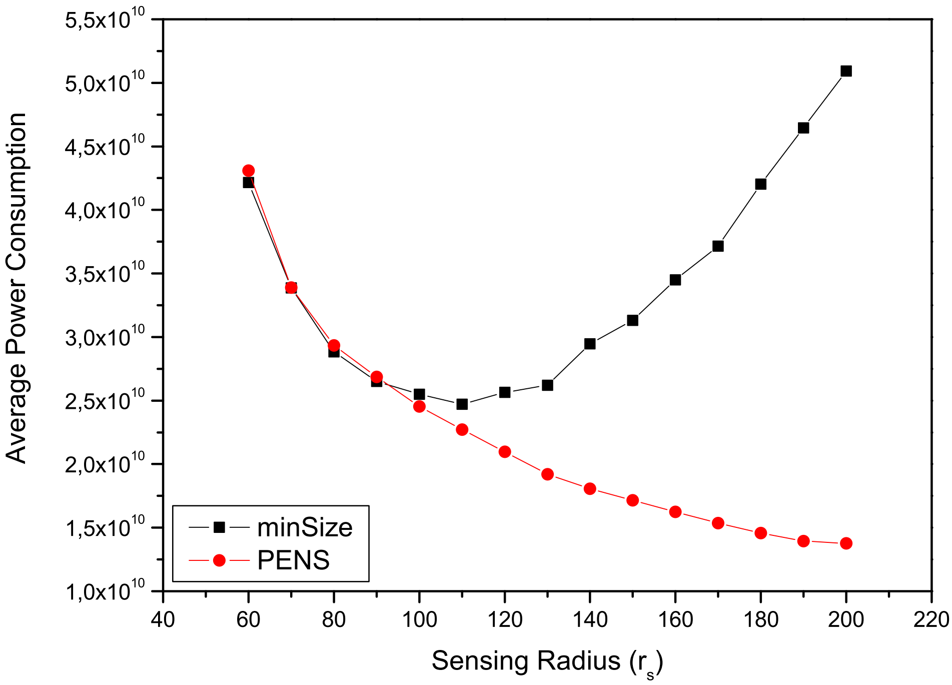


Figure 6.9: Effect of sensing radius on average power consumption (2D).

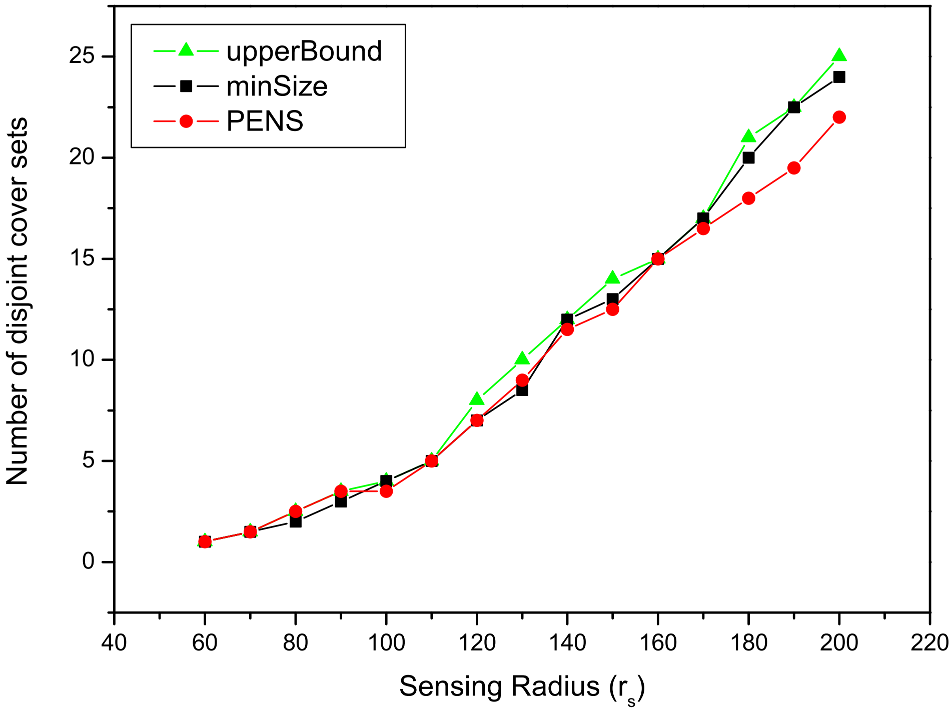


Figure 6.10: Effect of sensing radius on number of disjoint cover sets (2D).

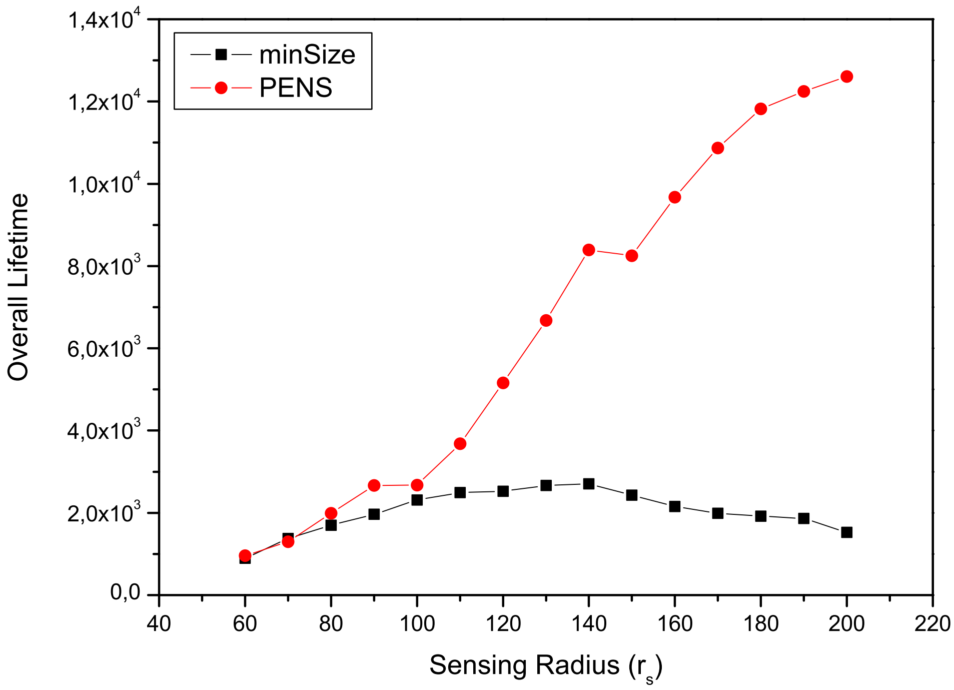


Figure 6.11: Effect of sensing radius on overall lifetime (2D).

number of disjoint cover sets experimentally. So let IC_k denote the number of sensor nodes which cover an intersection point $k \in I$. The upper bound can be found by $\min_{k \subset I}(IC_k)$. Figure 6.10 shows the experiment results on number of disjoint cover sets found by each method compared with this upper bound. As seen in the figure both methods are very close to each other and the upper bound.

Finally, Figure 6.11 shows the overall lifetimes achieved by two methods. It is obvious that the PENS approach outperforms the minimum size approach in overall lifetime. This is a direct result of the low energy consumption of PENS approach. Since the number of disjoint cover sets is almost the same, the difference is because of the average power consumption in a round.

6.5 Summary

In this chapter we first defined a problem Maximum Lifetime Node Scheduling (MLNS) which addresses finding a schedule of connected cover sets which provides maximum functional system lifetime. Although almost every study in the literature on node scheduling try to minimize the number of nodes in a connected cover set, we have realized the fact that in some conditions keeping more nodes active can provide a better energy consumption if the nodes are capable of power control. We presented a theoretical analysis about these conditions where using more nodes results in improved lifetime, compared to using minimum number of nodes to achieve full coverage. It turned out that if the sensing range of the nodes is greater than an application specific threshold, it is better to keep more nodes active and use them as relay nodes in terms of functional system. We have also proposed a simple protocol, which we call Power Efficient Node Scheduling protocol (PENS), to illustrate this idea and showed that the approach outperforms its minimum connected cover set based alternative. Proposed protocol is an adaptive protocol which tries to give the optimum solution adjusted according to the application setting. It achieves success by choosing the right number of nodes to keep active as well as using a well-studied powerful routing solution. The simulation results showed that PENS protocol improves the functional system lifetime by 10× compared to its alternative.

Our proposed protocol is just a demonstration about how well a node scheduling algorithm can improve the lifetime of a data gathering application for sensor networks that consist of power control capable nodes. The proposed algorithm can be improved in several ways. PENS algorithm cannot guarantee the minimum energy consumption, rather it is just an improvement over protocols using minimum number of nodes approach. Better algorithms can be proposed to provide improved energy consumption. The protocol also does not consider the remaining energy levels of the nodes in both selecting active nodes and in its routing solution. It also assumes that the connected cover sets must be disjoint. However, if a node is allowed to be the part of multiple cover sets, the nodes with low weights in one cover set can be reused in other sets to achieve a better lifetime.

Chapter 7

Conclusion

In this thesis, we worked on improving functional system lifetime of data gathering applications developed for wireless sensor networks. We can summarize the contributions of this work in four parts:

- a survey of approaches used in literature
- a theoretical model for lifetime analysis
- a localized, power-aware routing solution
- an energy efficient sleep scheduling solution

The survey part of this thesis gives a clear understanding of the problem and presents existing works with a categorization with respect to the approaches they use to solve the lifetime problem. The five classes of the solution approaches are: data volume minimization, efficient topology construction, routing, sleep scheduling and mobility. As a result of this survey we believed to have a good theoretical model to examine the lifetime problem.

In the second part of the thesis, an analysis on lifetime for data gathering applications is presented. There are two main constraints in this analysis: perfect aggregation and power control. After presenting the formulation of functional lifetime, we derived an upper bound for the data gathering scenario, which has a simple and intuitive form: the total initial energy over energy consumed in a round. We also gave a mathematical analysis for minimum energy expenditure, where the results of this analysis are used in developing a sleep scheduling solution.

After this analysis, we proposed a new distributed routing protocol, called L-PEDAP, which uses the advantage of power control and perfect aggregation. As a result of the analysis given in the second part, main focus of the proposed routing protocol was to minimize the power consumption in a round, while balancing the load among the nodes. The proposed protocol combines the desired properties of minimum spanning tree and shortest path tree based routing schemes, by using powerful localized structures such as RNG and LMST. The results of our comprehensive simulations showed that L-PEDAP protocol outperformed previous proposed methods in the literature. However, as a result of these simulations, we realized that increasing the number of nodes in the system does not help in improving the functional system lifetime regardless of used routing scheme.

In the final part, in order to solve the unscalability problem of routing solutions, we defined a new problem which we call Maximum Lifetime Node Scheduling (MLNS). As a solution to MLNS problem, we propose a new sleep scheduling algorithm, called PENS. Different from the previous sleep scheduling algorithms, the proposed algorithm tries to keep optimum number of nodes alive, instead of keeping minimum number of nodes alive. In this part of the thesis, we derived mathematical formulations of conditions where keeping more nodes alive is more efficient, and confirmed these formulations by running several simulations. The simulation results were very promising.

Although we have already integrated many of the lifetime improvement methods in our solution framework (efficient topologies, routing, node scheduling, data aggregation), in order to extend this work, other lifetime improvement methods can be incorporated into our solution, such as prediction based communication and mobility.

In our scenario, the users only require an aggregated value, and therefore they can tolerate a small error in the final value. In such cases, the nodes can predict what to be sent by using a statistical model considering the previously transmitted data. The source node can decide to send the new data depending on whether the next sensed value is appropriate with the statistical model or not. If the next value can be predicted with a small error, the source does not send the actual data, and the destination node uses the next value from the model, as if it is sent by the source. This communication model can extend the lifetime of the applications, where the data is correlated in time dimension, by an order of magnitude, since the data transmission is the most expensive operation in sensor networks in terms of energy consumption.

If possible for the specific application, utilization of mobile base stations can also improve the lifetime of a wireless sensor network dramatically. The use of mobile base stations reduces the load of a node in two ways. Firstly, it decreases the distance to communicate by getting closer to the senders of data. Secondly, it decreases the degree of a node by lowering the need for multi-hop communication. Since the load of a node is determined with these two parameters (degree and distance), the energy gain that a mobile base station can provide is highly desirable.

As a future work, we will try to incorporate these techniques into our solution framework by proposing a new mobile base station scheduling algorithm to determine the actual route of the base station, and a new statistical model for predicting the next data to be sent. Moreover, we will also improve the performance of our node scheduling solution to the extent possible.

Bibliography

[1] http://robotics.eecs.berkeley.edu/∼pister/SmartDust/.

[2] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Elsevier Ad Hoc Network Journal*, 3(3):325 – 249, 2005.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.

[4] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6–28, 2004.

[5] D. Aldous and J. M. Steele. Asymptotics for euclidean minimal spanning trees on random points. *Probability Theory and Related Fields*, 92:247–258, 1992. 10.1007/BF01194923.

[6] N. Alon, D. Moshkovitz, and S. Safra. Algorithmic construction of sets for k-restrictions. *ACM Trans. Algorithms*, 2:153–177, April 2006.

[7] J. Bachrach and C. Taylor. Localization in sensor networks. In *Stojmenovic I, editor. Handbook of Sensor Networks: Algorithms and Architectures*, pages 277–310. Wiley, 2005.

[8] M. Bhardwaj, A. Chandrakasan, and T. Garnett. Upper bounds on the lifetime of sensor networks. In *IEEE International Conference on Communications*, pages 785–790, 2001.

[9] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: a media access protocol for wireless lan's. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 212–225, New York, NY, USA, 1994. ACM Press.

[10] A. Bogdanov, E. N. Maneva, and S. Riesenfeld. Power-aware base station positioning for sensor networks. In *Proceedings of INFOCOM*, 2004.

[11] E. Bulut and I. Korpeoglu. Sleep scheduling with expected common coverage in wireless sensor networks. *ACM Wireless Networks Journal*, 17:19–40, January 2011.

[12] M. Cardei and D.-Z. Du. Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, 11(3):333–340, 2005.

[13] J. Carle and D. Simplot-Ryl. Energy-efficient area monitoring for sensor networks. *Computer*, 37(2):40–46, 2004.

[14] J.-H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *IEEE INFOCOM 2000*, pages 22–31, March 2000.

[15] J.-H. Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12:609–619, August 2004.

[16] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, pages 85–96, Rome, Italy, July 2001.

[17] V. Chvatal. A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

[18] I. C. Committee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* IEEE Strandard 802.11-1997, New York, NY, 1997.

[19] F. Emekci, H. Yu, D. Agrawal, and A. E. Abbadi. Energy-conscious data aggregation over large-scale sensor networks. Technical Report 2003-32, University of California Santa Barbara, November 2003.

[20] O. Escalante, T. Perez, J. Solano, and I. Stojmenovic. RNG-based searching and broadcasting algorithms over internet graphs and peer-to-peer computing systems. In *AICCSA '05: Proceedings of the ACS/IEEE 2005 International Conference on Computer Systems and Applications*, pages 17–I. IEEE Computer Society, 2005.

[21] U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45:634–652, July 1998.

[22] R. G. Gallager, P. A. Humblet, and P. M. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Languages and Systems*, 5(1):66–77, January 1983.

[23] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic. Localized sensor area coverage with low communication overhead. In *Pervasive Computing and Communications, 2006. PerCom 2006. Fourth Annual IEEE International Conference on*, pages 10 pp.–337, March 2006.

[24] Y. Gu, D. Bozdag, E. Ekici, F. Özgüner, and C.-G. Lee. Partitioning based mobile element scheduling in wireless sensor networks. In *Sensor and Ad Hoc Communications and Networks, IEEE SECON 2005*, pages 386–395.

[25] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 81–95, New York, NY, USA, 2003. ACM.

[26] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *33rd Annual Hawaii International Conference on System Sciences,*, pages 3005–3014, 2000.

[27] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[28] L. Hu and D. Evans. Localization for mobile sensor networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 45–57, New York, NY, USA, 2004. ACM.

[29] C. Hua and T.-S. P. Yum. Optimal routing and data aggregation for maximizing lifetime of wireless sensor networks. *IEEE/ACM Trans. Netw.*, 16(4):892–903, 2008.

[30] S. Hussain and O. Islam. An energy efficient spanning tree based multi-hop routing in wireless sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 4383–4388, March 2007.

[31] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1):2–16, 2003.

[32] D. Johnson, D. Maltz, and J. Broch. *DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*, chapter 5, pages 139–172. Addison-Wesley, 2001.

[33] K. Kalpakis, K. Dasgupta, and P. Namjoshi. Maximum lifetime data gathering and aggregation in wireless sensor networks. In *Proceedings of the 2002 IEEE International Conference on Networking (ICN'02)*, pages 685–696, August 2002.

[34] P. Karn. MACA - a new channel access method for packet radio. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, page 134?140. ARRL, 1990.

[35] R. M. Karp. Reducibility Among Combinatorial Problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[36] M. Khan, G. Pandurangan, and B. Bhargava. Energy-efficient routing schemes for sensor networks. Technical Report CSD TR 03-013, Department of Computer Science, Purdue University, July 2003.

[37] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer. A cone-based distributed topology-control algorithm for wireless multi-hop networks. *IEEE/ACM Transaction on Networking (TON)*, 13(1):147–159, 2005.

[38] N. Li, J. C. Hou, and L. Sha. Design and analysis of an MST-based topology control algorithm. In *INFOCOM*, 2003.

[39] S. Lindsey and C. S. Raghavendra. PEGASIS: Power-efficient gathering in sensor information systems. In *IEEE Aerospace Conference*, March 2002.

[40] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41:960–981, September 1994.

[41] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol. In *European Wireless Conference*, 2002.

[42] F. J. Ovalle-Martinez, I. Stojmenovic, F. Garcia-Nocetti, and J. Solano-Gonzalez. Finding minimum transmission radii for preserving connectivity and constructing minimal spanning trees in ad hoc and sensor networks. *Journal of Parallel and Distributed Computing*, 65(2):132–141, 2005.

[43] J. Pan, L. Cai, Y. T. Hou, Y. Shi, and S. X. Shen. Optimal base-station locations in two-tiered wireless sensor networks. *IEEE Transactions on Mobile Computing*, 4(5):458–473, 2005.

[44] S.-J. Park and R. Sivakumar. Energy efficient correlated data aggregation for wireless sensor networks. *Int. J. Distrib. Sen. Netw.*, 4(1):13–27, 2008.

[45] M. D. Penrose. The longest edge of the random minimal spanning tree. *The Annals of Applied Probability*, 7(2):340–361, 1997.

[46] T. Perez, J. Solano-Gonzalez, and I. Stojmenovic. LMST-based searching and broadcasting algorithms over internet graphs and peer-to-peer computing systems. In *IEEE International Conference on Signal Processing and Communications (ICSPC 2007)*, pages 1227–1230, Dubai, United Arab Emirates, November 2007.

[47] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, page 90, 1999.

[48] R. Prim. Shortest Connecting Networks and Some Generalizations. *Bell Syst. Tech. J.*, 36:1389–1401, 1957.

[49] R. Ramanathan and R. Hain. Topology control of multihop wireless networks using transmit power adjustment. In *INFOCOM 2000*, pages 404–413, Tel Aviv, Israel, March 2000.

[50] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, STOC '97, pages 475–484, New York, NY, USA, 1997. ACM.

[51] P. Rentala, R. Musunuri, S. Gandham, and U. Saxena. Survey on sensor networks. Technical Report UTDCS-10-03, University of Texas at Dallas, 2003.

[52] V. Rodoplu and T. Meng. Minimum energy mobile wireless networks. *IEEE J. Select. Areas Commun.*, 17(8):1333 – 1344, Aug. 1999.

[53] A. Salhieh, J. Weinmann, M. Kochhal, and L. Schwiebert. Power efficient topologies for wireless sensor networks. In *Proceedings of the International Conference on Parallel Processing*, pages 156–163, September 2001.

[54] C. Schurgers, V. Tsiatsis, and M. B. Srivastava. STEM: Topology management for energy efficient sensor networks. In *Aerospace Conference 2002*, March 2002.

[55] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: modeling a three-tier architecture for sparse sensor networks. In *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 30–41, 2003.

[56] S. Singh, M. Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Mobile Computing and Networking*, pages 181–190, 1998.

[57] A. Sinha and A. Chandrakasan. Dynamic power management in wireless sensor networks. *Trans. Design and Test of Computers*, 18(2):62–74, Mar./Apr. 2001.

[58] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *ICC 2001*, Helsinki, Finland, June 2001.

[59] I. Solis and K. Obraczka. In-network aggregation trade-offs for data collection in wireless sensor networks. *International Journal on Sensor Networks (IJSNet)*, 1(2), 2006.

[60] A. A. Somasundara, A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava. Controllably mobile infrastructure for low energy embedded networks. *IEEE Transactions on Mobile Computing*, 05(8):958–973, 2006.

[61] J. M. Steele. Growth rates of euclidean minimal spanning trees with power weighted edges. *Annals of Probability*, 1988.

[62] I. Stojmenovic and X. Lin. Power-aware localized routing in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(11):1122–1133, 2001.

[63] I. Stojmenovic and M. Seddigh. Broadcasting algorithms in wireless networks. In *Proc. Int. Conf. on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet SSGRR*, L'Aquila, Italy, 2000.

[64] H. Ö. Tan and I. Korpeoglu. Power efficient data gathering and aggregation in wireless sensor networks. *SIGMOD Record*, 32(4):66–71, 2003.

[65] H. O. Tan, I. Korpeoglu, and I. Stojmenovic. A distributed and dynamic data gathering protocol for sensor networks. In *AINA '07: Proceedings of the 21st International Conference on Advanced Networking and Applications*, pages 220–227, Niagara Falls, CA, 2007. IEEE Computer Society.

[66] H. O. Tan, I. Korpeoglu, and I. Stojmenovic. Computing localized power-efficient data aggregation trees for sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(3):489–500, 2011.

[67] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *WSNA '02: Proceedings of the 1st ACM*

international workshop on Wireless sensor networks and applications, pages 32–41, New York, NY, USA, 2002. ACM Press.

[68] C.-K. Toh. Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks. *IEEE Communications Magazine*, 39(6):138–147, June 2001.

[69] G. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*, 12:231–268, 1980.

[70] S. Upadhyayula and S. K. S. Gupta. Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (dac) in wireless sensor networks. *Ad Hoc Netw.*, 5(5):626–648, 2007.

[71] W. Wang, V. Srinivasan, and K.-C. Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. In *MobiCom '05: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, pages 270–283, New York, NY, USA, 2005. ACM Press.

[72] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 28–39, New York, NY, USA, 2003. ACM.

[73] X. Wang, Q. Zhang, W. Sun, W. Wang, and B. Shi. A coverage-based maximum lifetime data gathering algorithm in sensor networks. In *MDM '06: Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)*, page 33, Washington, DC, USA, 2006. IEEE Computer Society.

[74] Y.-H. Wang, H.-J. Mao, C.-H. Tsai, and C.-C. Chuang. HMRP: Hierarchy-based multipath routing protocol for wireless sensor networks. In *Embedded and Ubiquitous Computing (EUC) Workshops*, volume 3823, pages 452–459, 2005.

[75] Wikipedia. Set cover problem — Wikipedia, the free encyclopedia, 2011.

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

[77] J. Wu and I. Stojmenovic. Ad hoc networks. *IEEE Computer*, 37:29–31, 2004.

[78] Y. Wu, S. Fahmy, and N. B. Shroff. On the construction of a maximum-lifetime data gathering tree in sensor networks: NP-completeness and approximation algorithm. In *INFOCOM*, pages 356–360. IEEE, 2008.

[79] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Trans. Sen. Netw.*, 1(1):36–72, 2005.

[80] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 70–84, Rome, Italy, July 2001. ACM.

[81] F. Ye, G. Zhong, S. Lu, and L. Zhang. PEAS: A robust energy conserving protocol for long-lived sensor networks. In *3rd International Conference on Distributed Computing Systems (ICDCS '03), Rhode Island*, May 2003.

[82] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the IEEE Infocom*, pages 1567–1576, New York, NY, USA, June 2002. IEEE.

[83] W. Ye, F. Silva, and J. Heidemann. Ultra-low duty cycle MAC with scheduled channel polling. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 321–334, New York, NY, USA, 2006. ACM.

[84] H. Zhang and J. Hou. Maintaining Sensing Coverage and Connectivity in Large Sensor Networks. *Ad Hoc & Sensor Wireless Networks*, 1(1-2):89–123, January 2005.

[85] Q. Zhang, Z. Xie, W. Sun, and B. Shi. Tree structure based data gathering for maximum lifetime in wireless sensor networks. In *7th Asia-Pacific Web Conference*, pages 513–522, March 2005.

[86] S. Zou, I. Nikolaidis, and J. J. Harms. Efficient data collection trees in sensor networks with redundancy removal. In *Third International Conference Ad-Hoc, Mobile, and Wireless Networks, ADHOC-NOW 2004*, pages 252–265, 2004.