國立交通大學

資訊科學與工程研究所

博士論文

適用於無線感測網路之節能路由與中繼感測器安置演算法

Energy-Aware Routing and Relay Sensors Placing Algorithms in Wireless Sensor Networks

研究生：張志輝

指導教授：簡榮宏 博士

中華民國一００年十一月

適用於無線感測網路之節能路由與中繼感測器安置演算法

Energy-Aware Routing and Relay Sensors Placing Algorithms in Wireless Sensor Networks

研究生： 張志輝　　Student: Jyh-Huei Chang

指導教授： 簡榮宏 博士　　Advisor: Dr. Rong-Hong Jan

國立交通大學

資訊科學與工程研究所

博士論文

A Dissertation
Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in

Computer Science

November 2011

Hsinchu, Taiwan, Republic of China
中華民國一00年十一月

適用於無線感測網路之節能路由與中繼感測器安置演算法

研究生：張志輝　　　　　指導教授：簡榮宏 博士

國立交通大學資訊科學與工程研究所

摘要

無線感測網路 (wireless sensor networks, WSNs) 是由許多散佈於各地的感測節點 (sensor nodes) 所組成，主要用以蒐集各種環境資料，例如溼度、壓力、溫度等。

節能路由與中繼感測器安置是無線感測網路重要的研究議題，在本論文中我們提出數種適用於無線感測網路之節能路由與中繼感測器安置演算法 (relay sensors placing algorithm)，其中包括兩種節能叢集路由演算法 (energy-aware cluster-based routing algorithms), 兩種訊息渡輪路由演算法 (message ferry routing algorithms), 及一種中繼感測器安置演算法。

在節能路由 (energy-aware routing) 議題中, 叢集路由演算法 (cluster-based routing algorithm), 具有增加擴展性與有效性之優點, 如何以叢集路由演算法來最大化無線感測網路之生命週期 (lifetime) 亦是一個重要的研究議題。

針對節能路由議題，我們提出兩種適用於無線感測網路的節能叢集路由演算法 (cluster-based routing algorithm), 稱為 ECRA 及 ECR-2T。ECRA 及 ECRA-2T 之效能優於其他演算法。 這是因為 ECRA 及 ECRA-2T 旋轉叢集頭 (intra-cluster-heads)以平衡每個感測器的負荷量。 ECRA-2T 具有縮短傳輸距離的優點。

有關訊息渡輪路由 (message ferry routing) 議題, 在特殊環境裡, 例如戰場、疫區、廣域監視區等, 大多數路由演算法無法將接收到的訊息傳送至目的地。 因此, 如何在分離的無線感測網路 (partitioned wireless sensor networks) 中收集資料, 是一個重要的研究議題。

針對訊息渡輪路由議題，我們提出兩種適用於有容量限制(buffer-limited) 的無線感測網路的路由演算法, 稱為 MFRA1 及MFRA2。MFRA1 及 MFRA2 將原拜訪路徑 (initial visit sequence) 分割成一些次路徑 (sub-sequences), 在一個完整的拜訪順序中 (a complete sequence), 過載感測器 (overflow sensor) 會被 message ferry 拜訪兩次, 因此能繼續正常運作。 MFRA1 及MFRA2 在資料遺失量 (the amount of data loss) 之效能優於其他演算法。 因為其他方法忽略了感應器的過載 (overflow) 問題。

在中繼感測器安置 (relay sensors placing) 議題中, 隨機部署的無線感測網路中, 存在通訊缺口(communication gaps), 如何以最少的中繼感測器來保持網路連通, 是一個重要的研究議題。

針對中繼感測器安置 (relay sensors placing) 議題，我們提出一種適用於無線感測網路的中繼感測器安置演算法, 稱為 ERSPA。 ERSPA 在平均中繼感測器數量的效能優於 Minimum Spanning Tree 演算法及 Greedy 演算法。 Minimum Spanning Tree 演算法所需的平均中繼感測器數量約為 ERSPA 的兩倍。 這是因為 ERSPA 將中繼感測器安置在最佳位置以連通整個感測網路。

Energy-Aware Routing and Relay Sensors Placing Algorithms in Wireless Sensor Networks

Student: Jyh-Huei Chang Advisor: Dr. Rong-Hong Jan

Institute of Computer Science and Engineering
National Chiao Tung University

Abstract

In WSNs, there are spatially distributed sensors which cooperatively monitor environmental conditions, such as humidity, pressure, temperature, motion, or vibration, at different locations. Energy-aware routing and relay placing problems are important research issues in wireless sensor networks. In this dissertation, we design several efficient algorithms in wireless sensor networks, including two kinds of energy-aware cluster-based routing algorithms, two kinds of message ferry routing algorithms, and a relay placing algorithm.

In energy-aware routing problem, cluster-based routing protocols have special advantages that help enhance both scalability and efficiency of the routing protocol. Likewise, finding the best way to arrange clustering so as to maximize the network's lifetime is now an important research topic in the field of wireless sensor networks.

For energy-aware routing problem, we propose an energy-aware cluster-based routing algorithm (ECRA) for wireless sensor networks to maximize the network's lifetime. The ECRA selects some nodes as cluster-heads to construct Voronoi diagram and rotates the cluster-head to balance the load in each cluster. A two-tier architecture (ECRA-2T) is also proposed to enhance the performance of the ECRA. The simulations show that both the ECRA-2T and ECRA algorithms outperform other routing schemes such as direct communication, static clustering and LEACH. This strong performance stems from the fact that the ECRA and ECRA-2T rotate intra-cluster-heads to balance the load to all nodes in the sensor networks. The ECRA-2T also leverages the benefits of short transmission distances for most cluster-heads in the lower tier.

In message ferry routing problem, some particular environments such as battlefield, disaster recovery and wide area surveillance, most existing routing algorithms will fail to deliver messages to their destinations. Thus, it is an important research issue of how to deliver data in disconnected wireless sensor networks.

For message ferry routing problem, we propose two efficient message ferry routing algorithms in partitioned and buffer-limited wireless sensor networks, denoted as MFRA1 and MFRA2. MFRA1 and MFRA2 fix the overflow by partitioning the initial visit sequence into some sub-sequences such that the ferry visits the overflow node twice in the resulting sequence. The above process will continue until a feasible solution is found. Simulation results show that both MFRA1 and MFRA2 are better than other schemes in terms of the amount of data loss, because the other schemes neglect the case of sensor overflow.

In relay placing problem, randomly deployed sensor networks often make initial communication gaps inside the deployed area, even in an extremely high-density network. How to add relay sensors such that the underlying graph is connected and the number of relay sensors added is minimized is an important problem in wireless sensor networks.

For relay placing problem, we propose an efficient relay sensors placing algorithm (ERSPA) for disconnected wireless sensor networks. Compared with the minimum spanning tree algorithm and the greedy algorithm, ERSPA achieves better performance in terms of the number of relay sensors added. Simulation results show that the average number of relay sensors added by the minimal spanning tree algorithm is approximately two times that of the ERSPA algorithm. This is because ERSPA places the relay sensors in optimal places to connect the maximum number of initial connected sub-graphs such that the average number of relay sensors can be minimized.

Acknowledgements

Special thanks goes to my advisor Professor Rong-Hong Jan for his guidance in my dissertation work and the instructions on writing articles. Thanks also to all members of Computer Network Lab for their assistance and kindly help in both the research and the daily life during these years. Finally, I will dedicate this dissertation to my families for their love and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, the Micro-Electro-Mechanical Systems (MEMS) technologies have been booming. These MEMS technologies combined with advances in the wireless communication, make it possible to deploy low-cost, and low-power sensor networks. A wireless sensor network (WSN) consists of a large number of sensor nodes working together to monitor a region to obtain data about the environment [1]. The applications for WSN include environment exploration, military target tracking and surveillance, natural disaster relief, biomedical health monitoring, and seismic sensing. There are two types of WSN: structured and unstructured. In a structured WSN, the sensor nodes are deployed in a pre-planned manner. The advantage of a structured network is that fewer nodes can be deployed with lower network maintenance and management cost. An unstructured WSN is one that contains a dense collection of sensor nodes. In an unstructured WSN, network maintenance such as managing connectivity and detecting failures is difficult [2].

Many studies focus on the following potential applications for WSN:

1) Environment exploration: In environmental monitoring, a WSN can monitor air soil and water[3].

2) Military target tracking and surveillance: In military target tracking and surveillance, a WSN can assist in intrusion detection and identification such as spatially-correlated and coordinated troop and tank movements [5,6].

3) Natural disaster relief: In natural disasters, sensor nodes can detect the environment to forecast disasters before they occur [7].

4) Biomedical health monitoring: In biomedical applications, surgical implants of sensors can help monitor a patient's health [8,9].

5) Seismic sensing: For seismic sensing, sensor nodes can detect the development of eruptions and earthquakes [10].

In spite of these diverse applications, most sensor networks encounter the following operational challenges:

1) Ad hoc deployment: The sensor networks should be able to cope with the resultant distribution and connection between the nodes.

2) Unattended operation: In most cases, once deployed, sensor networks have no human intervention. Hence the sensor nodes are responsible for reconfiguration in case of any changes.

3) Untethered: The sensor nodes are not connected to any energy source. There is only a finite source of energy, which must be optimally used for processing and communication. The communication dominates processing in energy consumption. Thus, in order to make optimal use of energy, communication should be minimized as much as possible.

4) Dynamic changes: Dynamic environmental conditions require the sensor networks to be adaptive in nature to change connectivity and node failure [4].

A number of studies propose solutions to one or more of the above challenges for WSNs [1, 2]. In this dissertation, we focus on the following issues which are important for WSNs:

1) Energy consumption: Energy consumption is the most important factor to determine the lifetime of a sensor network because sensor nodes only have a small and finite source of energy. Many solutions, both hardware and software related, have been proposed to optimize energy usage.

2) Routing: Communication costs play a great role in deciding the routing technique. Conventional routing protocols have several limitations when being used in sensor networks due to the energy constrained nature of these networks. The routing protocols designed for sensor networks should be able to overcome the energy constraint and look at newer ways of conserving energy to increase the lifetime of the network.

3) Localization: In most of the cases, sensor nodes are deployed in an ad hoc manner. It is up to the nodes to identify themselves in some spatial co-ordinate system. This problem is referred to as localization. Many studies proposed the solutions to ensure **optimum placement of nodes**. Mostly, problems arise due to the unpredictable nature of environmental conditions. Nodes thus will also need to be able to adapt to environmental changes.

Thus, in this dissertation, we consider three important problems which are energy-aware routing problem, message ferry routing problem and relay placement problem.

In energy-aware routing problem, the power for sensor nodes comes from their batteries. Thus, finding the best use for the limited battery power is a crucial research issue in wireless sensor networks. Cluster-based routing protocols have special advantages that help enhance both scalability and efficiency of the routing protocol. Likewise, finding the best way to arrange clustering so as to maximize

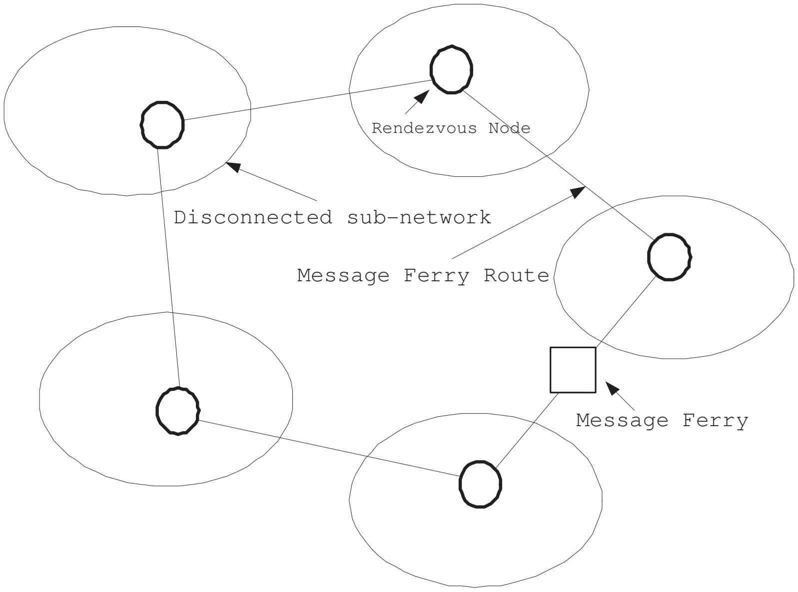


Figure 1.1: Example of message ferry scheme.

the network's lifetime is now an important research topic in the field of wireless sensor networks.

In message ferry routing problem, a partitioned wireless sensor networks, most existing routing algorithms will fail to deliver messages to their destinations. The Message Ferry scheme is an approach for message delivery in the disconnected wireless sensor network. As shown in Fig. 1.1, there is a wireless sensor network with several separated sub-networks. Each sub-network has a rendezvous node to connect and buffer the data. A special node, called Message Ferry, visits these rendezvous nodes to collect the buffered data of the sensing field based on a predefined route. The message ferry route problem is to find a route for message ferry to visit rendezvous nodes and collect data.

The previous schemes focus on the routing problems under the assumption that the buffer size of each sensor node is unlimited. These schemes do not deal with the condition that the sensor may overflow and thus the sensing data will lose.In real applications, there are different kinds of sensors such as surveillance sensors and data sensors. The surveillance sensor has high sampling rate to capture video messages. The data sensor has low sampling rate to collect temperature or noise data. In such a sensing environment, each sensor has a limited buffer size and thus the surveillance sensor may overflow before the message ferry visits all sensor nodes. It is a serious problem that a sensor loses critical messages due to overload. Therefore, how to avoid overflow should be an important research issue in the message ferry routing problem.

In relay placement problem, randomly deployed sensor networks often make initial communication gaps inside the deployed area, even in an extremely high-density network. How to add relay sensors such that the underlying graph is connected and the number of relay sensors added is minimized is an important problem in wireless sensor networks.

Our solution methods can connect the heterogeneous wireless sensor networks and gather data to base station.

The rest of this dissertation is organized as follows. In Chapter 2, we introduce the energy-aware routing problem and our solution methods for general WSNs. Next, we illustrate the message ferry routing problem and our solution methods for partitioned and buffer-limited WSNs in Chapter 3. The relay placement problem and our solution methods for disconnected WSNs will be described in Chapter 4. Finally, a conclusion is given in Chapter 5.

Chapter 2

Energy-aware routing problem for general WSNs

In this chapter, we first describe related works of energy-aware routing problem. Then, we illustrate problem formulation and our solution methods. Finally, we present simulation results of our scheme.

2.1 Related works of energy-aware routing problem

Many studies have focused on saving energy in different ways such as reducing the power spent on the modulation circuits [11], or managing the power usage on the MAC layer of sensor nodes [12-13]. However, these schemes focused of the individual device, and that approach is too narrow when working with wireless sensor networks. Since sensor nodes have limited transmitting ranges, only a few nodes can communicate directly with the sink node. In most cases, the sensor nodes gather sensing data which must then be forwarded by the other node to the sink node. However, these cumbersome relaying operations consume too much energy, thus causing the relay nodes to rapidly expend much of their power. Therefore, developing a load-balanced routing algorithm to maximize the network's lifetime has become an important research topic.

A large number of routing protocols [14-20] for wireless sensor networks have been proposed, but most are flawed in one way or another. In the fixed path schemes [14-15], sensor nodes arrayed in a fixed path will consume much energy and get exhausted rapidly because they continually provide relaying service. The flooding scheme consumes too much energy for relaying duplicate packets. Source routing schemes [16-17] solved some of the drawbacks of the flooding approach; however, they can not operate well when the number of hops between the source and sink is large. Energy-aware, multi-path routing schemes [18-20] have the advantage of sharing the energy among all the sensors in the wireless networks. Nevertheless, the chief disadvantage of multi-path routing schemes [18-19] is that the sensor nodes only keep a local view of energy usage and the nodes in the network can not have an even traffic dispatch.

In addition, many studies have focused on cluster-based energy-efficient routing protocol for wireless sensor networks [21-35]. Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [31] prolongs the network lifetime with a chain topology. But the delay is significant although the energy is saved. Hybrid Energy-Efficient Distributed Clustering (HEED) [32] considers a hybrid of residual energy and communication cost when selecting cluster-head. A sensor has highest residual energy can become a cluster-head. However, if the residual energy of the sensors in a cluster is nearly the same, it takes many iterations and expends much energy to elect cluster-head. The Low Energy Adaptive Clustering Hierarchy (LEACH) [33-34] randomly selects some nodes as cluster-heads and rotates the cluster-head to distribute the load to all sensors in the wireless sensor networks. Its

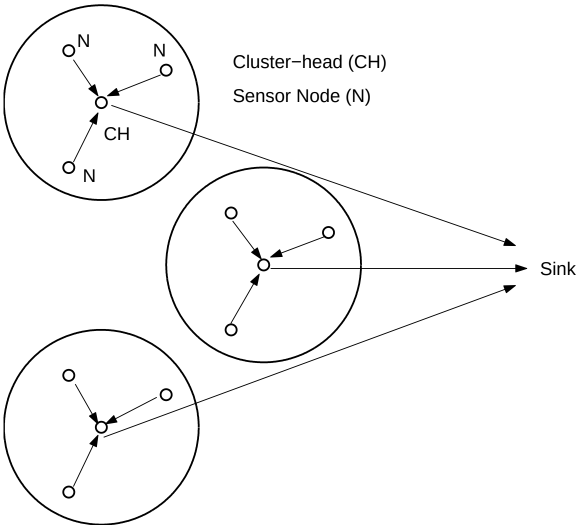


Figure 2.1: Wireless sensor network organized in clusters.

performance is better than that of the direct communication and static clustering routing protocols. Figure 2.1 shows an example of the cluster-based routing scheme for wireless sensor networks. In Figure 2.1, each cluster has one cluster-head. The non-cluster-head nodes transmit their sensing data to cluster-heads which then forward the aggregated data to the sink node. The use of clusters leverages the benefits of short transmission distances for most nodes. The cluster-head acts as a fusion point to aggregate the sensing data so that the amount of data that is actually transmitted to the sink node is reduced [23-24]. Thus, network clustering can increase system lifetime and energy efficiency. Cluster-based routing protocols have special advantages: they can enhance the scalability and efficiency of the routing protocol to reduce the routing complexity [35], reduce the complexity of location management [36], and improve the power control procedure [37]. However, LEACH may also have several problems: First, if the coverage of the cluster-heads is too small, then some cluster-heads may not have any members in their clusters. Second, LEACH has a long transmission range between the cluster-heads and the sink node. Third, the LEACH requires global cluster-heads rotation. This cluster-head selection greatly increases processing and communication overhead, thereby consuming more energy.

2.2 Problem formulation and network model

The network model and assumptions of our research are described as follows: 1) All sensors are location aware. That is, they can convey their location information to the base station in the initialization phase (phase 0). 2) The base station has a power supply so we assume it has infinite energy. Therefore, the energy required for the base station to inform each cluster-head can be ignored. 3) Base stations can compute the residual energy of all sensors in each round according to their location and the amount of transmission data.

The energy model of our study is the same as in [33]. In this energy model, the electronic energy $E_{elec} = 50 \quad nJ/\text{bit}$ in needed to operate the transmitter or receiver circuit. The transmitter amplifier is $\epsilon_{amp} = 100\ pJ/bit/m^2$. Equations (2.1) and (2.2) are used to calculate the transmission energy, denoted as $E_{Tx}(k,d)$, required for a k bits message over a distance of d,

$$\begin{aligned} E_{Tx}(k,d) &= E_{Tx\_elec}(k) + E_{Tx\_amp}(k,d) && (2.1) \\ &= E_{elec} \times k + \epsilon_{amp} \times k \times d^2. && (2.2) \end{aligned}$$

To receive this message, the energy required is:

$$E_{Rx}(k) = E_{Rx\_elec}(k) = k \times E_{elec} \tag{2.3}$$

where E_{Tx_elec} is the energy dissipation of transmitter electronics and E_{Rx_elec} is the energy dissipation of receiver electronics. E_{Tx_amp} is the energy of the transmitter amplifier. Assume that $E_{Tx\_elec} = E_{Rx\_elec} = E_{elec}$. From equation (2.3), one can see that receiving data is also a high overhead procedure. Thus, the number of transmission and receiving operations must be cut to reduce the energy dissipation. We also assume that the radio channel is symmetric such that the energy required to transmit a message from node i to node j is the same as the energy required to transmit a message from node j to node i for a given signal-to-noise ratio.

2.3 Solution methods for energy-aware routing problem

2.3.1 Three phases of ECRA

Our ECRA algorithm includes three phases: clustering, data transmission and intra-cluster-head rotation. The details of the algorithm are given as follows.

Phase 1: Clustering

First, we define of the Voronoi diagram and Centroidal Voronoi Tessellation (CVT) [38-40]. Consider an open set $\Omega \subseteq \Re^2$ and a set of points $\{z_i\}_{i=1}^{n}$ belonging to $\bar{\Omega}$ where $\bar{\Omega}$ is the closed set of Ω. Let $|.|$ denote the Euclidean norm in $\Re^2$. The Voronoi region V_i corresponding to the points z_i is defined by

$$V_i = \{x \in \Omega | \; |x - z_i| < |x - z_j| \;\text{ for } j = 1, ..., n, j \neq i\} \tag{2.4}$$

where $V_i \cap V_j = \emptyset$ for $i \neq j$ and $\cup_{i=1}^{n} \bar{V}_i = \bar{\Omega}$. The set of $\{V_i\}_{i=1}^{n}$ is a Voronoi diagram of Ω and each V_i is referred to as the Voronoi region corresponding to z_i.

The points $\{z_i\}_{i=1}^{n}$ are called generators.

CVT is a Voronoi tessellation whose generating points are the centroids of mass for their corresponding Voronoi regions. Formally, CVT can be defined as follows. Given a region $V_i \subseteq \Re^2$ and a density function $\rho(x)$, defined in V_i, the mass centroid z_i^* of V_i is defined by

$$z_i^* = \frac{\int_{V_i} x\rho(x)dx}{\int_{V_i} \rho(x)dx} \qquad \text{for } i = 1, ..., n. \tag{2.5}$$

Given n points $\{z_i\}_{i=1}^{n}$, if the points $z_i{=}z_i^*$ for $i = 1, ..., n$, then we call the Voronoi tessellation defined by equation (2.4) as a CVT. That is, the points z_i that serve as the generators for Voronoi regions V_i are themselves the mass centroids of those regions. Figure 2.2 shows the CVTs with $\rho(x) = c$ for $n = 2, 3, 4, 5$.

We apply the following two steps to partition the sensor nodes into n clusters.

Step 1: Given sensing field Ω, a positive integer n, and a density function $\rho(x) = c$, construct a centroidal Voronoi tessellation such that V_i is the Voronoi region for z_i^* and z_i^* is the mass centroid of V_i for each i. That is, the sensing field is partitioned into n Voronoi regions.

Step 2: Let $w_i, i = 1, \ldots, m$, denote the sensor nodes in sensing field Ω.

(a) For each node w_i, if $w_i \in V_j$, then we assign node w_i to cluster C_j.

(b) For each $C_j, j = 1, \ldots, n$, find a sensor node w_j^* that is closest to z_j^*, the mass centroid of V_j. Then, we choose sensor node w_j^* as the initial cluster head of cluster C_j.

Figure 2.3 shows that these cluster-heads are located nearest their corresponding centroidal points. The advantage of the above clustering method is that

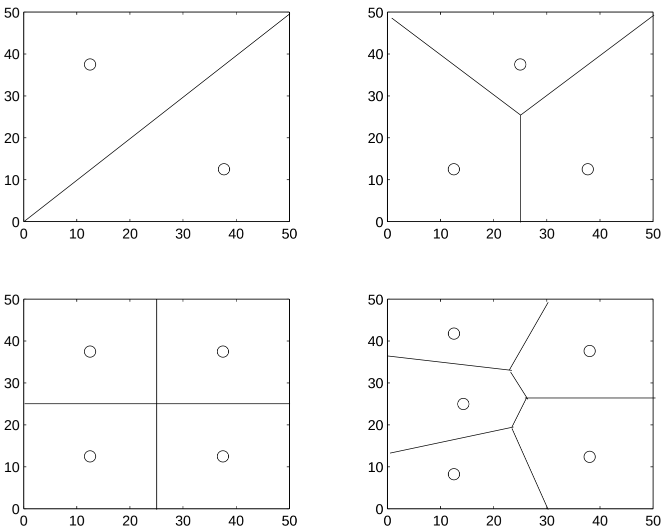


Figure 2.2: The examples of CVTs for $n = 2, 3, 4, 5$.

each cluster head has a nearly equal number of members; this is because the sensor nodes are uniformly distributed. In contrast with Figure 2.3, Figure 2.4 shows the Voronoi diagram in which the generating points were selected randomly. Note that the clusters have diverse number of members.

In the ECRA scheme, we assume that the sensor nodes are location-aware. That is, the base station knows every sensor node's location. The base station constructs CVT for the sensing field. Each cluster has only one cluster-head. The base station tells each cluster-head which nodes are its member. The cluster-head broadcasts an advertisement to their members. By listening to the advertisement, each node knows which cluster it belongs to. Then, the sensor node sends an acknowledgement to its cluster-head and confirms that it will be a member of the cluster. During this time, all cluster-heads must remain in active.

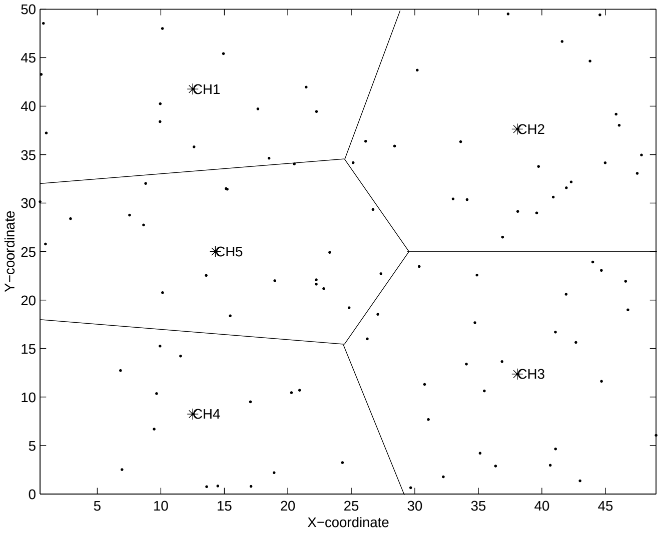


Figure 2.3: Voronoi diagram of the 100-node random sensor network with centroidal points.

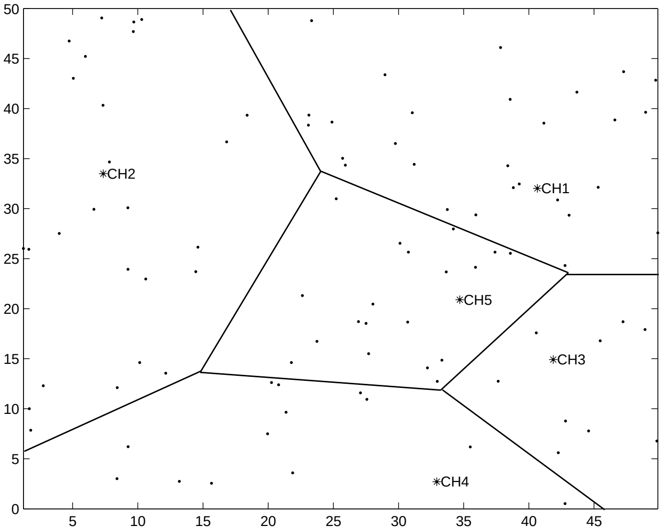


Figure 2.4: Voronoi diagram of the 100-node random sensor network without centroidal points.

Phase 2: Data transmission

When the clusters are created, data transmission can begin. The nodes use single hops to communicate with their cluster-heads, and the cluster-heads communicate with the base station. Each node has M bits messages to transmit. The non-cluster-head node can turned-off until its allocated transmission time in order to minimize the energy usage. When all data from the nodes have been received, the cluster-head aggregates the total data into a single message to reduce the amount of information transmitted to the base station.

Phase 3: Intra-cluster-head rotation

When a round is ended, next rotate the cluster-head within the same cluster based on a parameter called O_{ij} which is a function of communication cost $E_{d_{ij}}$ and residual energy E_{ij}^{new}. The distance d_{ij}, $i = 1, \ldots, n, j = 1, \ldots, |C_i|$, represents the distance from node j in cluster C_i to the base station and is given as

$$d_{ij} = \sqrt{(x_{ij} - x^*)^2 + (y_{ij} - y^*)^2} \tag{2.6}$$

where (x_{ij}, y_{ij}) is the position of node j in cluster C_i and (x^*, y^*) is the position of the base station. The residual energy E_{ij}^{new} is defined as

$$E_{ij}^{new} = E_{ij}^{old} - E_{ij}^{expend} \tag{2.7}$$

where E_{ij}^{old} is the residual energy of node j in cluster C_i at the beginning of the current round. E_{ij}^{expend} is the energy expended by the node in the current round. $E_{d_{ij}}$ is the energy expended by the cluster-head to transmit data to base station. Then, we define the parameter O_{ij} as

$$O_{ij} = \frac{E_{ij}^{new}}{E_{d_{ij}}}, \qquad i = 1, \ldots, n, j = 1, \ldots, |C_i| \tag{2.8}$$

For each cluster C_i, we find

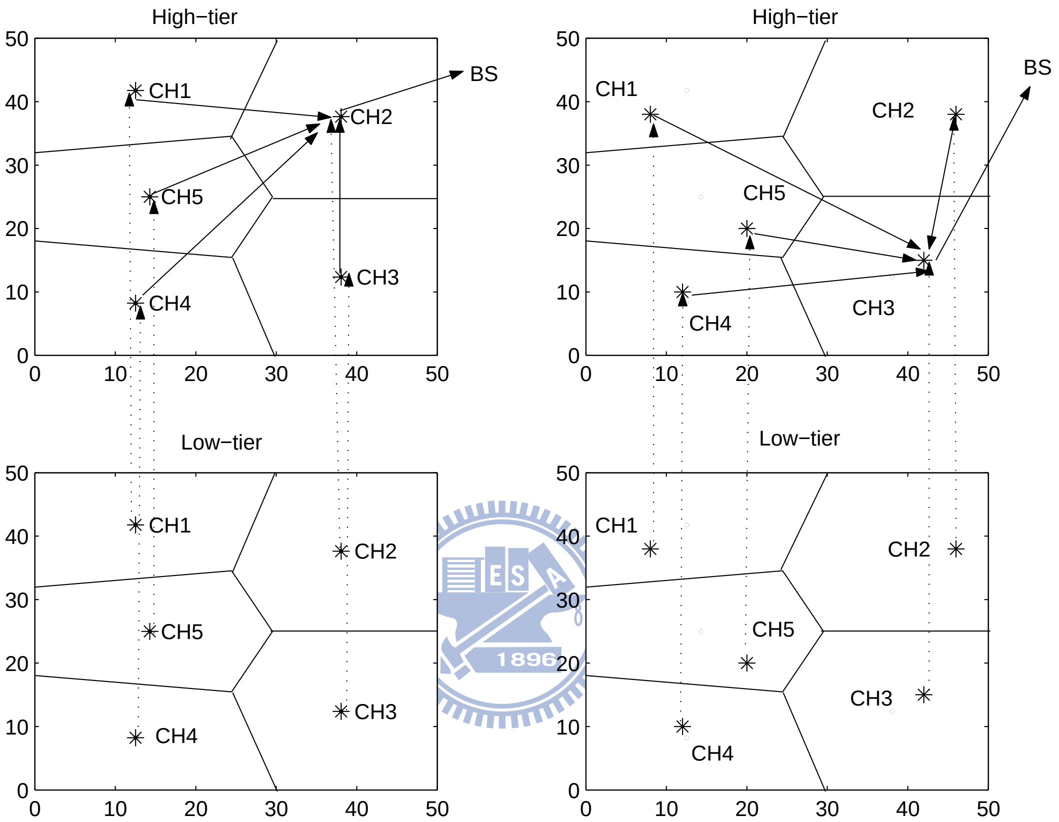
$$O^{(i)} = \max\{O_{ij} | j = 1, \ldots, |C_i|\}.$$

Node j in cluster C_i with the value of $O^{(i)}$ will become a cluster-head of cluster C_i at the next round. That is, when all data are received in the current round, the base station first calculates the value of O_{ij} for each node j in cluster C_i, then finds $O^{(i)}$ for each C_i, and finally informs the node with value $O^{(i)}$ to become the new cluster-head at the next round. Note that the base station has the location of each node and it also knows that each sensor node has sent M bit messages, and thus the base station can calculate the value of O_{ij}.

When the current round is ended, the role of the cluster-head will rotate to the node with value $O^{(i)}$ that is designated by the base station. Then, the new cluster-head begins to advertise using the method given in phase 1.

2.3.2 Enhancement of ECRA

The ECRA can be enhanced by adding an extra tier, called a high tier, on top of the original architecture (see Figure 2.5). The high tier has only one cluster. All cluster-heads in the low tier are also the members in the high tier. This architecture is called a two-tier ECRA (denoted as ECRA-2T). The nodes in the high tier forward their aggregated data to the node with the maximal remaining energy, called the *main cluster-head*. The main cluster-head transmits the aggregated data to the sink. When a round is over, rotate the cluster-head of the low-tier in the sensing field based on the parameter O_{ij} (see equation (2.8)). The members of



(a) T

(b) T + d

Figure 2.5: The operation of two-tier architecture in enhanced ECRA, where T isthe current round and $T + d$ is the next round, and so on.

the high-tier in the next round consist of these cluster-heads. Figure 2.5 illustrates the high-tier operation in ECRA-2T. In the current round T, CH_2 is the main cluster-head. In next round $T + d$, CH_3 has a maximal remaining energy that is selected as the main cluster-head, and so on.

2.4 Simulation results

2.4.1 Performance metrics and environment setup

This section presents the performance analysis of the ECRA algorithm. The performance metrics are given as follows.

1) The lifetime for the first node to die (FND): FND is defined as the time required for the first node to run out of energy. The non-cluster-head nodes transmitted their sensing data to the cluster-head. The cluster-heads forwarded their aggregated data to the sink periodically. We use the number of *rounds* to represent the network lifetime of FND. A round is defined as all nodes in the wireless network that finish returning their gathered data to the sink. The time interval between two rounds is assumed to be large enough for the last node to return its sensing data.

2) The lifetime for the last node to die (LND): LND is defined as the time required for the last node to run out of energy, at which time the network crashed. We also use the number of *rounds* to represent the network lifetime of LND.

3) The total energy dissipation (TED): This value is defined as the energy dissipation for all nodes that finish returning their gathered data.

4) The cost of energy×delay: This value is the cost for each round of data gathering from sensor node to sink node. The energy cost can be calculated from the

energy model described in Section 2. The delay cost can be calculated as units of time. On a link with 2Mbps, a message of 2000 bits can be transmitted in 1ms. Therefore each unit of delay will be about 1 ms for a sensor node with a single channel. We assume that the delay cost is 1 unit for each message of 2000 bits transmitted.

We evaluate the performance of our study implemented with C++ and $MATLAB$. Four different sizes of deploying regions were simulated: $50 \times 50\, m^2$, $100 \times 100\, m^2$, $150 \times 150\, m^2$ and $200 \times 200\, m^2$. In each region, 100 nodes were deployed by uniform distribution. Assume that the energy model is the same as in [33]. The electronics energy is $E_{elec} = 50\, nJ/$bit. $\epsilon_{amp} = 100\, pJ/bit/m^2$. The energy of data aggregation is 5 nJ/bit/message. The cluster-heads use a 1-bit message to inform their members in each round. Then, the members send their data to their cluster-heads. The negotiation energy consumption is included in each round. The sink node was located at the position $((x, y) = (25, -100))$. Each sensor has 2000 bits of data sent to the base station during each round.

First, we determined the number of cluster header n. Note that if n is small, then the average length from sensor node to its cluster is large. This means that the energy costs between sensor node and cluster head is large. However, if n is large, then the total energy costs between cluster header and the base station is large. By simulation, Figure 2.6 shows the normalized total energy dissipation related to different percentages of cluster-heads in ECRA. From Figure 2.6, we learn that the normalized total energy dissipation is minimized at the 5 % of the total number of sensor nodes for ECRA. Thus, we chose $n = 5$ from 100 sensor nodes as cluster headers.

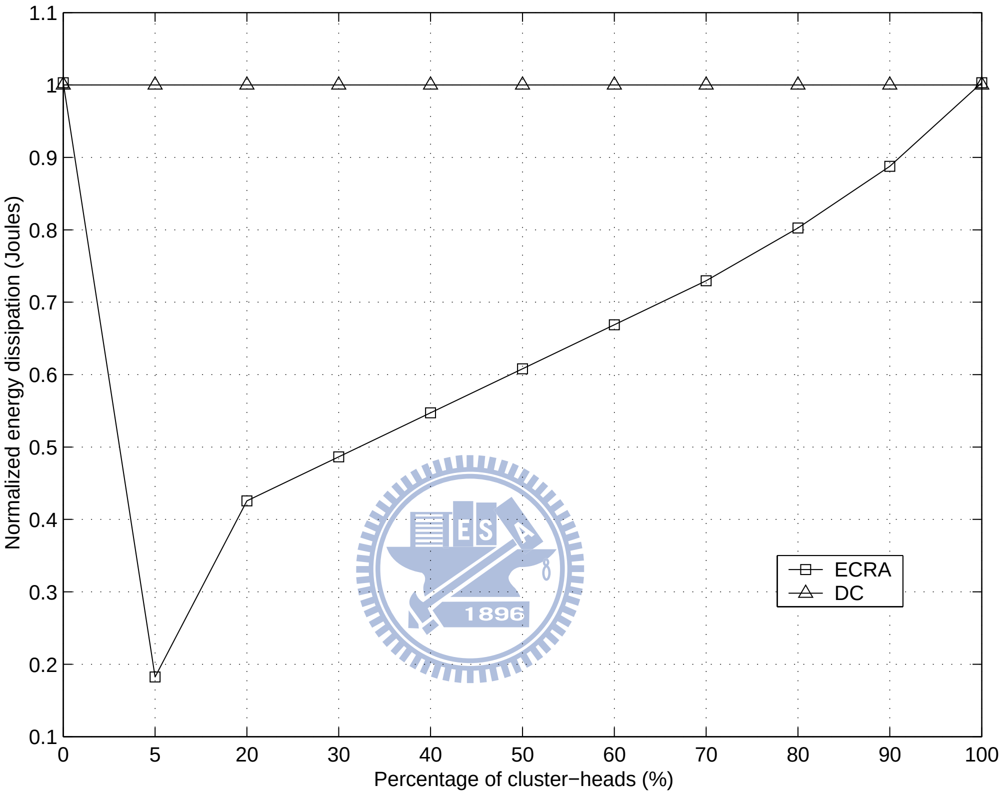


Figure 2.6: The normalized total energy dissipation related to different percentage of cluster-heads in ECRA.

2.4.2 Numerical results

Comparisons of the four performance metrics were made for six schemes: the direct communication (DC), static clustering (SC), LEACH, PEGASIS [21], HEED [32], ECRA and ECRA-2T. The results are given as follows.

1) The lifetime of FND under different initial energy levels

   Figure 2.7 shows that of the lifetime of ECRA-2T in FND is better than LEACH, HEED, direct communication, and static clustering if the initial energy of each sensor is 1 J. Figure 2.8 shows the lifetime of FND under different methods with different initial energy of each node. Overall, the lifetime of FND increases when the initial energy of each sensor is greater. Both ECRA and ECRA-2T have better performance than other schemes. The lifetime of FND in ECRA-2T is approximately twice that of LEACH, but over eight times than that of direct communication and static clustering. That is, ECRA-2T gives better performance than DC, SC, HEED and LEACH in the lifetime of FND.

2) The lifetime of LND under different initial energy levels

   Figure 2.9 shows that the lifetime of ECRA-2T in LND is better than that of LEACH, HEED, PEGASIS, direct communication and static clustering if the initial energy of each sensor is 1 J. Figure 2.10 shows the lifetime of LND under different methods with different initial energy of each node. Overall, the lifetime of LND increases when the initial energy of each sensor is greater. Both ECRA and ECRA-2T have better performance than other schemes. The lifetime of LND in ECRA-2T is approximately 2.5 times longer than LEACH but over nine times greater than direct communication and static clustering. The results show that ECRA-2T gives better performance than DC, SC, PEGASIS, HEED and LEACH in the lifetime of LND. From Figures 2.7 and 2.9, note that if a scheme shares the load evenly with all sensor nodes in the network, it can achieve a longer lifetime.

3) Total energy dissipation under different network diameters

Figure 2.11 shows that ECRA-2T uses much less energy compared to direct communication. In other words, using clusters lets one leverage the benefits of short transmission distances for most nodes and distributes the energy among the sensor nodes in the network, thus reducing total energy dissipation.

Figure 2.12 shows the number of alive nodes under different routing schemes. This number decreases when the number of rounds is greater. From Figure 2.12, we note that the performance of ECRA-2T is better than that of LEACH. According to the above analysis, our ECRA-2T algorithm has better performance than do other schemes regarding system lifetime and energy dissipation. These simulation results also show that ECRA-2T has the advantages of balanced loads and saved energy.

4) The cost of energy×delay: Figure 2.13-(a) shows that ECRA-2T is better than LEACH, PEGASIS, and direct communication for a 50 m × 50 m network in terms of energy×delay. Figure 2.13-(b) shows that ECRA-2T is also better than LEACH and direct communication for a 100 m × 100 m

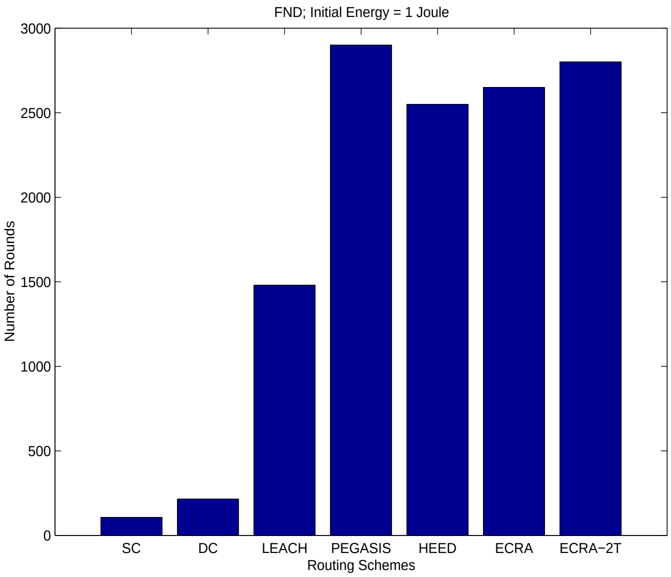


Figure 2.7: The lifetime of first node died (FND) under different methods. The initial energy of each sensor is 1 Joule.

network in terms of energy×delay. This is because the two-tier architecture leverages the benefits of short transmission distances for most cluster-heads in the low-tier.

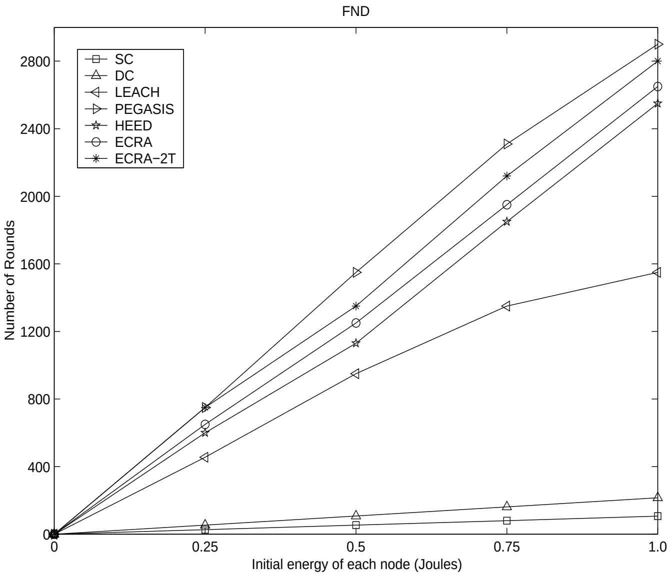


Figure 2.8: The lifetime of first node died (FND) using different amounts of initial energy for the sensors.

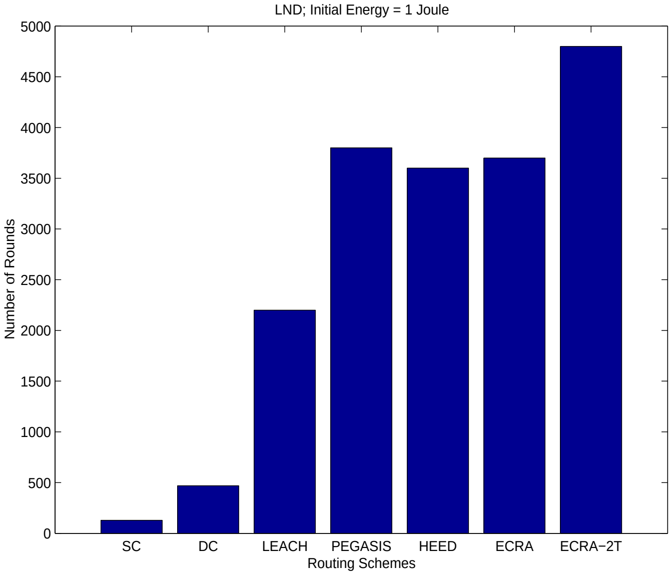


Figure 2.9: The lifetime of last node died (LND) under different methods. The initial energy of each sensor is 1 Joule.

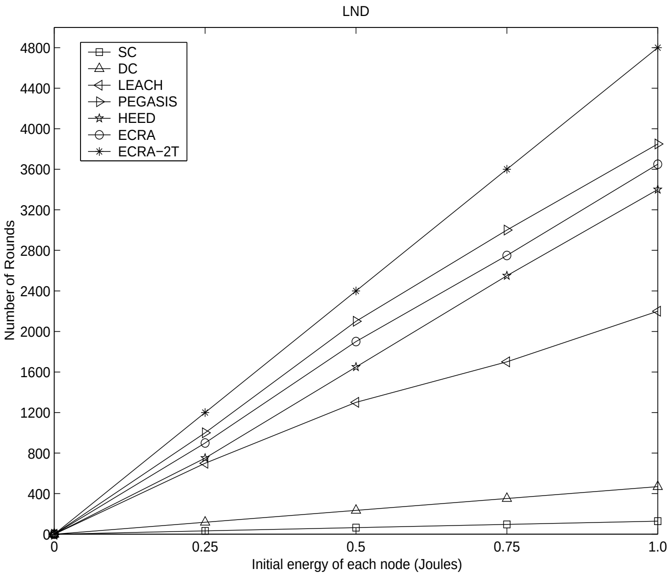


Figure 2.10: The lifetime of last node died (LND) using different amounts of initial energy for the sensors.

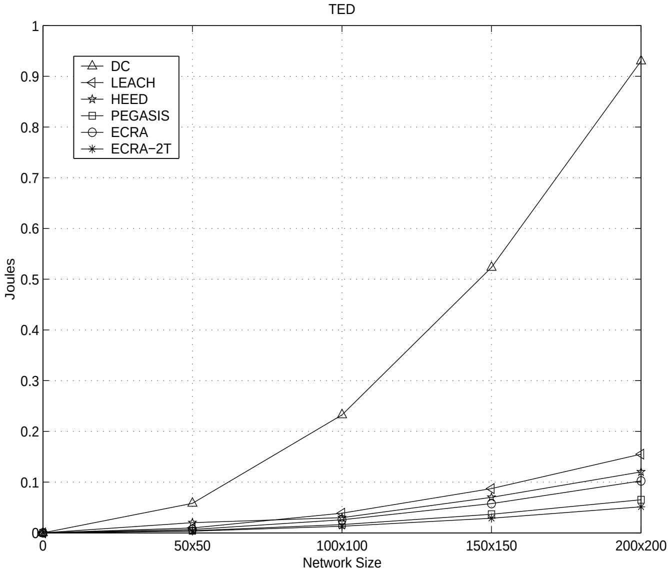


Figure 2.11: Total energy dissipation (TED) using direct communication, LEACH, ECRA, and ECRA-2T. The messages are 2000 bits.

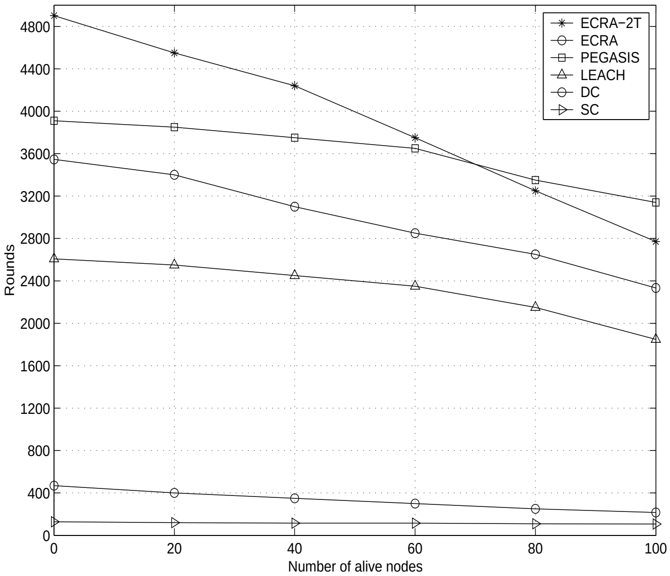


Figure 2.12: Number of alive nodes under different routing schemes. The initial energy of each sensor is 1 Joule.

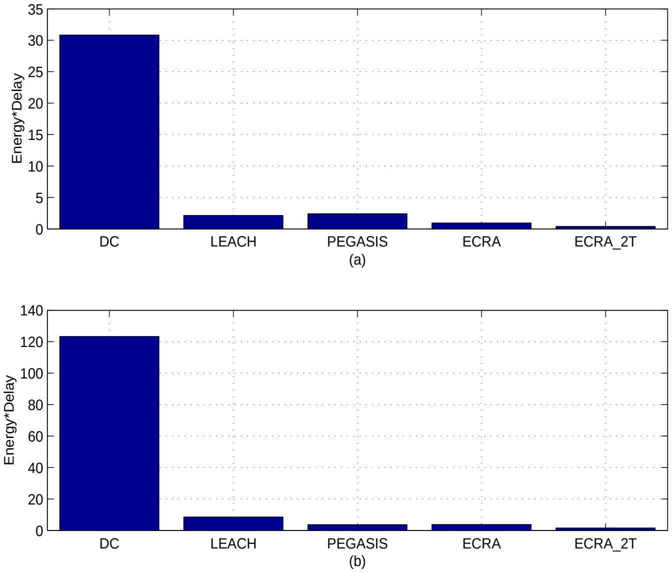


Figure 2.13: (a)Energy×Delay cost for different routing schemes in a 50 m x 50 m network. (b)Energy×Delay cost for different routing schemes in a 100 m x 100 m network.

Chapter 3

Message ferry routing problem for partitioned and buffer-limited WSNs

In this chapter, we first describe related works of message ferry routing problem. Then, we illustrate problem formulation and our solution methods. Finally, we present simulation results of our scheme.

3.1 Related works for message ferry routing problem

Several schemes have been proposed to solve the message ferry route problem in partitioned wireless ad hoc networks [41-45]. In [41], the authors propose a message ferry scheme to solve the data delivery problem in high-partitioned wireless ad hoc networks. In [42], the authors introduce a non-randomness in the movement of nodes to improve data delivery performance and reduce the energy consumption in sensor nodes. Epidemic routing [44] is also a well-known routing method for partitioned wireless ad hoc networks. In this scheme nodes forward messages to other nodes they meet. However, this scheme transmits many redundant messages.

Compared to Epidemic routing, message ferry scheme is very efficiency in data delivery and energy consumption. However, the synchronization between nodes and ferry is a problem in the message ferry scheme. An optimized way-points (OPWP) algorithm [43] was proposed. It generates a ferry route to achieve good performance without any online collaboration between nodes and the ferry. OPWP outperforms other naive ferry routing schemes.

Many studies deal with efficient routing for intermittently connected mobile ad hoc networks [46-53]. In [46-47], the authors proposed a routing scheme with two types of ferries and gateways. This scheme improves delivery rate and delay without online collaboration between ferry and mobile nodes. However, the local message ferry, global message ferry and gateway nodes of this scheme need more resources to buffer the messages. In [48], the authors proposed single-copy routing schemes that use only one copy per message, and hence significantly reduce the resource requirements of flooding-based algorithms. In [49], the authors proposed a routing scheme that sprays a few message copies into the network, and then routes each copy independently toward the destination. This scheme can reduce the delay in flooding-based scheme.

Some studies deal with the data scheduling problem for message ferry [54-55]. In [54], the authors present an elliptical zone fording (EZF) scheme for a ferry to deliver messages among partition nodes that are moving around. EZF scheme gives priority to urgent messages that are already in the message ferry buffer. However, there may be urgent message waiting to be picked up at other nodes that have closer deadlines than the most urgent message in the delivery up queue. Three ferry routes with look-ahead schemes were proposed in [55] to overcome the drawback of EZF scheme. The dynamic look-ahead scheme provides the best performance compared with other schemes.

Several studies focus on mobile element scheduling problem [56-58] for wireless sensor networks (WSNs). The mobile element works as a mobile sink in WSNs, which is similar to the message ferry. In [56], the authors present an architecture to connect sensors in sparse sensor networks. The advantage of this scheme is the potential of large power savings that can occur at the sensors because communication takes place over a short-range. Its disadvantage is the increasing latency because sensors have to wait for a mobile element to approach before the transfer can occur. In [57], the authors proposed a load balancing algorithm to balance the number of sensor nodes that each mobile element services. The network scalability and traffic may make a single mobile element insufficient. Using multiple mobile elements scheme can overcome this problem.

3.2 Problem formulation and network model

3.2.1 Network model

The network model and assumptions of our research are described as follows.

1) A set of sensors are randomly deployed in a two-dimensional sensing area. These sensors form a disconnected network. The buffer size of each sensor is constant.

2) Each sub-network has a sensor node, called rendezvous node, which collects and buffers the sensing data from other nodes. Each sensor node has different sampling rate to sense data and has to deliver sensing data to the rendezvous node in its sub-network.

3) A message ferry visits each rendezvous node to collect data. The moving speed of the message ferry is constant. The message ferry works as a mobile sink and has infinite memory. When the message ferry visits a rendezvous node, the buffer

of the rendezvous node will refresh to empty. Message ferry and sensor nodes have the same transmission range.

4) The data transmitting time from a rendezvous node to the message ferry is ignored.

5) Without loss of generality, we assume that there is only one sensor node in each sub-network. Thus, the terms, rendezvous node and sensor node, are used interchangeably in the following sections.

3.2.2 Problem formulation

We formulate the problem as follows. Let $N = \{n_1, n_2, ..., n_m\}$ be a set of m sensors in a two-dimensional sensing field, and let d_{ij} be the distance between nodes n_i and n_j. The message ferry visits each sensor node at a constant speed to collect sensing data. We want to find a visit sequence for the message ferry such that the buffer of each sensor does not overflow between two visits. A complete sequence is defined as the visit sequence of message ferry which visits every sensor node at least once and returns to the start sensor node. That is, a sequence $n_{i_1}, n_{i_2}, \ldots, n_{i_k}, k \geq m$, is said to be a complete sequence if $n_{i_1} = n_{i_k}$ and $\cup_{j=1}^{k}\{n_{i_j}\} = N$. The message ferry can repeat this complete sequence again and again if the sequence is feasible. Thus, any sensor node n_{i_j} in the sequence $n_{i_1}, n_{i_2}, \ldots, n_{i_{k-1}}, n_{i_1}$ can act as start node and the resulting sequence $n_{i_j}, n_{i_{j+1}}, \ldots, n_{i_{k-1}}, n_{i_1}, n_{i_2}, \ldots, n_{i_j}$ is equivalent to the sequence $n_{i_1}, n_{i_2}, \ldots, n_{i_{k-1}}, n_{i_1}$. Next, we define the travel time t_{i_1} of message ferry between two visits for node n_{i_1} with respect to the complete sequence $n_{i_1}, n_{i_2}, \ldots, n_{i_{k-1}}, n_{i_1}$ as follows.

1. If $n_{i_j} \neq n_{i_1}, j = 2, \ldots, k-1$, then

$$t_{i_1} = \frac{(\sum_{j=1}^{k-2} d_{i_j i_{j+1}}) + d_{i_{k-1} i_1}}{s}$$

where s is the speed of massage ferry.

2. If n_{i_1} in the complete sequence $n_{i_1}, n_{i_2}, \ldots, n_{i_{k-1}}, n_{i_1}$ repeats $p+2, (p \geq 1)$ times, say $n_{i_{h_1}} = \cdots = n_{i_{h_p}} = n_{i_1}$, then

$$t_{i_1} = \max\{\frac{\sum_{j=1}^{h_1-1} d_{i_j i_{j+1}}}{s}, \frac{\sum_{j=h_1}^{h_2-1} d_{i_j i_{j+1}}}{s}, \ldots, \frac{(\sum_{j=h_p}^{k-2} d_{i_j i_{j+1}}) + d_{i_{k-1} i_1}}{s}\}$$

   In this case, we say the complete sequence $n_{i_1}, n_{i_2}, \ldots, n_{i_{k-1}}, n_{i_1}$ is partitioned into p sub-sequences on the node n_{i_1}.

Formally, we define the *message ferry routing problem* as follows. We are given a set of sensor nodes $N = \{n_1, n_2, ..., n_m\}$ and the distance between every pair of m sensors in form of an $m \times m$ matrix $[d_{ij}]$, where $d_{ij} > 0$. Each sensor node n_i has a sensing rate r_i to collect data and a buffer with size b_i to store sensing data. The message ferry visits each sensor node with constant speed s to pick up the sensing data. A complete sequence $n_{i_1}, n_{i_2}, \ldots, n_{i_k}, k \geq m$, is a closed path that visits every sensor at least once. The message ferry routing problem is to find a complete sequence such that the buffer of each sensor does not overflow between two visits. That is, $t_{i_j} \leq \frac{b_{i_j}}{r_{i_j}}$ for every node n_{i_j} where t_{i_j} is the travel time of message ferry between two visits for node n_{i_j} with respect to sequence $n_{i_1}, n_{i_2}, \ldots, n_{i_k}$.

Consider an example of a sensor network with a set of sensor nodes $N = \{n_1, n_2, \ldots, n_{10}\}$, as shown in Fig. 3.1-(a), and the distance between every pair of

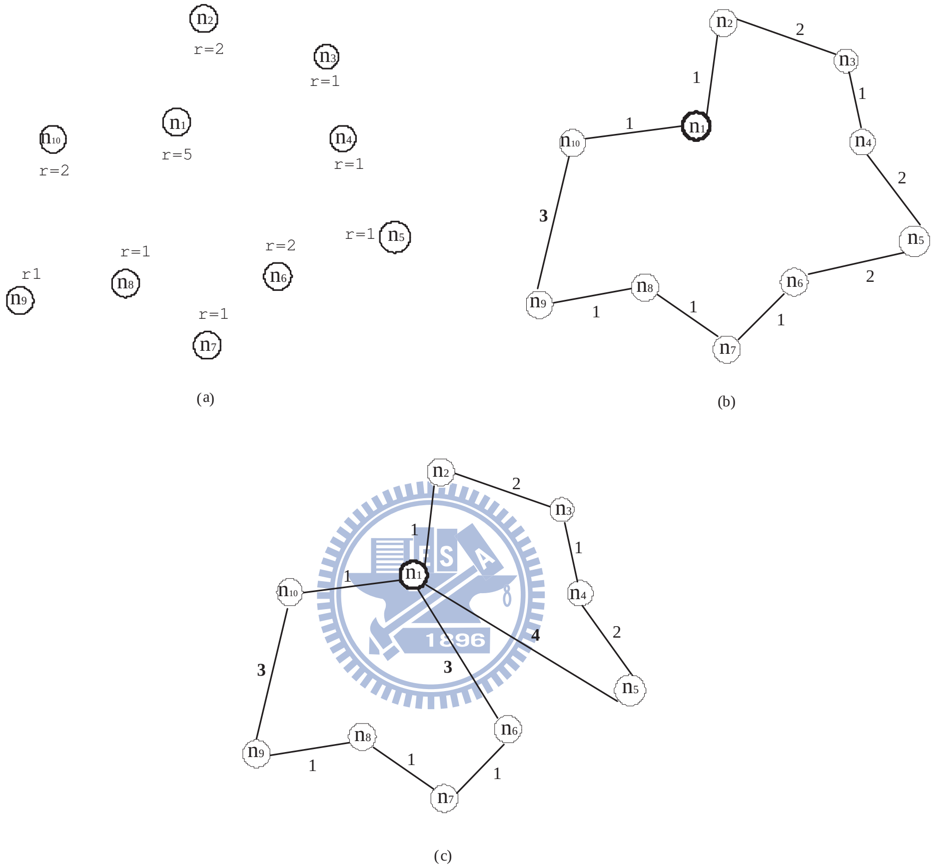


Figure 3.1: (a) Ten sensors in the sensing field. (b) A least cost visit sequence with one critical node (Node n_1). (c) A feasible complete sequence $n_1, n_2, n_3, n_4, n_5, n_1, n_6, n_7, n_8, n_9, n_{10}, n_1$.

10 sensors

$$[d_{ij}] = \begin{bmatrix} - & 1 & 2.3 & 2.5 & 4 & 3 & 3.8 & 3.2 & 3.5 & 1 \\ 1 & - & 2 & 2.7 & 4.5 & 3.9 & 4.7 & 4 & 4.5 & 2 \\ 2.3 & 2 & - & 1 & 2.9 & 3.3 & 4.2 & 4.1 & 5 & 3.5 \\ 2.5 & 2.7 & 1 & - & 2 & 2.3 & 3.3 & 3.5 & 4.5 & 3.5 \\ 4 & 4.5 & 2.9 & 2 & - & 2 & 3 & 3.8 & 5 & 5.1 \\ 3 & 3.9 & 3.3 & 2.3 & 2 & - & 1 & 1.1 & 1.8 & 3.5 \\ 3.8 & 4.7 & 4.2 & 3.3 & 3 & 1 & - & 1 & 2.3 & 3.8 \\ 3.2 & 4 & 4.1 & 3.5 & 3.8 & 1.1 & 1 & - & 1 & 2.9 \\ 3.5 & 4.5 & 5 & 4.5 & 5 & 1.8 & 2.3 & 1 & - & 3 \\ 1 & 2 & 3.5 & 3.5 & 5.1 & 3.5 & 3.8 & 2.9 & 3 & - \end{bmatrix}.$$

The sensing rate $(r_1, \ldots, r_{10}) = (5, 2, 1, 1, 1, 2, 1, 1, 1, 2)$ and the buffer size $b_i = 74, i = 1, \ldots, 10$. Assume that a message ferry with constant speed $s = 1$ to collect the sensing data along the visiting sequence $n_1, n_2, \ldots, n_{10}, n_1$ (see Fig. 3.1-(b)). Then, the travel time t_i of message ferry between two visits for node n_i is

$$t_i = \frac{1+2+1+2+2+1+1+1+3+1}{1} = 15, \qquad i = 1, 2, \ldots, 10,$$

and the amount of data sensed during two visits is $(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}) = (75, 30, 15, 15, 15, 30, 15, 15, 15, 30)$ where $a_i = t_i \times r_i$. Note that the visiting sequence $n_1, n_2, \ldots, n_{10}, n_1$ is infeasible because the amount of data sensed by node n_1 is 75 that is greater than the buffer size 74. We call n_1 a critical node since it is infeasible. Then, we can partition on the critical node n_1 and obtain a new sequence, say $n_1, n_2, n_3, n_4, n_5, n_1, n_6, n_7, n_8, n_9, n_{10}, n_1$. If the message ferry visits the sensors along the sequence $n_1, n_2, n_3, n_4, n_5, n_1, n_6, n_7, n_8, n_9, n_{10}, n_1$ (see Fig. 3.1-(c)), then the travel time t_i of message ferry between two visits for node n_i is

$$t_i = \frac{1+2+1+2+4+3+1+1+1+3+1}{1} = 20, \qquad i = 2, \ldots, 10,$$

except

$$t_1 = \max\{\frac{1+2+1+2+4}{1}, \frac{3+1+1+1+3+1}{1}\} = 10.$$

And, the amount of data sensed during two visits for each node is $(a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}) = (50, 40, 20, 20, 20, 40, 20, 20, 20, 40)$. All a_is are less than 74 and thus the complete sequence $n_1, n_2, n_3, n_4, n_5, n_1, n_6, n_7, n_8, n_9, n_{10}, n_1$ is feasible.

3.3 Solution methods for message ferry routing problem

We propose two Message Ferry Routing Algorithms for data collection in disconnected wireless sensor networks referred as MFRA1 and MFRA2.

3.3.1 MFRA1

The MFRA1 algorithm includes two phases: finding a least distance visit sequence and partitioning complete sequence. Details of the MFRA1 algorithm are illustrated as follows.

Phase 1: Find a least distance visit sequence

We first solve the Traveling Salesman Problem (TSP) by branch-and-cut algorithm [59] to find a least cost tour (i.e., a least distance visit sequence) for a set N of sensor nodes with distance matrix $[d_{ij}]$. Then we check buffer size constraint for each sensor node. If all buffer size constraints are satisfied, a solution is found; Otherwise, the least distance visit sequence is infeasible and go to phase 2.

Phase 2: Partition complete sequence

Phase 2 of MFRA1 recursively executes the following steps: partition sequence, construct TSP sequence and check the feasibility of the visit sequence.

1) Partition sequence

If there is an overflow sensor in the initial visit sequence, MFRA1 fixes the overflow by partitioning the initial visit sequence into some sub-sequences such that the ferry visits the overflow node twice in the resulting sequence. That is, given a least distance visit sequence $n_{i_1}, n_{i_2}, ..., n_{i_k}, n_{i_1}$, if there is a critical node n_{i_1} (an overflow sensor) in this least distance visit sequence $n_{i_1}, n_{i_2}, ..., n_{i_k}, n_{i_1}$. MFRA1 partitions the nodes $\{n_{i_2}, n_{i_3}, ..., n_{i_k}\}$ into two sub-sets and each sub-set includes the critical node n_{i_1} such as $\{\{n_{i_1}, n_{i_2}\}, \{n_{i_1}, n_{i_3}, ..., n_{i_k}\}\}$, $\{\{n_{i_1}, n_{i_3}\}, \{n_{i_1}, n_{i_2}, n_{i_4}, ..., n_{i_k}\}\}$,..., $\{\{n_{i_1}, n_{i_k}\}, \{n_{i_1}, n_{i_2}, ..., n_{i_{k-1}}\}\}$, $\{\{n_{i_1}, n_{i_2}, n_{i_3}\}, \{n_{i_1}, n_{i_4}, n_{i_5}, ..., n_{i_k}\}\}$, $\{\{n_{i_1}, n_{i_2}, n_{i_3}, n_{i_4}\}, \{n_{i_1}, n_{i_5}, n_{i_6}, ..., n_{i_k}\}\}$,...,$\{\{n_{i_1}, n_{i_2}, ..., n_{i_{k-1}}\}, \{n_{i_1}, n_{i_k}\}\}$, and so on.

Similarly, if there is another critical node, says n_{i_3} in a sub-set such as $\{n_{i_1}, n_{i_3}, ..., n_{i_k}\}$, MFRA1 partitions the nodes $\{n_{i_1}, n_{i_3}, ..., n_{i_k}\}$ into two sub-sets. By the way, MFRA1 repeats the partition process as the same as the above step until no other sub-set can be partitioned.

2) Construct TSP sequence

MFRA1 constructs TSP sequence of each sub-set which includes critical node.

3) Check the feasibility of the visit sequence

For each visit sequence $n_{i_1}, n_{i_2}, ..., n_{i_1}$, we compute the travel time t_{i_j} for n_{i_j} by $1 \leq j \leq k-1$. Then, for each node n_{i_j}, we check the feasibility by $t_{i_j} \leq b_{i_j}/r_{i_j}$. If any visit sequence is feasible, the solution is found. Otherwise, repeat phase 2 until no other sub-set can be partitioned. In this case, we can claim that there is no feasible solution.

Consider the example in Fig. 3.2. There is a sensor network with a set of sensor nodes $N = \{n_1, n_2, \ldots, n_5\}$. The sensing rate $(r_1, \ldots, r_5) = (5, 2, 1, 2, 1)$ and the buffer size $(b_1, b_2, b_3, b_4, b_5) = (44, 44, 44, 44, 44)$. The distance between every

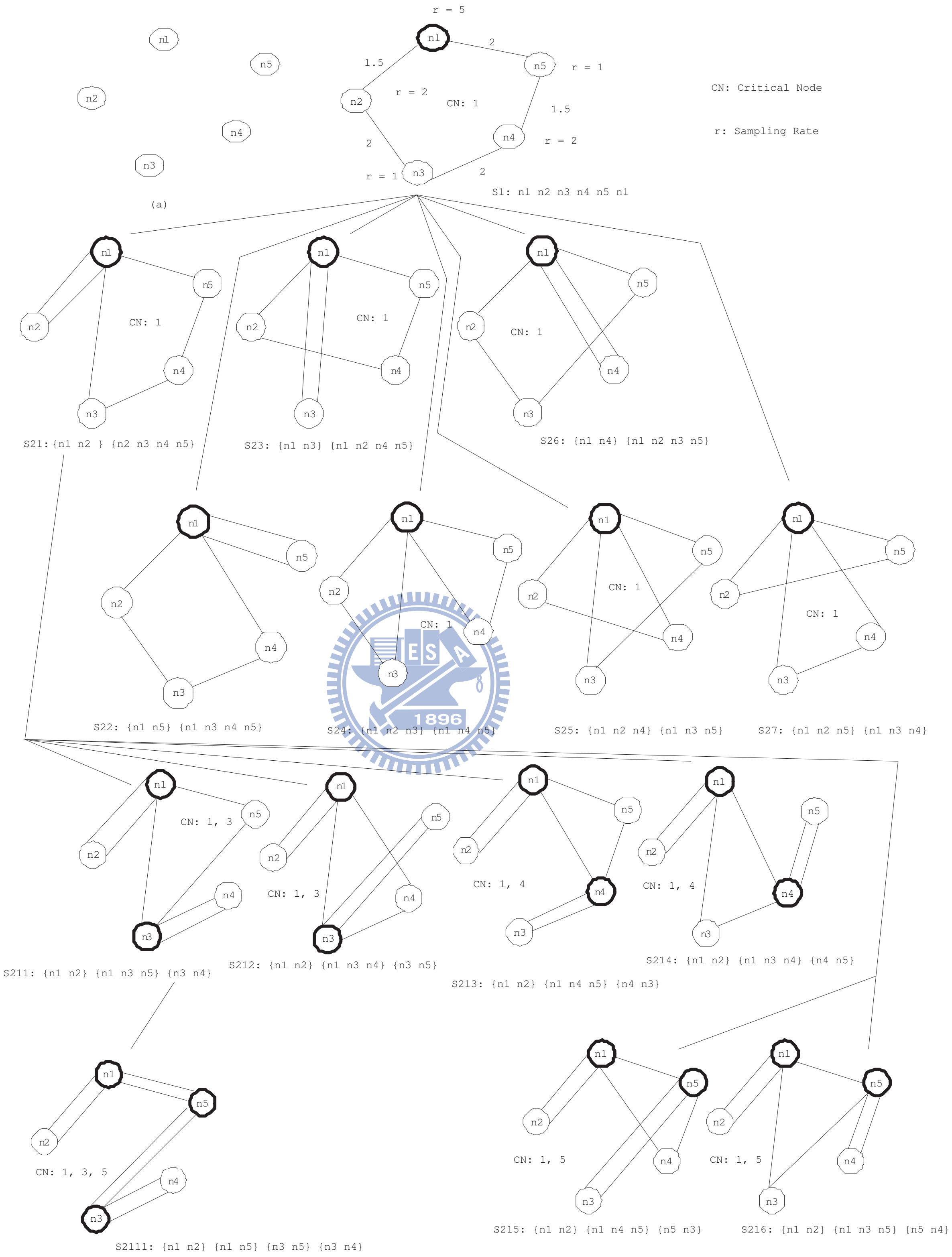


Figure 3.2: An illustrated example of MFRA1.

pair of 5 sensors

$$[d_{ij}] = \begin{bmatrix} 0 & 1.5 & 3.4 & 2.6 & 2 \\ 1.5 & 0 & 2 & 2.3 & 2.5 \\ 3.4 & 2 & 0 & 2 & 3.2 \\ 2.6 & 2.3 & 2 & 0 & 1.5 \\ 2 & 2.5 & 3.2 & 1.5 & 0 \end{bmatrix}.$$

Assume that a message ferry with constant speed $s = 1$ to collect the sensing data. Applying phase 1, we find a least visiting sequence $n_1, n_2, \ldots, n_5, n_1$ and a critical node n_1 (as shown in Fig. 3.2, state S_1). In phase 2, start form critical node n_1 and partition all nodes $\{n_1, n_2, n_3, n_4, n_5\}$ into two sub-sets $\{\{n_1, n_2\}$,$\{n_1, n_3, n_4, n_5\}\}$, $\{\{n_1, n_3\}$,$\{n_1, n_2, n_4, n_5\}\}$,..., $\{\{n_1, n_2, n_5\}$,$\{n_1, n_3, n_4\}\}$ (see states states $S_{21}, S_{22}, ..., S_{27}$ in Fig. 3.3). Then, MFRA1 construct TSP sequence of each sub-set, and check the feasibility of the visit sequence.

After executing steps 1 to 3 of phase 2, if there is not feasible solution, then we choose state S_{21} for further partition. Assume that there is another critical node n_5 in $\{n_1, n_3, n_4, n_5\}$, MFRA1 continues to partition the sequence $\{n_1, n_3, n_4, n_5\}$ (see states $S_{211}, S_{212}, ..., S_{216}$ in Fig. 3.2), construct the TSP sequence, and check the feasibility of the sequence.

After executing above steps, if there is another critical node n_3 in sequence $\{n_1, n_3, n_5\}$ in state S_{211}, MFRA1 continues to partition the sequence $\{n_1, n_3, n_5\}$, construct the TSP sequence, and check the feasibility of the sequence.

Fig. 3.3 is the solution space for this illustrated example. Level 1 of Fig. 3.3 is the initial complete sequence. If all buffer size constraints are satisfied in level 1, then the solution is found. Otherwise, the least distance visit sequence is infeasible and search level 2 states. If all buffer size constraints are satisfied in any state of level 2, then the solution is found. Otherwise, search level 3 states, and so on. MFRA1 will stop after finding a feasible solution or checking all possible sequences. The following Lemma can help to speed up the search procedure.

Lemma 1: It is infeasible if partitioning critical node i leads node j to overflow and then partitioning node j leads critical node i to overflow in complete sequence k, where $1 \leq i$, $j \leq m$, $i \neq j$, and m is the number of nodes.

Proof: Since TSP sequence is the shortest path, no other sequence has a path shorter than the TSP sequence. Therefore, it is infeasible if partitioning critical node i leads node j to overflow and then partitioning node j leads critical node i to overflow in complete sequence k. Q.E.D.

3.3.2 MFRA2

The MFRA1 algorithm can find the solution if the feasible solution exists. However, the MFRA1 is an exhaustive search and the computational time is untraceable. Thus, we propose a heuristic algorithm, called MFRA2 which can find a solution more quickly. Details of the MFRA2 algorithm are illustrated as follows.

Phase 1: Find a least distance visit sequence

Phase 1 of MFRA2 is the same as Phase 1 of MFRA1.

Phase 2: Partition complete sequence

Start from a critical node, says n_{i_1}, partition the initial complete sequence $n_{i_1}, n_{i_2}, ..., n_{i_m}$ into sub-sequences in anti-clockwise direction, and check the feasibility of each sensor node. Note that MFRA2 only generates $m-2$ sequences $n_{i_1}n_{i_2}n_{i_1}n_{i_3}...n_{i_m}n_{i_1}, n_{i_1}n_{i_2}n_{i_3}n_{i_1}n_{i_4}...n_{i_m}n_{i_1}, ..., n_{i_1}n_{i_2}...n_{i_{m-1}}n_{i_1}n_{i_m}n_{i_1}$ in the level one and check their feasibility. As shown in Fig. 3.5, a complete sequence $n_1n_2n_3n_4n_5n_1$ with one critical node n_1 is partitioned into 3 sequences: $n_1n_2n_1n_3n_4n_5n_1$,

Figure 3.3: The solution space of MFRA1 for the example.

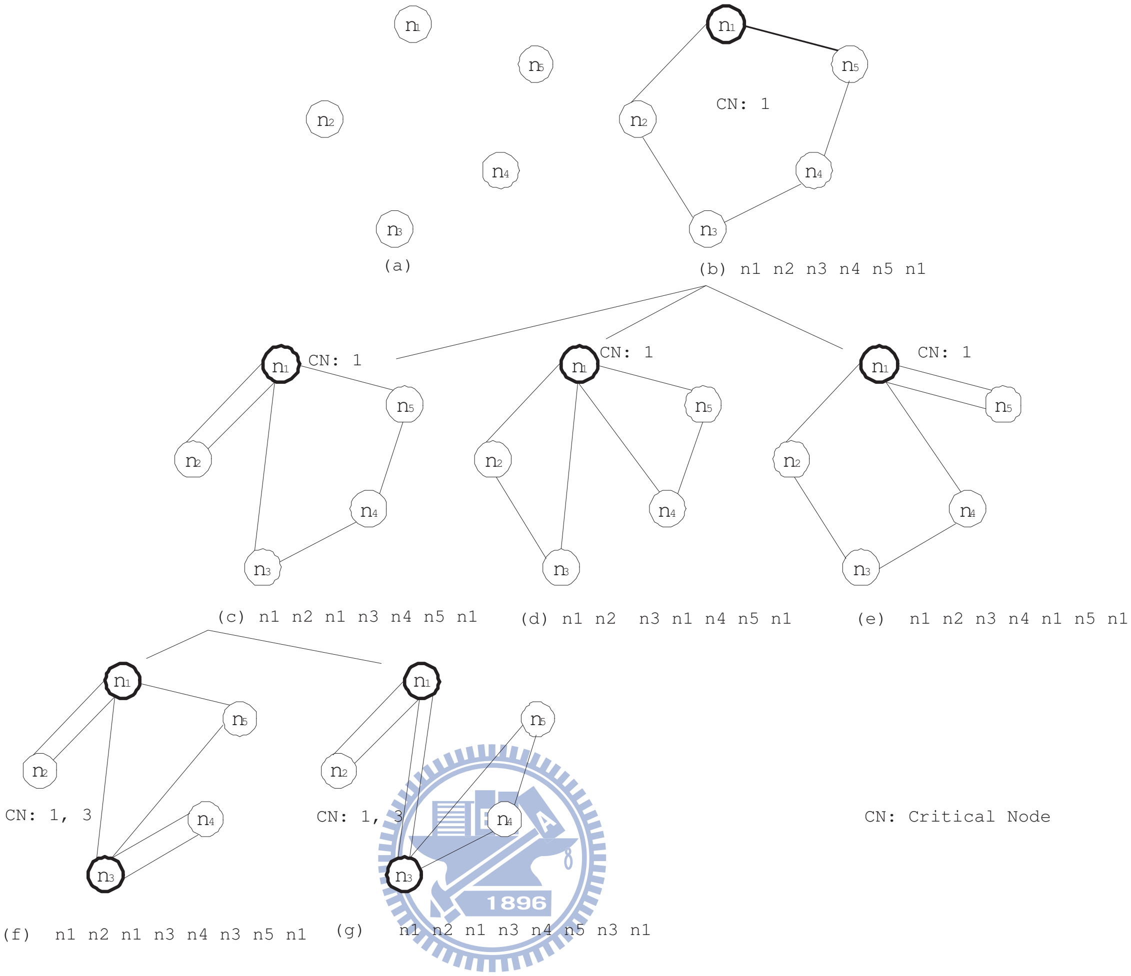


Figure 3.4: An illustrated example of MFRA2.

$n_1n_2n_3n_1n_4n_5n_1$ and $n_1n_2n_3n_4n_1n_5n_1$.

If all of the 3 sequences are infeasible, we choose one of them for further partition. For example, the sequence $n_1n_2n_1n_3n_4n_5n_1$ with critical node n_3 in sub-sequence $n_1n_3n_4n_5n_1$ is partitioned into two sequences $n_1n_2n_1n_3n_4n_3n_5n_1$ and $n_1n_2n_1n_3n_4n_5n_3n_1$ (see Fig. 3.4).

An example of the solution space of MFRA2 is shown in Fig. 3.5. Level 1 of Fig. 3.5 is the initial complete sequence. If all buffer size constraints are satisfied in

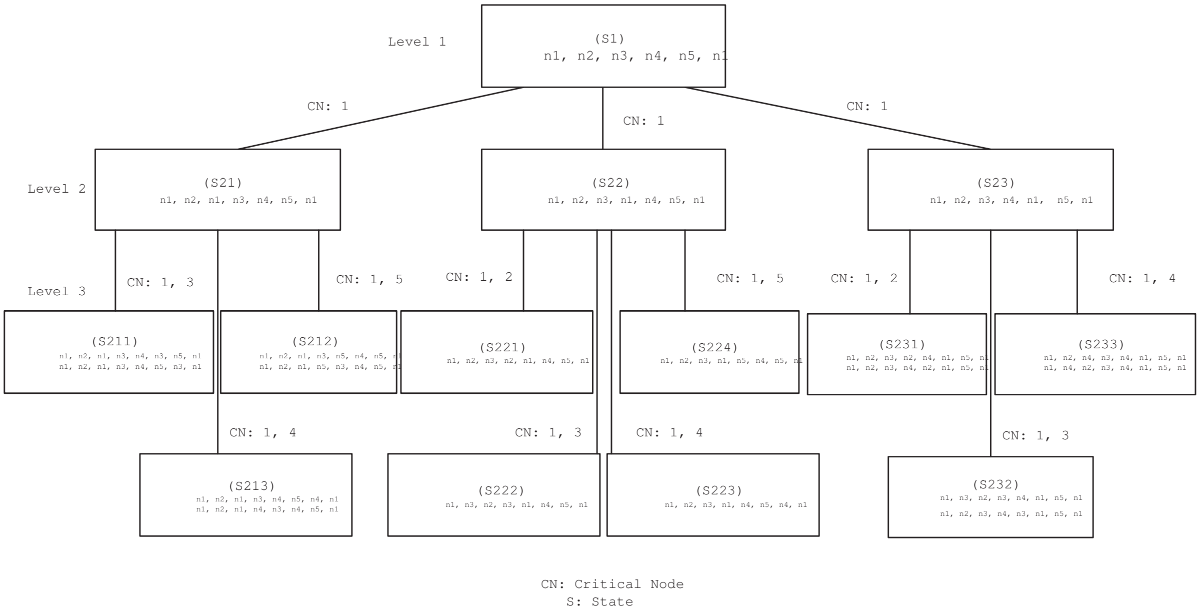


Figure 3.5: The solution space of MFRA2.

level 1, the solution is found. Otherwise, the initial complete sequence is infeasible and search level 2 states. If all buffer size constraints are satisfied in any state of level 2, the solution is found. Otherwise, search level 3 states, and so on. MFRA2 will stop after finding a feasible solution or checking all generated sequences.

3.4 Simulation results

3.4.1 Performance metrics and environment setup

This section presents the performance analysis of MFRA1 and MFRA2 algorithms. The environment setup of simulation is described as follows. There are different kinds of sensors such as surveillance sensors and data sensors in a two dimensional sensing area. The surveillance sensor has a high sampling rate to capture video message. Data sensor has a low sampling rate to collect temperature or noise data.

We study the following performance metrics.

1) **The travel time**

The travel time of message ferry is defined as the time that message ferry goes through every node in the complete sequence.

2) **Number of sequences checked**

The number of sequences checked is the number of sequences generated and feasibilities checked by the MFRA1 (or MFRA2) algorithm.

3) **The amount of data loss**

For a complete sequence (found by MFRA1, MFRA2, Greedy algorithm, Nearest Neighbor, Lin Kernighan [60] or PBS [58]), we calculate the amount of data loss if the message ferry collects the data along this complete sequence. The greedy algorithm finds a Hamiltonian cycle as a complete sequence greedily. The nearest neighbor algorithm constructs a Hamiltonian cycle as a complete sequence by starting at a node n_0, choosing the nearest neighbor node as next node and so on, and finally returning to n_0.

3.4.2 Numerical results

1) **The travel time**

There are two set of sensors, ($N = 5$ and $N = 10$), in a 10 $km \times 10$ km two-dimensional sensing field. The speed of the message ferry is 36 km/hr. The travel time vs. the number of critical nodes for MFRA1 and MFRA2 is shown in Fig. 3.6. Overall, the travel time increases when the number of critical nodes increases. This is because MFRA1 and MFRA2 continue to partition a sequence and the number of nodes in the resulting sequence increases.

2) **Number of sequences checked**

As shown in Fig. 3.7, the number of checked sequences of MFRA1 is much larger than MFRA2. This is because MFRA1 checks all sequences to find feasible solutions. MFRA1 algorithm can find the solution if the feasible solution exists. However, the MFRA1 is an exhaustive search and it may consume lots of computation time. The MFRA2 is a heuristic algorithm and it can find a solution more quickly. But, the MFRA2 may not find the solution whenever the solution exists. This is because MFRA2 does not check all sequences. Fig. 3.8 shows the ratio of the solution found for MFRA1 and MFRA2. As shows in Fig. 3.8, the ratio of the solutions found of MFRA1 is nearly the same as MFRA2. Therefore, MFRA2 is an efficient algorithm.

3) **The amount of data loss**

As shown in Fig. 3.9, the amount of data loss of MFRA1 and MFRA2 algorithms is much smaller than Greedy, Nearest Neighbor and Lin Kernighan algorithms. Greedy, Nearest Neighbor and Lin Kernighan algorithms will lose data when a sensor overflows. However, MFRA1 and MFRA2 work well without losing data. Fig. 3.10 shows the amount of data loss for different buffer sizes with 3 critical nodes. Fig. 3.11 shows the amount of data loss for different buffer sizes with 2 critical nodes. MFRA1 and MFRA2 perform better than other schemes.

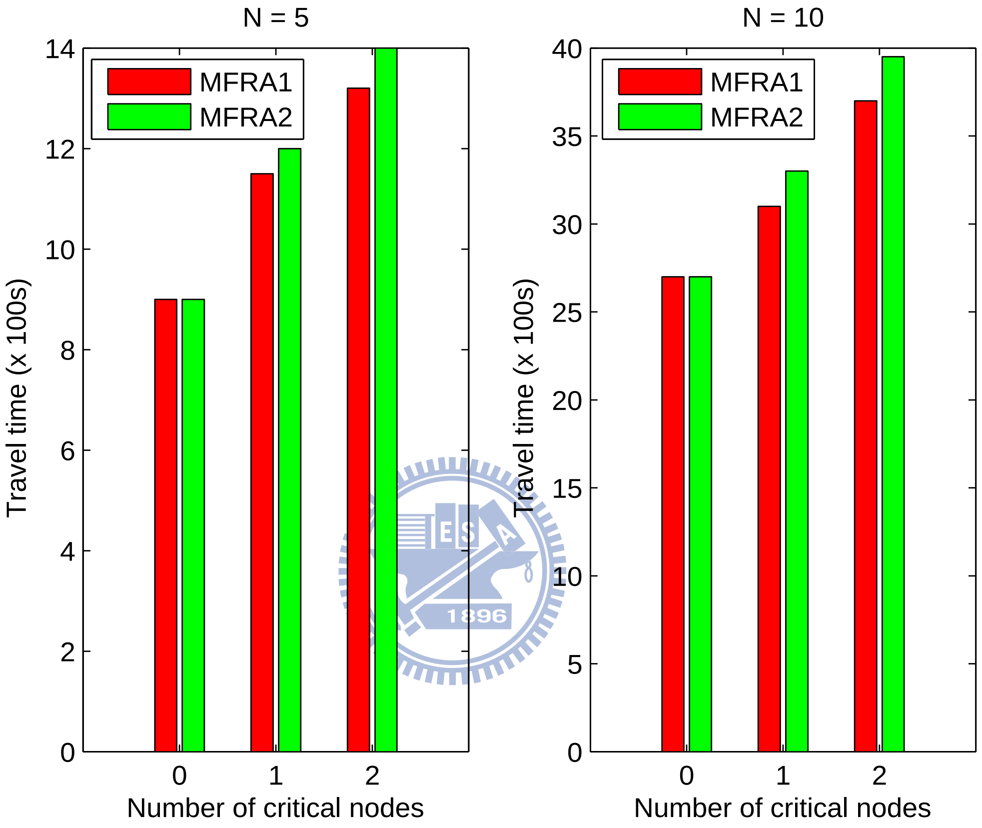


Figure 3.6: The travel time vs. the number of critical nodes for MFRA1 (MFRA2) with 5 and 10 sensors.

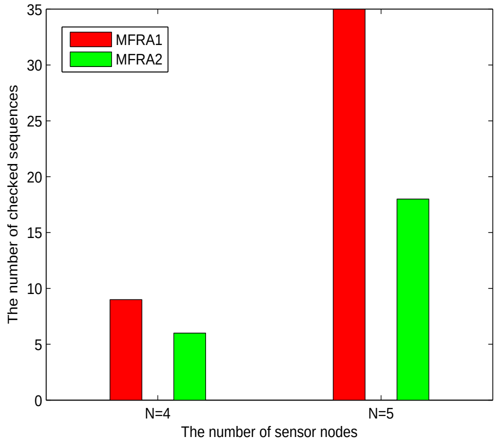


Figure 3.7: The number of checked sequences of MFRA1 and MFRA2 algorithms.

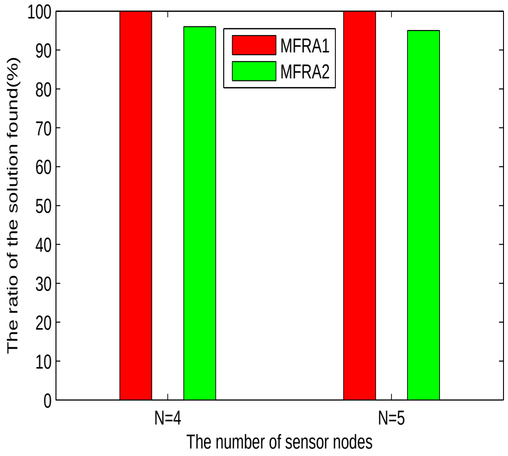


Figure 3.8: The number of feasible solutions of MFRA1 and MFRA2 algorithms.

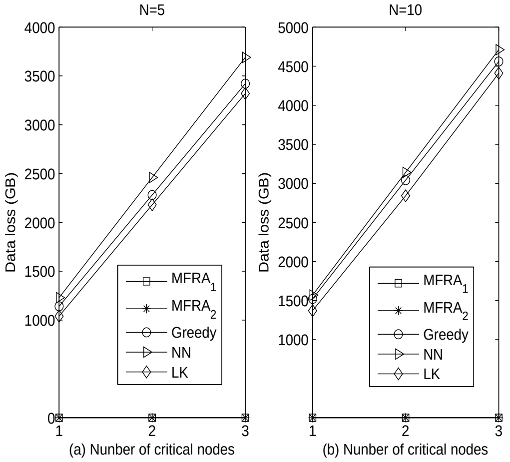


Figure 3.9: The amount of data loss for different methods.

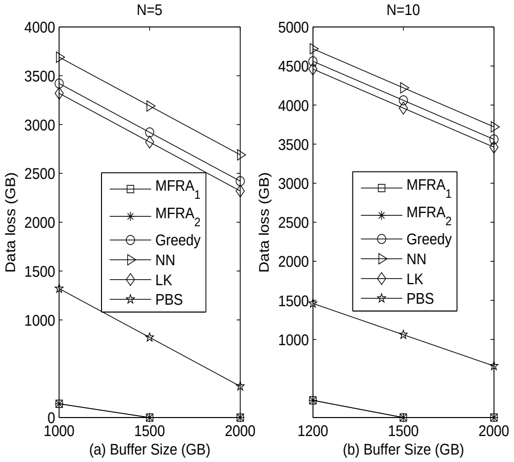


Figure 3.10: The amount of data loss for different buffer sizes with 3 critical nodes.

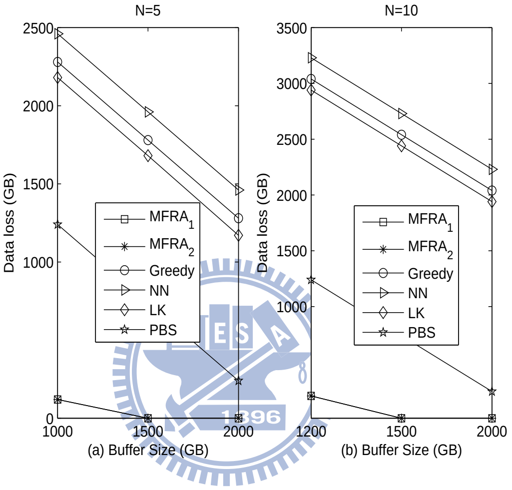


Figure 3.11: The amount of data loss for different buffer sizes with 2 critical nodes.

Chapter 4

Relay placement problem for disconnected WSNs

In this chapter, we first describe related works of relay placement problem. Then, we illustrate problem formulation and our solution methods. Finally, we present simulation results of our scheme.

4.1 Related works of relay placement problem

In a finite domain, the connectivity of a random network depends only on the probability distribution of the critical transmission range. Many studies have tried to find efficient algorithms for determining the critical transmission range for connectivity [61-63]. The asymptotic distribution of the critical transmission radius for k-connectivity is derived in [61]. This study proved that the critical transmission range in the unit-area square is $r_n = \sqrt{\frac{\log n+(2K-1)\log\log n+\xi}{\pi n}}$ where n is the number of network nodes and ξ is a constant. The critical transmitting range for connectivity in mobile ad hoc networks is proved in [62]. The author showed the critical transmission range (CTR) for a mobility model M is $r_M = \sqrt{\frac{\log n}{\pi n}}$ for some constants $C \geq 1$ where n is the number of nodes in the network. The mobility model M is assumed to be obstacle free, and nodes are allowed to move only within a certain bounded area. In addition, many studies have focused on maintaining sensing coverage and connectivity in wireless sensor networks [64-67]. The transmission range R_t being at least twice that of the sensing range R_s is the sufficient condition to ensure that complete coverage preservation implies connectivity among active nodes [64]. Another study [65] enhanced the work in [64] to prove that the sufficient condition for complete coverage implies that connectivity is $R_t = 2R_s$. In [67], a coverage configuration protocol is proposed to achieve guaranteed degrees of coverage and connectivity. This work provided different degrees of coverage requested by applications. To measure the coverage, the work divides the sensing area into 1m × 1m patches. The coverage degree of a patch is approximated by measuring the number of active nodes that cover the center of the patch.

Note that the above studies [61-67] do not discuss how to place relay sensors to improve connectivity for a disconnected ad hoc network. Thus, there are many papers proposed for finding the optimal location to place the additional nodes to achieve network connectivity [68-72]. This can be reduced to a minimal Steiner tree problem. In [68], a relay sensor placement algorithm to maintain connectivity is proposed. The authors formulated this problem into a network optimization problem, named the Steiner Minimum Tree with Minimum Number of Steiner Points (SMT-MSP). This study restricts the transmission power of each sensor to a small value and adds relay sensors to guarantee connectivity. Simulation results show that their method can achieve better performance in terms of total consumed power and maximum degree, especially for sparse network topology. However, their algorithm has a time complexity in $O(N^3)$. Some heuristic algorithms for the bounded edge-length Steiner tree problem with a good approximate ratio are proposed in [69-72]. Nevertheless, these heuristic algorithms do not consider the heterogeneous transmission ranges of terminal nodes and relay nodes.

Many studies have focused on finding efficient heuristic algorithms to solve the minimal additional nodes placing problem and to prolong network lifetime [73-75]. A heuristic algorithm for the energy preserving problem is proposed in [73]. This algorithm transforms the mixed-integer nonlinear problem into a linear programming problem. This study provides additional energy on the existing nodes and deploys relay nodes into the network to prolong network lifetime. In [74], three heuristic algorithms are proposed for achieving the connectivity of a randomly deployed ad hoc wireless network. This work connects the network with a minimum number of additional nodes and maximizes utility from a given number of additional nodes for the disconnected network. The time complexity of the greedy algorithms is $O(N^2)$ in a two dimensional space, where N is the number of terminal nodes.

4.2 Problem formulation and network model

Consider a wireless sensor network. We assume that the sensing area is in a two-dimensional space which is a bounded convex subset R of the Euclidean space. In this sensing area, the initially deployed sensors, referred to as terminal nodes, have been placed and a set of relay sensors is available to be added for connectivity. All terminal nodes and relay nodes are location aware so that the location information can be collected. The set of n terminal nodes is denoted as $N_t = \{n_{t1}, n_{t2}, ..., n_{tn}\}$. The transmission range of each terminal node is adjustable. Initially, the terminal nodes can adjust their transmission range to convey their location information to the base station, then limit their transmission range in a bounded value Rt for energy efficiency. The set of locations of n terminal nodes is denoted as $L_t = \{p_i \in R | i = 1, ..., n\}$. The set of the initial network topology is in the form of an undirected graph denoted as $G(N_t, E(L_t, R_t))$ where $E(L_t, R_t) = \{(l_i, l_j) | l_i, l_j \in L_t, i \neq j, \parallel l_i - l_j \parallel \leq R_t\}$. In order to construct the connected communication graph, we can add the relay sensors to connect the initial separated sub-graphs. A solution is a set of locations to place relay sensors, $L_r = \{q_i \in R | i = 1, ..., r\}$. The set of m relay sensors is denoted as $N_r = \{n_{r1}, n_{r2}, ..., n_{rm}\}$. We formulate our problem as follows: A randomly deployed sensor network with a $nR_t \times nR_t$ sensing area in two dimensional space. Given N_t and R_t, find the L_r for minimum relay sensor set N_r to make the graph $G(N_t \cup N_r, E(L_t \cup L_r, R_t))$ connected.

4.3 Solution methods for relay placement problem

The ERSPA algorithm includes the following two phases: 1) Construct Delaunay and find the initial graph $G(N_t, E(L_t, R_t))$; 2) Add relay nodes. The details of the algorithm are given as follows.

Phase 1: Construct Delaunay and find the initial graph $G(N_t, E(L_t, R_t))$ Construct the Delaunay by using terminal nodes [76]. The construction of Delaunay is illustrated as follows. Let S be a set of points in a two dimensional space. The Voronoi diagram of S, denoted as $Vol(S)$ which is decomposed into Voronoi cells $\{V_a : a \in S\}$ is defined as equation (4.1).

$$V_a = \{x \in R^2 : |x - a| \leq |x - b| \forall b \in S\} \tag{4.1}$$

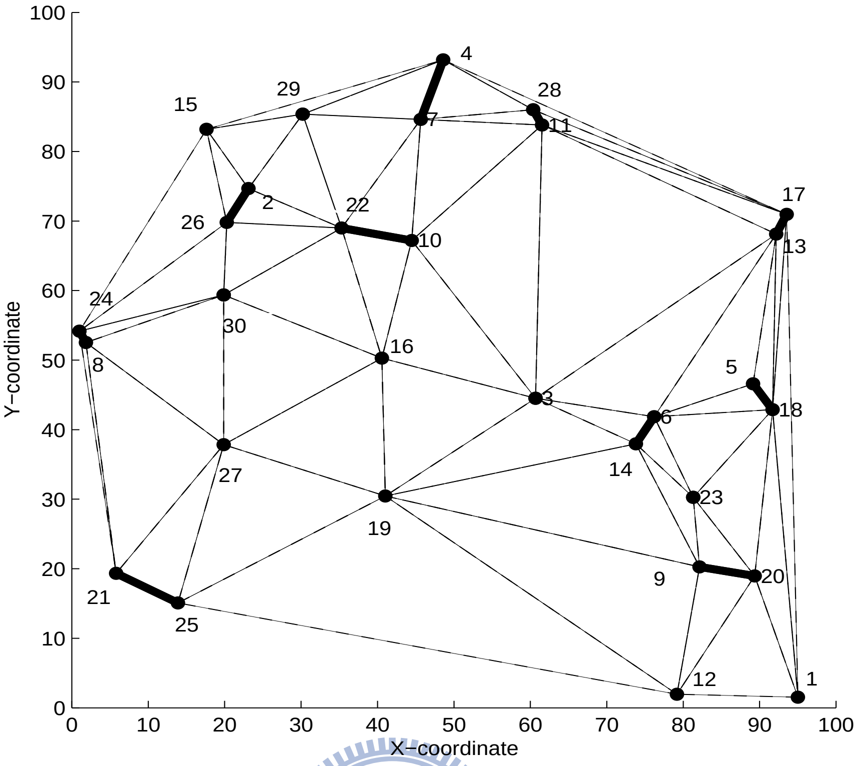


Figure 4.1: An example of constructing Delaunay using 30 terminal nodes. The thin solid lines represent the edges of the Delaunay triangulation that are not connected. The initial disconnected terminal nodes are indicated by the '●'-sign. The initial connected sub-graphs are indicated by the thick solid lines.

The dual of the Voronoi diagram is the Delaunay triangulation $Del(S)$. $Del(S)$ is geometrically realized as a triangulation of the convex hull of S (see Figure 4.1). After constructing Delaunay, we calculate the length of the three edges of each triangle. If the length of the edge is no more than transmission range R_t, then connect it. As shown in Figure 4.1, the initial connected sub-graphs in phase 1 are indicated by the thick solid line.

Phase 2: Add relay nodes

We first illustrate the following lemmas about adding relay nodes.

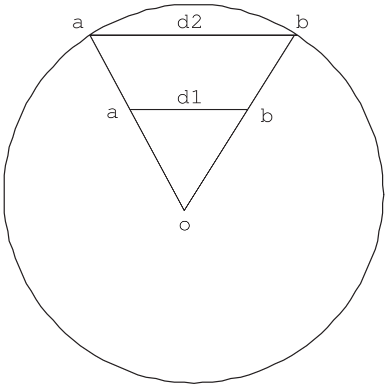


Figure 4.2: An example of lemma 1.

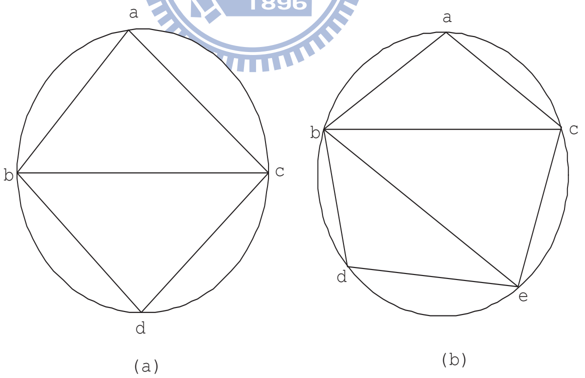


Figure 4.3: (a) An example of four nodes covered by a minimum circle of radius R. (b) An example of five nodes covered by a minimum circle of radius R.

Lemma 1: One relay node can connect at most five nodes that belong to different sub-graphs [71].

Proof: We prove this lemma by contradiction. As shown in Figure 5.2, if six nodes can be covered by a minimum circle of radius R where R is no more than the transmission range R_t, we draw a circle of radius R_t to cover the six nodes centered at o, then draw six lines from each node to the center of the circle. Assume that the shortest length of the adjacent nodes pair (a, b) is d_1. We extend the length of line (a, b) from d_1 to d_2. That is, nodes a and b are on the circle. If d_2 is equal to R_t, each edge of triangle (a, o, b) has the same length. Angle (a, o, b) is equal to 60°. If d_2 is larger than R_t, then the angle (a, o, b) is larger than 60°. The sum of all six angles is larger than 360°. It is a contradiction. Therefore, one relay node can connect at most five nodes that belong to different sub-graphs.

Lemma 2: Four nodes a, b, d, c can be covered by a minimum circle of radius R if and only if all the four triangles (a, b, d), (b, d, c), (d, c, a), (c, a, b) can be covered by a minimum circle of radius R.

Proof: As shown in Figure 4.3-(a), triangles (a, b, c) and (b, d, c) form a quadrilateral (a, b, d, c). From lemma 8 in [61], four nodes a, b, d, c can be covered by a disc of radius R if all the four triangles (a, b, d), (b, d, c), (d, c, a) and (c, a, b) can be covered by a minimum circle of radius R.

Lemma 3: Five nodes a, b, d, e, c can be covered by a minimum circle of radius R if and only if all the four quadrilaterals (a, b, d, e), (b, d, e, c), (d, e, c, a), (e, c, a, b) can be covered by a minimum circle of radius R.

Proof: As shown in Figure 4.3-(b), each of the three adjacent triangles (c, a, b), (c, b, e) and (b, e, d) can be covered by a minimum circle of radius R. We prove this lemma by contradiction. If one of the four quadrilaterals (a, b, d, e), (b, d, e, c), (d, e, c, a) and (e, c, a, b) can not be covered by a minimum circle of radius R, say (e, c, a, b). From lemma 2, there exists at least one triangle of the four triangles (c, a, b), (a, b, e), (b, e, c), (e, c, a) that cannot be covered by a minimum circle of radius R. Assume that the triangle is (c, a, b). This contradicts our assumption that triangle (c, a, b) can be covered by a minimum circle of radius R. Therefore, five nodes a, b, d, e, and c can be covered by a minimum circle of radius R if and only if all four quadrilaterals (a, b, d, e), (b, d, e, c), (d, e, c, a), (e, c, a, b) can be covered by a minimum circle of radius R.

After phase 1, we divide the Delaunay triangles into three types. In type 1, the length of all edges of the triangle is larger than R_t and smaller than $2R_t$. In type 2, the longest edge of the triangle is at most $4R_t$, while the shortest edge is larger than R_t and at most $2R_t$. The properties of triangles different from type 1 and type 2 are defined as type 3. For the triangles of type 1, we place the relay nodes as per the following steps. First, we place one relay node to connect five nodes that are formed by three adjacent triangles. Second, we place one relay node to connect four nodes that are formed by two adjacent triangles. Third, we place one relay node to connect three nodes of one triangle. For the triangles of type 2, we try to place two relay nodes to connect three nodes of one triangle. For the triangles of type 3, we add relay nodes to connect the nearest disconnected nodes pair along the edge of the triangle.

The details of adding relay nodes in the ERSPA algorithm are illustrated as follows.

1) Find the initial sub-graphs $G(N_t, E(L_t, R_t))$, the maximal connected sub-graph,

and the type 1 and type 2 triangles.

2) Find the point where adding one relay node to connect the nodes in type 1 triangles yields the maximal increase in the performance metric. The performance metric is the number of relay sensor. Add the relay node to N_r. Maintain the new sub-graphs, and the maximal connected sub-graph. Repeat this as long as new triangles can be connected.

3) Find the point to add one relay node to connect two different sub-graphs and add the relay node to N_r.

4) Find the point to add two relay nodes to connect three different sub-graphs and add the relay nodes to N_r.

5) If the graph $G(N_t \cup N_r, E(L_t \cup L_r, R_t))$ is not yet connected, add relay nodes to connect the nearest disconnected nodes pair along the edge of the triangle.

In step 2 of phase 2, two adjacent triangles can form a quadrilateral. For example, as shown in Figure 4.1, triangle (2, 29, 15) and triangle (2, 29, 22) form a quadrilateral (2, 29, 15, 22). From lemma 2, if the circumradius of each triangle is at most Rt, then we can place one relay node in the center of the minimal enclosing circle that covers the four points of the quadrilateral. We select the triangles that belong to the three different sub-graphs. The triangles inside the convex hull of the initial connected sub-graphs are not required to be searched. If the circumradius of the selected triangle is at most R_t, then add a node on the circumcenter of the triangle. The circumcenter is the intersection of the perpendicular bisectors of the sides of the triangle. The circumradius of a triangle is the radius of the circumscribed circle. The circumscribed circle of a triangle is the circle that passes through the three vertices of the triangle. For example, as shown in Figure 4.4, R_1 is the circumcenter of the triangle (30, 26, 22). If the distance from R_1 to each

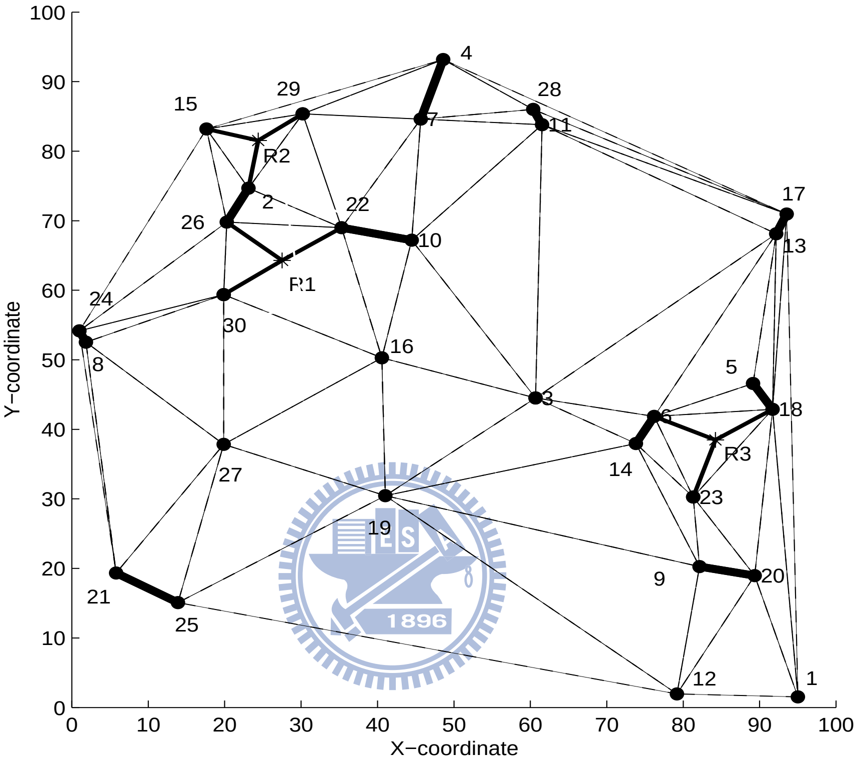


Figure 4.4: An example of adding relay sensors after step 2 of phase 2 for the ERSPA algorithm. Placing one relay node to connect three nodes is indicated by the circle centered at R_1, R_2 and R_3. The relay sensors are indicated by the $'*'$-sign.

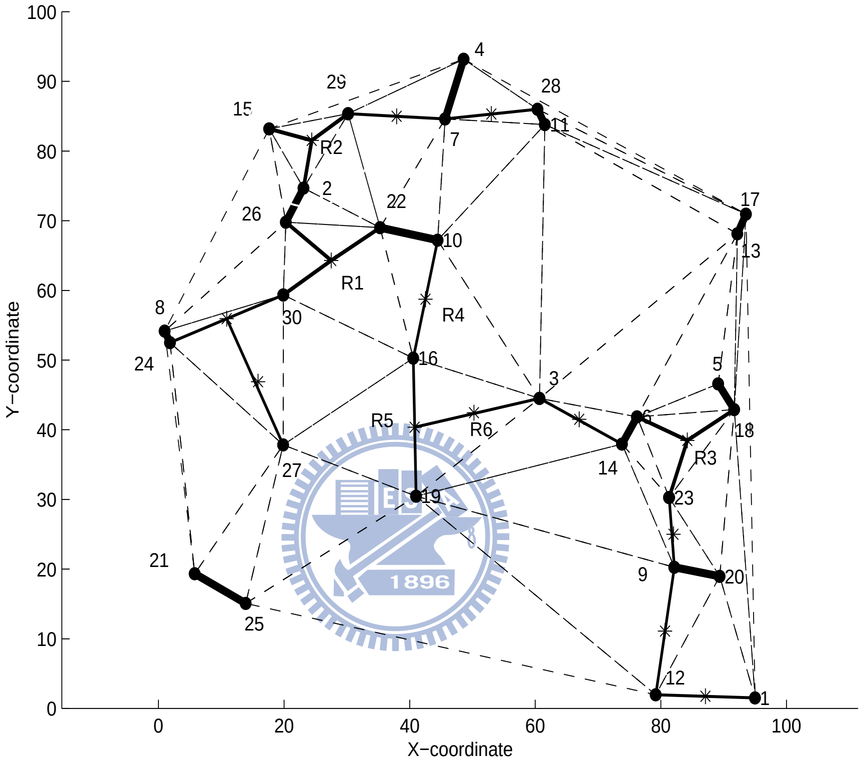


Figure 4.5: The connected sub-graphs after steps 3 and 4 of phase 2 in the ERSPA algorithm.

apex of the triangle is not larger than transmission range R_t, then add a relay node $R1$ on the circumcenter. The relay sensors are indicated by the $'*'$-sign. The $'\bullet'$-sign represents the initial disconnected terminal nodes, and the initial connected sub-graphs are represented by the thick solid line. After adding a relay node, the edge becomes connected, which is indicated by a thin solid line. The performance metric of step 2 of ERSPA is the number of relay sensor.

After step 2, we place one relay node to connect two sub-graphs. For example, as shown in Figure 4.5, we place one relay node R_4 to connect nodes 10 and 16. After step 3, we place two relay nodes to connect three different sub-graphs. For example, as shown in Figure 4.5, we place one relay node R_6 to connect nodes R_5 and node 3. After step 4, if the graph $G(N_t \cup N_r, E(L_t \cup L_r, R_t))$ is not yet connected, then place K relay nodes to connect two sub-graphs. Step 5 exists here to handle the case of type 3 triangle. The number of required relay sensors to add into the edge of the nearest pair of nodes is illustrated as equation (4.2).

$$K \times R_t \leq d(u, v) \leq (K + 1) \times R_t, K = 3, ..., n \tag{4.2}$$

where K is the number of relay nodes. $d(u, v)$ is the distance between nodes u and v. For example, as shown in Figure 4.6, we place two relay nodes to connect nodes 21 and 27. Repeat phase 2 until the complete communication graph is connected (see Figure 4.6).

In phase 1, constructing Delaunay takes $O(N \log N)$ times. Steps 1 - 2 of phase 2 cost $O(N)$ in plane and $O(N^2)$ in three-dimensional sensing space. Steps 3 - 5 of phase 2 cost $O(N)$. Therefore, phase 2 requires $O(N)$ times to add relay sensors in two-dimensional sensing space. The total time complexity of the ERSPA algorithm is $O(N \log N)$. It is feasible in a two-dimensional space.

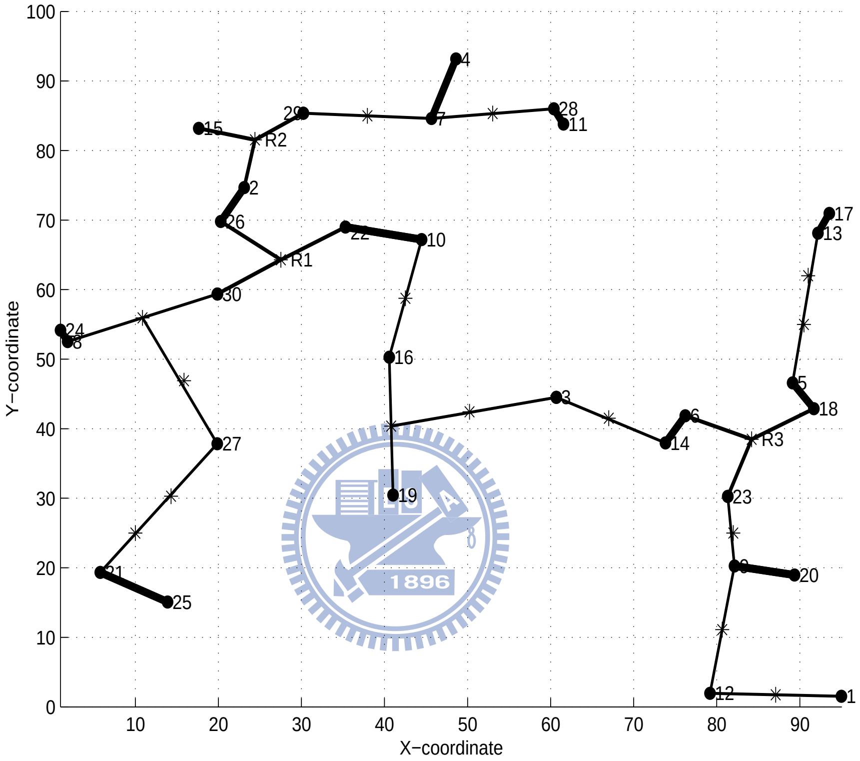


Figure 4.6: The final connected communication graph. The transmission range is 10 percent of the side of the square sensing field.

4.4 Simulation results

4.4.1 Performance metrics and environment setup

This section presents the performance analysis of the ERSPA algorithm. The metrics for performance are given as follows.

1) Average number of relay sensors: The average number of relay sensors is defined as the required minimal average number of additional sensors to make the network connected.

2) Time complexity: Time complexity is defined as the time to run the algorithm.

The environment setup of the simulation is described as follows. There are different numbers of terminal sensors randomly deployed in a 100×100 two-dimensional sensing area. The maximum transmission range is 10 percent of the side of the square sensing field. The network can convey location information of terminal nodes to the base station, then limit the transmission range to a bounded value Rt for energy efficiency.

4.4.2 Numerical results

Comparisons of the two performance metrics were made for three schemes: the ERSPA algorithm, the Minimum Spanning Tree (MST) algorithm and the Greedy algorithm [74]. The MST algorithm has two steps. First, it generates a minimum spanning tree to connect the terminal nodes. Then, it places the relay nodes on the edges of the minimal spanning tree that are longer than the transmission range R_t. The MST algorithm takes $O(N \log N)$ times.

The performance metrics include the total number of relay sensors and the time complexity. The details are illustrated as follows.

1) Average number of relay sensors: Figure 4.7 shows that the average number of

relay nodes of ERSPA is smaller than that of the minimum spanning tree algorithm and the greedy algorithm with different number of terminal nodes. Figure 4.8 shows that the average number of relay nodes of ERSPA is smaller than the minimum spanning tree algorithm and the greedy algorithm when N_t is 30 and R_t is different. Figure 4.9 shows that the average number of ERSPA relay nodes is smaller than that of the minimum spanning tree and the greedy algorithms under different terminal nodes when R_t is 10 percent of the side of the square sensing field. Figure 4.10 shows that the average number of relay nodes of ERSPA is also smaller than that of the minimum spanning tree and the greedy algorithms when R_t is 12 percent of the side of the square sensing field. The average number of relay sensors in MST is approximately two times that of the ERSPA algorithm. This is because ERSPA finds the optimal locations to place relay sensors and connects the maximal number of disconnected sub-graphs.

2) Time complexity: The time complexity of ERSPA algorithm is illustrated as follows. Phase 1 costs $O(N \log N)$ times to construct Delaunay triangulations where N is the number of terminal nodes. Steps 1 - 2 of phase 2 cost $O(N)$ in plane and $O(N^2)$ in three-dimensional sensing space. Steps 3 - 5 of phase 2 cost $O(N)$. Therefore, phase 2 requires $O(N)$ times to add relay sensors in two-dimensional sensing space. The total time complexity of the ERSPA algorithm is $O(N \log N)$. The greedy algorithm takes $O(N^2)$ times. The minimum spanning tree takes $O(N \log N)$ times. The time complexity of ERSPA is feasible in a two-dimensional space.

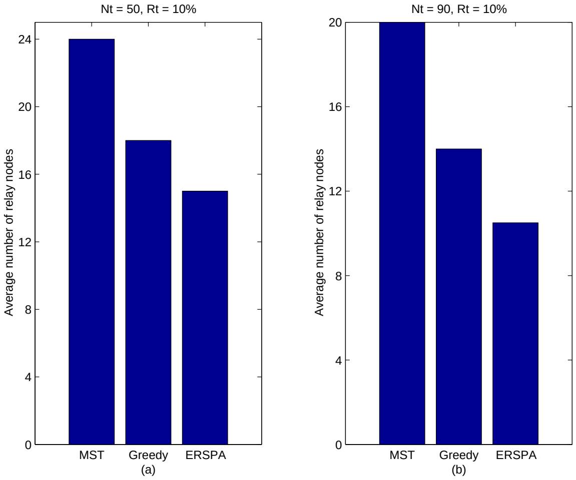


Figure 4.7: (a) The average number of relay nodes for connectivity with 50 terminal nodes (b) The average number of relay nodes for connectivity with 90 terminal nodes. The transmission range is 10 percent of the side of the square sensing field.

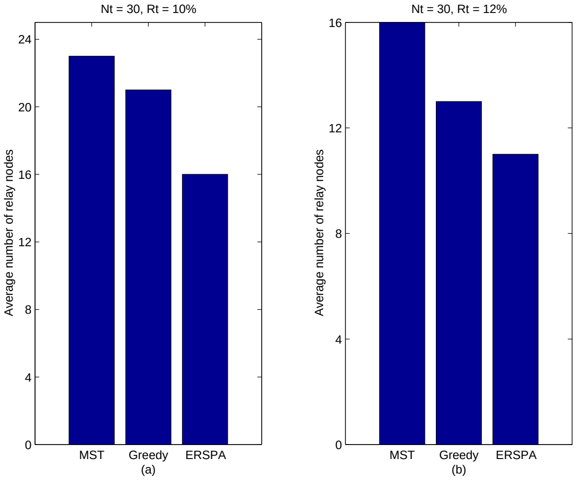


Figure 4.8: (a) The average number of relay nodes for connectivity with 30 terminal nodes. The transmission range is 10 percent of the side of the square sensing field. (b) The average number of relay nodes for connectivity with 30 terminal nodes. The transmission range is 12 percent of the side of the square sensing field.

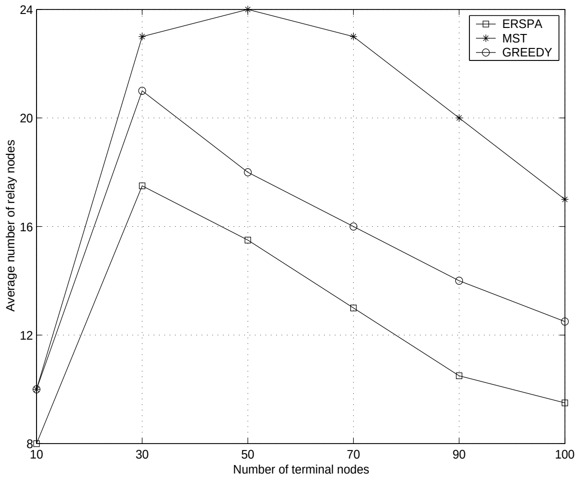


Figure 4.9: The average number of relay nodes for connectivity under different terminal nodes. The transmission range is 10 percent of the side of the square sensing field.

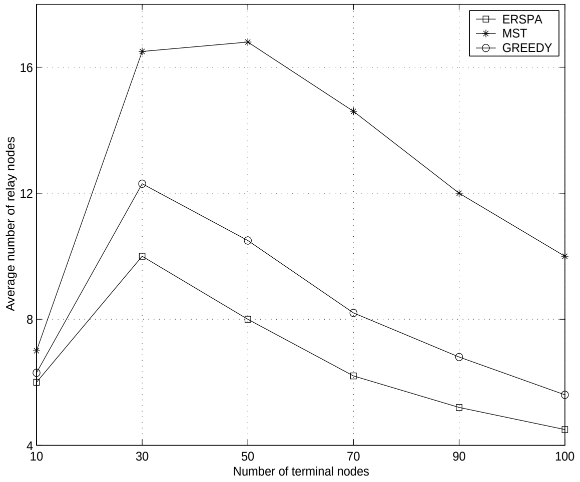


Figure 4.10: The average number of relay nodes for connectivity under different terminal nodes. The transmission range is 12 percent of the side of the square sensing field.

Table 4.1: Time complexity of different schemes.

Schemes	Time complexity
MST	$O(NlogN)$
Greedy	$O(N^2)$
ERSPA	$O(N \log N)$

Chapter 5

Conclusions and future works

In this dissertation, we introduced recent developments in the field of wireless sensor networks and investigated several efficient algorithms for WSNs. We designed several efficient algorithms, including two kinds of energy-aware cluster-based routing algorithms, two kinds of message ferry routing algorithms, and a relay sensor placing algorithm.

For energy-aware routing problem, we proposed two schemes, namely ECRA and ECRA-2T. The ECRA and ECRA-2T rotates intra-cluster-heads to balance the load to all nodes in the sensor networks. The CVT in clustering phase of our scheme is the key different from previous schemes. The CVT can achieve a better performance than other previous schemes. The numerical results also prove that ECRA-2T balances loads better and saves more energy than other schemes. For message ferry routing problem, we proposed two schemes, namely MFRA1 and MFRA2. MFRA1 and MFRA2 are designed for data collection in buffer-limited and partitioned wireless sensor networks. The partition phase of our scheme is the key difference compared to the previous schemes. It can achieve a better performance than other previous schemes. For relay placement problem, we propose a solution method, namely ERSPA. Compared with the minimal spanning tree and the greedy algorithms, our ERSPA algorithm gives better performance in terms of the average number of relay sensors. This is because ERSPA places the relay sensors in optimal places to connect the maximum number of initial connected sub-graphs such that the average number of relay sensors can be minimized.

There are still various uncovered issues in wireless networks. For example, the energy aware routing in mobile phone sensor networks, and message ferry routing in mobile phone sensor networks [77-79]. The above issues might be interesting for possible future work.

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, pp. 102-114, 2002.

[2] J. Yick, B. Mukherjee and D. Ghosal,"Wireless sensor network survey," *Computer Networks*, pp. 2292-2330, 2008.

[3] D. Estrin, R. Govindan, J. Heidemann and S. Kumar, "Next century challenges: scalable coordinationin sensor networks," *in Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks*, 1999.

[4] N. Bulusu, D. Estrin, L. Girod and J. Heidemann, "Scalable Coordination for wireless sensor networks: self-configuring localization systems," *in Proceedings of the Sixth International Symposium on Communication Theory and Applications*, 2001.

[5] G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J. Sallai and K. Frampton, "Sensor network-based countersniper system," *in Proceedings of the Second International Conference on Embedded Networked Sensor Systems*, 2004.

[6] J. Yick, B. Mukherjee and D. Ghosal, “Analysis of a prediction-based mobility adaptive tracking algorithm,” *in Proceedings of the IEEE Second International Conference on Broadband Networks*, 2005.

[7] M. Castillo-Effen, D.H. Quintela, R. Jordan, W. Westhoff and W. Moreno, “Wireless sensor networks for flash-flood alerting,” *in Proceedings of the Fifth IEEE International Caracas Conference on Devices, Circuits, and Systems*, 2004.

[8] T. Gao, D. Greenspan, M. Welsh, R.R. Juang and A. Alm, “Vital signs monitoring and patient tracking over a wireless network,” *in Proceedings of the 27th IEEE EMBS Annual International Conference*, 2005.

[9] K. Lorincz, D. Malan, T.R.F. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh and S. Moulton, “Sensor networks for emergency response: challenges and opportunities,” *IEEE Pervasive Computing*, 2004.

[10] G. Wener-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees and M. Walsh, “Deploying a wireless sensor network on an active volcano,” *IEEE Internet Computing*, 2006.

[11] C. Chien, I. Elgorriaga, and C. McConaghy, “Low-power direct-sequence spread-spectrum modem architecture for distributed wireless sensor networks,” *in Proceedings of the 6th ACM International Symposium on Low Power Electronics and Design (ISLPED’01)*, pp. 251-254, 2003.

[12] A. Sinha and A. Chandrakasan, “Dynamic power management in wireless sensor networks,”*IEEE Design and Test of Computers*, Vol. 8(2), pp. 62-74, 2001.

[13] Y. Wei, J. Heideman, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," *in Proceedings of 21th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, pp. 1567-1576, 2000.

[14] J. Moy, *"OSPF version 2,"* RFC 2178, Internet Engineering Task Force, 1997.

[15] J. Moy, *"OSPF : Anatomy of an internet routing protocol,"* Addison-Wesley, 1998.

[16] A. Boukerche, X. Cheng, and J. Linus "Energy-aware data-centric routing in microsensor networks," *in Proceedings of the 6th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, pp. 459-468, 2003.

[17] R. C. Shah and J. M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," *in Proceedings of Wireless Communications and Networking Conference*, 2001.

[18] X. Hong, M. Gerla, R. Bagrodia, J. K. Taek, P. Estabrook and P. Guangy, "The mars sensor network: efficient, power aware communications," *in Proceedings of Military Communications Conferences (MILCOM'01)*, pp. 418-422, 2001.

[19] X. Hong, M. Gerla, W. Hanbiao, and L. Clare, "Load balanced, energy-aware communications for mars sensor networks," *in Proceedings of IEEE Aerospace Conference*, pp. 1109-1115, 2002.

[20] K. Sohrabi, J. Pottie, “Protocols for self-organization of a wireless sensor network,” *IEEE Personal Communications,*Vol. 7(5), pp. 16-27, 2000.

[21] S. Lindsey and C. Raghavendra, “PEGASIS: Power-efficient gathering in sensor information systems,” *in Proceedings of IEEE Aerospace Conference,*Vol. 3, pp. 1125-1130, 2002.

[22] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” *in Proceedings of the ACM MobiCom’99*, pp. 174-185, 1999.

[23] J. Kulik, W. R. Heinzelman, and H. Balakrishnan, “Negotiation-based protocols for disseminating information in wireless sensor networks,” *Wirelwss Networks*, pp. 521-534, 2002.

[24] C. Chiasserini, I. Chlamtac, P. Monti, and A. Nucci, “An energy-efficient method for nodes assignment in cluster-based ad hoc networks,” *ACM Wireless Networks*, Vol. 5(10), pp. 223-231, 2004.

[25] S. Bandyopadhyay and E. J. Coyle, “An energy efficient hierarchical clustering algorithm for wireless sensor networks,” *IEEE INFOCOM,*Vol. 3, pp. 1713-1723, 2003.

[26] M. Younis, M. Youssef, and K. Arisha, “Energy-aware routing in cluster-based sensor networks,” *in Proceedings of the 10th IEEE International Symposium*, pp. 129-136, 2002.

[27] V. Mhatre and C. Rosenberg, “Homogeneous vs. homogeneous clustered sensor Networks: a comparative study,” *IEEE International Conference on Communications (ICC’04)*, 2004.

[28] A.-A. Abbasi and M. Younis “A survey on clustering algorithms for wireless sensor networks,” *Computer Communications* , pp. 2826-2841, 2007.

[29] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks* , vol. 52(12), pp. 660-669, 2008.

[30] S. Basagni, “Distributed clustering algorithm for ad hoc networks,” *in Proceedings of the International Symposium on Parallel Architectures Algorithms and Networks* , pp. 310-315, 1999.

[31] H. Chan and A. Perrig, “ACE: an emergent algorithm for highly uniform cluster formation,” *in Proceedings of the 1st European Workshop on Sensor Networks*, pp. 154-171, 2004.

[32] O. Younis and S. Fahmy, “HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 3(4), pp. 660-669, 2004.

[33] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” *in Proceedings of HICSS*, Hawaii, pp. 1-10, 2000.

[34] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on Wireless Communications,*Vol. 1(4), pp. 660-669, 2002.

[35] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla and T.-W. Chen, "Scalable routing strategies for ad hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, Vol. 17(8),pp. 1369-1379, 1999.

[36] W. Chen, N. Jain, and S. Singh, "ANMP: Ad hoc network management protocol," *IEEE Journal on Selected Areas in Communications* , Vol. 17(8), pp. 1506-1531, 1999.

[37] T. Kwon and M. Gerla, "Clustering with power control," *in Proceedings of Military Communications Conferences (MILCOM'99)*, pp. 1424-1428, 1999.

[38] K. Mulmuley, "*Computational Geometry,*" Prentice Hall, 1994.

[39] F. Aurenhammer, "Voronoi diagrams - a survey of a fundamental geometric data structure," *ACM Computing Surveys 23*, pp. 345-405, 1991.

[40] Q. Du, V. Faber and M. Gunzburger, "Centroidal voronoi tessellations: applications and algorithms," *SIAM Review 41*, pp. 637-676, 1999.

[41] W. Zhao and M. Ammar, "Message ferrying: proactive routing in highly partitioned wireless ad-hoc networks," *in Proceedings of the 9th IEEE Workshop on Future Trends in Distributed Computing Systems*, pp. 308-314, 2003.

[42] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," *in Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing* , pp. 187-198, 2004.

[43] M. M. B. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," *in Proceedings of the Interna-*

tional Symposium on Mobile Ad Hoc Networking and Computing Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, pp. 37-48, 2006.

[44] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," *Technical report, Duke University*, 2000.

[45] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," *in Proceedings of IEEE INFOCOM*, pp. 1407-1418, 2005.

[46] R. C. Suganthe and P. Balasubramanie, "Efficient routing for intermittently connected mobile ad hoc network," *International Journal of Computer Science and Network Security*, pp.184-190, 2008.

[47] R. C. Suganthe and P. Balasubramanie, "Improving Qos in disconnected mobile ad hoc network," *Journal of Mobile Communication*, pp.99-104, 2008.

[48] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: the singlecopy case," *IEEE/ACM Trans. on Networking*, pp.63-76, 2008.

[49] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: the multicopy case," *IEEE/ACM Trans. on Networking*, pp.77-90, 2008.

[50] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *IEEE Communications Surveys*, 2006.

[51] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, pp.134-141, 2008.

[52] A. A. Hanbali, M. I. Inria, V. Simon, E. Varga and I. Carreras, "A survey of message diffusion protocols in mobile ad hoc networks," *International Conference on Performance Evaluation Methodologies and Tools*, 2008.

[53] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," *in Proceedings of IEEE INFOCOM* , 2005.

[54] R. Viswanathan, J. Li, and M. Chuah, "Message ferrying for constrained scenarios," *in Proceedings of the 6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, pp. 487-489, 2003.

[55] M. Chuah and P. Yang, "A message ferrying scheme with differentiated services," *in Proceedings of IEEE MILCOM*, 2005.

[56] R. C. Shah, S. Jain, and W. Brunette, "Data mules: modeling a three-tier architecture for sparse sensor networks," *in Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 30-41, 2003.

[57] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadline," *in Proceedings of the 25th IEEE International Real Time Systems Symposium*, 2004.

[58] Y. Gu, D. Bozdağ, E. Ekici, F. Özgüner, and C.-G. Lee, "Partitioning based mobile element scheduling in wireless sensor networks," *in Proceedings of the Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, 2005.

[59] D. Applegate, R.E. Bixby, V. Chvatal, and W. Cook, "On the solution of traveling salesman problems," *Documenta Mathematica*, Extra Volume ICM III, pp. 645-656, 1998.

[60] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling salesman problem," *Operation Research*, pp. 498-516, 1973.

[61] R.-J Wan and C.-W. Yi, "Asymptotic critical transmission radius and critical neighbor number for K-connectivity in wireless ad hoc Networks," *in Proceedings of the 5th ACM international symposium on mobile ad hoc networking and computing*, 2004, pp. 1-8.

[62] P. Santi, "The Critical Transmitting range for connectivity in mobile ad hoc networks," *IEEE Transaction on Mobile Computing*, vol. 4, pp. 310-317, 2005.

[63] H. Koskinen, "A Simulation-based method for predicting connectivity in wireless multihop networks," *Telecommunication Systems*, vol. 26, pp. 321-338, 2004.

[64] H. Zhang and J. C. Hou, "Maintain sensing coverage and connectivity in large sensor networks," *Ad Hoc and Sensor Networks*, vol. 1, pp. 89-124, 2005.

[65] D. Tian and N. D. Georganas, "Connectivity maintence and coverage preservation in wireless sensor networks," *Ad Hoc Networks*, vol. 3, pp. 744-761, 2005.

[66] H. Gupta, S. R. Das, and Q. Gu, "Connected sensor cover: self-organization of sensor networks for efficient query execution," *in Proceedings of MobiHoc*, 2003.

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

[68] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *IEEE Transaction on Computers*, pp. 1-19, 2001.

[69] E. L. Lloyd and G. Hue, "Relay node placement in wireless sensor networks," *IEEE Transactions on Computers*, vol. 58, pp. 134-138, 2007.

[70] I. Mandoiu and A. Zelikovsky, "A note on the MST heuristic for bounded edge-length steiner trees with minimum number of steiner points," *Information Processing Letters*, vol. 69, pp. 53-57, 1999.

[71] D. Chen, D.-Z. Du, X. Xu, G.-H. Lin, L. Wan, and G. Xue, "Approximations for steiner trees with minimum number of steiner points", *Journal of Global Optimization*, vol. 18, pp. 17-33, 2000.

[72] X. Cheng, D.-Z. Du, L. Wang and B. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Networks*, vol.14, pp. 347-355, 2008.

[73] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, "On energy provisioning and relay node placement for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 2579-2590, 2005.

[74] H. Koskinen, J. Karvo, and O. Apilo, "On improving connectivity of static ad-hoc networks by adding nodes*,"* *Med-Hoc-Net*, 2005.

[75] D. Ganesan, R. Christescu, and B. B. Lozano, "Power-efficient sensor placement and transmission structure for data gathering under distortion constrain," *ACM Transcactin on Sensor Networks*, vol. 2, pp. 155 - 181, 2006.

[76] K. Mulmuley, "Computational Geometry," *Prentice Hall*, 1994.

[77] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, pp.140-150, 2010.

[78] M. Musolesi, M. Piraccini, K. Fodor, A. Corradi, and A. T. Campbell, "Supporting energy-efficient uploading strategies for continuous sensing applications on mobile phones," *in Proceedings of Eighth International Conference on Pervasive Computing*, 2010.

[79] S. Lee and M. F. Younis, "EQAR: Effective QoS-aware relay node placement algorithm for connecting disjoint wireless sensor subnetworks," *IEEE Transactions on Computers*, vol. 60, pp. 1772-1787, 2011.