uOttawa

L'Université canadienne
Canada's university

**FACULTÉ DES ÉTUDES SUPÉRIEURES
ET POSTOCTORALES**

uOttawa

L'Université canadienne
Canada's university

**FACULTY OF GRADUATE AND
POSDOCTORAL STUDIES**

Di Tian

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Computer Science)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Node Activity Scheduling Schemes in Large-scale Wireless Sensor Networks

TITRE DE LA THÈSE / TITLE OF THESIS

Nicolas Georganas

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Michel Barbeau	Azzedine Boukerche
Ivan Stojmenovic	Terrence Todd

Gary W. Slater

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

NODE ACTIVITY SCHEDULING SCHEMES IN LARGE-SCALE WIRELESS SENSOR NETWORKS

By
Di Tian

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
UNIVERSITY OF OTTAWA
OTTAWA, ONTARIO
SEPTEMBER 2004

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-494-11029-5
Our file Notre référence
ISBN: 0-494-11029-5

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

To my parents and my husband, for their love and support,
As a gift to Kevin, my lovely baby son.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I am extremely grateful to Dr. Nicolas D. Georganas, supervisor of my thesis, for his excellent guidance, constant encouragement, and support during this research. He has always been my mentor in my research and beyond the academic scope. I want to thank my parents for their love and support. I would like to thank my husband Zhijun who has been my side with his love, encouraging me to continue my research work whenever I feel frustrated.

I also express my appreciation to Professor Michel Barbeau, Professor Emil M. Petriu and Professor Ivan Stojmenovic for their valuable comments and suggestions on my papers and my thesis proposal. I want to thank Francois Malric and Amir Ghavam for their assistance. I thank the former and current members of MCRlab and Discover Lab. They have all made my life and work at University of Ottawa an enjoyable experience.

Ottawa, Di Tian

July, 20, 2004.

Abstract

In wireless sensor networks that consist of a large number of low-power, short-lived, unreliable sensors, one of the main design challenges is to obtain long system lifetime without sacrificing sensing quality, i.e. sensing coverage in this context. In this thesis, we first propose a node-scheduling scheme, which can reduce system overall energy consumption, therefore increasing system lifetime, by identifying redundant nodes with respect to sensing coverage and then assigning them an off-duty operation mode which has lower energy consumption than the normal on-duty mode. Our scheme aims at completely preserving original sensing coverage. Practically, sensing coverage degradation caused by location error, packet loss and node failure is very limited, not more than 1%, as shown by our experimental results. We implement the proposed scheme in NS-2, as an extension of the LEACH protocol and compare its energy consumption with the original LEACH. Simulation results exhibit noticeably longer system lifetime with our scheme as compared to earlier algorithm. The first scheme we propose aims at completely preserving sensing coverage. This, however, requires each node to get, in some way, the knowledge of its own and its neighbors' location information. Also, in that scheme, each node has to perform some calculations to determine whether to take an off-duty status. To alleviate these restrictions, we propose and study several alternative node-scheduling schemes, which cannot guarantee

the complete preservation of the original system coverage, but are nonetheless more light-weighted and flexible than the previous one. The simulation results compare these schemes with the previous one and demonstrate their effectiveness. In a single wireless sensor network, sensors are performing two operations: sensing and communication. Therefore, there might exist two kinds of redundancy in the network. Most of the previous work addressed only one kind of redundancy: sensing or communication alone. Although there have been research efforts trying to combine consideration of coverage and connectivity maintenance in a single activity scheduling, their theoretical basis for safe scheduling integration condition is only applicable in those networks that are initially fully covered by sensors. Random node deployment often makes initial sensing holes inside the deployed area inevitable, even in an extremely high-density network. Therefore, in this thesis, we enhance these works to support general wireless sensor networks by proving another conclusion: "the communication range is twice the sensing range" is the sufficient condition and the tight lower bound to ensure that complete coverage preservation implies connectivity among active nodes, if the original network topology (consisting of all the deployed nodes) is connected. Also, we extend the results to k-degree network connectivity and k-degree coverage preservation

Chapter 1

Introduction

1.1 Motivation and Contribution

Recently, the concept of wireless sensor networks has attracted a great deal of research attention due to a wide-range of potential applications that are promised by such networks, e.g. battlefield surveillance, machine failure diagnosis, biological detection, home security, smart spaces, inventory tracking. [19] [26] [4] [45]. A wireless sensor network consists of tiny sensing devices, deployed in a region of interest. Each device has sensing, processing and wireless communication capabilities, which enable it to obtain information about the environment and to report to the remote base station (remote user). The base station aggregates and analyzes data received and decides whether there is an abnormal or concerned event occurrence in the monitored area.

In wireless sensor networks, the energy source provided for sensors is usually battery power, which has not yet reached the stage for sensors to operate for a long time without recharging. Moreover, since sensors are often intended to work in remote or hostile environments, such as a battlefield or desert, it is undesirable or impossible to recharge or replace the battery power for them. However, long system lifetime is always expected by many monitoring applications. The system lifetime, which is measured by the time till all nodes have been drained out of their battery power or the network no longer provides an acceptable event detection ratio, directly affects network usefulness. Therefore, conserving energy resource and prolonging system lifetime is an important issue in design of large-scale wireless sensor networks.

In wireless sensor networks, all nodes share a common sensing task, which implies that not all sensors are required to continuously monitor the environment during the while system life. Making some nodes sleep has no affect on the quality of monitoring service as long as there are enough active nodes to assure it. Furthermore, due to low cost of sensors, it is well accepted that the nodes in wireless sensor networks can be deployed with a very high density (even up to 20 nodes/m^3 [65]). In such highly dense networks, if all the sensors were vigilant all the time, a single event would trigger many redundant reports simultaneously travelling through the network, the network would suffer unnecessarily excessive power consumption and packet collision. Therefore, to prolong system lifetime, we can initially deploy a large number of sensors and schedule them to work alternatively, i.e. redundancy is exploited for energy saving.

In this thesis, we first propose a novel node-scheduling scheme for dynamically configuring nodes' sensing activity to minimize the number of active nodes while maintaining the original system coverage. In the scheme, each node autonomously and periodically determines its work status based on its own and neighbor positional information. To preserve sensing coverage, a node is eligible for off-duty only when its neighbors can help it to monitor its whole sensing area. To avoid simultaneous removing when two neighboring nodes expect each other's helping, a back-off delay is introduced to let each node postpone its evaluation with a random period of time.

This scheme relies on location information to calculate the union of overlapping areas among multiple nodes. It aims to completely preserve the system original sensing coverage. To alleviate the dependence on location, an intuitive thought is to relax the requirement of coverage preservation. In the second part, we propose three such alternative location-free scheduling schemes, that are based on neighbor number, nearest neighbor distance and probability, respectively. The common idea is that users are able to select the level of coverage percentage. Based on this level, the minimal neighbor number K, the nearest neighbor distance D or the off-duty probability ρ is determined by using a given expression or prior collected data-pairs. During the scheduling, each node determines its work status according to the corresponding value.

Monitoring is just one duty undertaken by sensors. Due to limited communication capability, each sensor has to serve as router to help others to forward data packets to remote base stations. Similarly, redundancy also exists in the communication domain in highly dense networks. Energy would be wasted when all the nodes continuously are listening to the media, but if adaptively turning off radio at some nodes has no influence on network connectivity and communication performance. In the literature, several protocols [80] [44] [11] have been proposed for dynamic managing node activity in the communication domain.

In highly dense wireless sensor networks, energy waste always exists if we only remove one kind of redundancy. Minimizing active nodes in both domains should consider network connectivity and sensing coverage at the same time. Without sufficient coverage, the network cannot guarantee the quality of monitoring service. Without network connectivity, active nodes may not be able to send data back to base stations. Most of the previous work addressed only one kind of redundancy: sensing or communication alone. [78] [33] are the earliest papers to address how to combine coverage and connectivity maintenance in a single activity scheduling. They both proved that "the communication range size is at least twice that of the sensing range" is the sufficient condition to ensure that a full coverage of a convex area implies connectivity among active nodes inside the convex area. However, we argue that their proofs only guarantee safe scheduling in those networks where all the deployed nodes can cover a convex hull without sensing holes inside. Random sensor deployment (such as throwing sensors from an aircraft) is always claimed as one of the major advantages of wireless sensor networks over traditional wired sensor networks. When such random deployment scheme is used, it is hard or impossible to 100% guarantee complete coverage of the monitored region even if the node density is extremely high. Therefore, in the third part of this thesis, we enhance their work to support general random wireless sensor networks by providing the proof of another conclusion: "the communication range size is twice that of the sensing range" is the sufficient condition and the tight lower bound to ensure that complete coverage preservation implies connectivity among active nodes, if the original network topology (induced by all the deployed nodes) is connected. It is worth to notice that node-scheduling algorithms with the capability of complete coverage preservation have existed in the literature, for instance [75] [77]. Furthermore, we extend conclusion from 1-degree to k-degrees for applications that may require different tolerance against node failures.

1.2 Publications

Referred book chapters:

1. D. Tian and N.D. Georganas. Book Chapter in Algorithms and Protocols for Wireless and Mobile Systems. Published by CRC (in press).

Referred journals:

1. D. Tian and N.D. Georganas. A Node Scheduling Scheme for Energy Conservation in Large Wireless Sensor Networks. Journal of Wireless Communications and Mobile Computing, Special issue on Algorithmic, Geometric, Graph, Combinatorial, and Vector Aspects of Wireless Networks and Mobile Computing, 2003: 3:271-290
2. D.Tian, N.D.Georganas. Location and Calculation Free Node-Scheduling Schemes in Large Wireless Sensor Networks. AdHoc Networks Journal(Elsevier Science), 2(2004): 65-85
3. D.Tian, N.D.Georganas. Connectivity Maintenance and Coverage Preservation in Wireless Sensor Networks. AdHoc Networks Journal(Elsevier Science) (to appear)

Referred Conference papers:

1. D. Tian and N.D. Georganas, "A Coverage-preserved Node Scheduling scheme for Large Wireless Sensor Networks," In proceedings of First International Workshop on Wireless Sensor Networks and Applications (WSNA'02), Atlanta, USA, September 2002

2. D. Tian and N.D. Georganas, "Energy Efficient Routing With Guaranteed Delivery in Wireless Sensor Networks," In proceedings of IEEE Wireless Communications and Networking Conference 2003(WCNC'03), New Orleans, Apr 2003

3. D.Tian, N.D.Georganas. Connectivity Maintenance on Coverage Preservation in Wireless Sensor Networks. In proceedings of IEEE Canadian Conference on Electrical and Computer Engineering(CCECE 2004), May 2-5, 2004, Niagara Falls, Canada

1.3 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2: introduces wireless sensor networks. It provides a literature review that covers the coverage and detectability problem in wireless sensor networks, activity scheduling algorithms in the sensing domain, activity scheduling and topology control algorithms in the communication domain, communication schemes in wireless sensor networks, and other related issues.

Chapter 3: defines terms and notations that will be used in the next chapters. It describes wireless sensor networks from two perspectives: communication and sensing. These descriptions are also used by our proposed algorithms.

Chapter 4: presents a location_based node scheduling scheme, which can prolong system lifetime by removing redundant nodes in the sensing domain. We describe several models to investigate off-duty eligibility for the cases when nodes have the same sensing range or when nodes have different sensing ranges or when only directional information is available for use. The proposed off-duty eligibility rule guarantees that the original sensing coverage can be completely preserved in theory. We then design a back-off based node scheduling algorithm based on the basic model. We implement this node scheduling scheme and evaluate its performance.

Chapter 5: describes three alternative location_free node scheduling schemes in the sensing domain: neighbor_number_based, nearest_neighbor_based, probability_based. We also evaluate their performance and compare with the located_based one.

Chapter 6: provides proof for: "the communication range size is twice that of the sensing range" is the sufficient condition and the tight lower bound to ensure that complete coverage preservation implies connectivity among active nodes if the original network topology (consisting of all the deployed nodes) is connected. We also extend the result for k-degree by proving that: "the communication range size is twice that of the sensing range" is the sufficient condition to ensure that k-degree coverage preservation implies k-degrees connectivity among active nodes if the original network topology (consisting of all the deployed nodes) is k-connected. These conclusions provide the theoretical basis for integrating activity scheduling in two domains into a single operation.

Chapter 7: concludes the thesis and identifies several future research directions.

Chapter 2

Background and Related work

2.1 Wireless Sensor Networks

Recent advances of Micro Electro-Mechanical Systems(MEMS) technology, low power and highly integrated digital electronics have fostered the emergence of inexpensive and low-power tiny devices, called micro-sensors. These devices are capable of measuring a wide variety of ambient conditions(such as temperature, pressure, humidity, nose and lighting) and detecting the abnormal or concerned event occurrence in the vicinity. Although the sensing capability of an individual micro-sensor may be restricted in range and accuracy, aggregating the detections from multiple micro-sensors will generate reliable "view" of environment. The wireless communication capability possessed by micro-sensors enables such coordination.

A wireless sensor network is envisioned as consisting of hundreds or thousands of tiny sensor devices randomly deployed in a terrain of interest to cooperatively perform a big sensing task. Compared with a traditional system composed of large, expensive and powerful macro-sensors wired together, wireless sensor networks are cheap in deployment and networking expense. Small size, light weight and low cost of micro-sensors allow us to "carelessly" scatter these devices in a monitored area(such

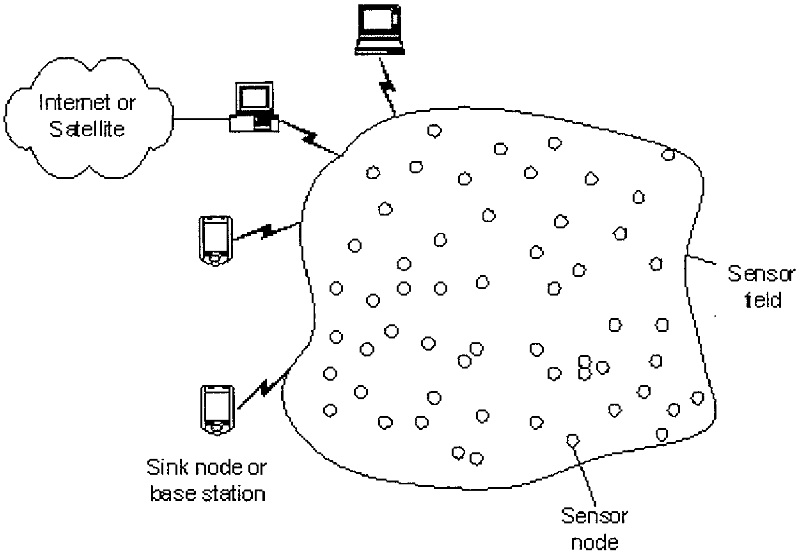


Figure 2.1: Typical scenario of wireless sensor network architecture

as throwing them out from an aircraft), and then making them self-organized in an ad hoc manner. Although a single micro-sensor is limited in sensing capability, the sheer number of nodes in wireless sensor networks guarantees enough redundancy in generated data and therefore the sensing accuracy would not be degraded. Furthermore, wireless sensor networks are more fault-tolerant than traditional macro-sensor networks because failure of a single micro-sensor would not render the entire system useless.

Figure 2.1 illustrates a typical scenario of wireless sensor networks [4]. Sensor nodes are scattered in a terrain of interest, called sensor field. Each of them has the capability of collecting data about environment and reporting to sink node(or base station). Exterior users may access data in wireless sensor networks from the sink nodes via internet or satellite. There can be multiple sink nodes in wireless sensor networks, but the number of them is much smaller than that of sensor nodes. Some

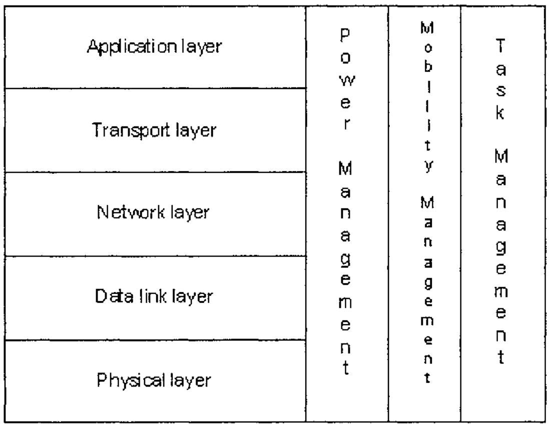


Figure 2.2: Sensor network communication protocol stack

researchers suggest to diffuse another kind of nodes, called gateways, into wireless sensor networks. These gateway nodes are more powerful than sensor nodes and therefore can perform local network management functions such as data aggregation, node organization, status assignment, etc. Sink nodes and gateway nodes can be mobile, but most of recent research results on wireless sensor networks assume that sensor nodes are stationary. That is because the cost and size of current sensor devices would increase dramatically when the mobility component is added.

Also, communications among sensor nodes, gateways and sink nodes are all in wireless manner. Since the communication range of a sensor node is short, due to its constrained energy supply, multiple-hop relaying is necessary for long-distance transmission. The communication protocol stack shared by all nodes, as illustrated in Figure 2.2, is proposed in [4]. The protocol stack consists of the application layer, transport layer, network layer, data link layer, physical layer as well as three planes. The functions of all layers conform to the specification of the ISO Seven Layer model. Note that although a layered architecture is shown in the figure, it does not mean that different layers are implemented in a completely independent way and there is no any interaction among layers. Actually, due to the application-related nature of wireless senor networks, a cross-layer design (sharing information between layers) can achieve a better performance than an independent-layer design [31]. For instance, SPIN [45] uses data naming and negotiation to suppress the transmission of redundant data in order to reduce unnecessary communication overhead. LEACH [31] and Directed Diffusion [36] aggregate multiple data packets upon routing to reduce the amount of data transmitted over long distance. In this stack, power, mobility and task management planes are responsible for providing power, mobility and task-related information to all the layers, so that these layers can exploit such information to optimize their operations. For instance, the mobility management plane is responsible for detecting movement of neighboring nodes. It can notify the network layer of any change of neighbor information so that the network layer can update routing table correspondingly. The power management plane is responsible for monitoring power usage of each sensor node. If the remaining power of a sensor node is not high enough, its future activity should be carefully controlled. The task management plane is used to determine when a sensor node will participate into a sensing task. The sensor nodes that are not joining the sensing task will not generate data reports. The task management plane can also be used to control nodes' communication activity. Not all of sensor nodes are necessarily in active status to relay packets for others all the time. Some nodes may enter into sleep mode to save energy and periodically wake up for sending and receiving packets.

The concept of wireless sensor networks have promised many civil and military applications. In the battle field, timely, detailed and valuable information about critical terrains and opposing forces can be quickly obtained by throwing micro-sensors from aircrafts. Destruction of some micro-sensors by hostile actions would not influence a military operation as much as that of traditional powerful macro-sensors [31]. In environmental aspects, sensors can be used for detecting fire and floods, tracking the movements of birds, insects or other animals. Other applications of wireless sensor networks include drug adminstration in hospitals, telemonitoring of human physiological data, smart environment, temperature control of an office building, interactive museum, monitoring car thefts and etc [31].

From random deployment and self-organization's perspective, wireless sensor networks are similar to wireless ad hoc networks. However, there are still a lot differences between them:

- The number of sensor nodes in a wireless sensor network can be much higher than the nodes in an ad hoc network. It is quite likely that sensors are deployed in a high density, even up to 20 nodes/m^3 [65]. Algorithm and protocol design for wireless sensor networks must scale up to large-populated networks. High density in node distribution leads to highly correlated data generated by multiple sensor nodes for the same event. Such redundancy can be exploited to optimize power and bandwidth utilization.

- Most ad hoc networks are based on one-to-one communication. While in wireless sensor networks, data flows are mainly query diffusion from a single or a few sink nodes to all the sensor nodes, or data collection in reverse direction. Therefore network layer design for wireless sensor networks focuses on efficient broadcasting or reverse-broadcasting protocols.

- Sensor nodes are limited in processing capability, storage space and transmission range. They are more severely constrained in energy supply than other wireless devices (such as PDA, celluar phones, etc). Energy efficiency is the primary challenge of wireless sensor networks.
- Due to low-cost manufacturing process, careless deployment method and hostile working environment, sensors are prone to failure. Fault-tolerance becomes one of important requirements in design of wireless sensor networks.

Due to such differences, although many protocols have proposed for wireless sensor networks, they are not well applicable or usable in wireless sensor networks if without any modification. Alternative approaches need to be explored in this new context.

2.2 Literature Review

In the previous section, we introduced what is a wireless sensor network and described it in the following aspects: network organization, operation mode and node characteristic and communication architecture. In this section, we will review the algorithms and protocols that are related to our work.

2.2.1 On coverage and detectability in wireless sensor networks

To the best of our knowledge, the earliest paper (earlier than our work in Chapter 4 and 5) on sensing scheduling appears in [83]. In this paper, Ye et al proposed a probing-based density control algorithm that exploits redundant to extend system lifetime. Their algorithm selects a subset of sensor nodes to monitor the environment initially and maintains their working mode until they run out of power or are damaged.

Other nodes fall asleep and wake up occasionally to probe their local neighborhood. They start working only if there is no working node within their probing range. In the algorithm, geometry knowledge is used to derive the approximate relationship between probing range and expected node density. The purpose of their design is to control node density instead of preserving the sensing coverage. In addition, their algorithm is implicitly restricted to the scenarios where all sensor nodes have exactly the same sensing range. Furthermore, their algorithm drains out energy resource in a subset of sensor nodes, and then depletes another subset's ones, without balancing energy consumption among all the sensor nodes.

In [77], Yan et al proposed another node-scheduling algorithm in the sensing domain. Their algorithm splits time into rounds T. Initially, each sensor node selects a random reference point within $[0, T)$, and exchanges its reference point and its location information with neighbors. Scheduling starts from each sensor node dividing its sensing area into small grids. For each grid point, the node finds all the neighbors that can cover this point and sorts their reference points in an ascending order. Then the node sets its start work time as the middle between its own reference point and the immediate predecessor in the sequence of reference nodes and sets the end work time as the middle between its own reference point and the immediate successor in the sequence of reference points. Then the node can determine its work duration by joining the work time for all the grid points it can cover.The algorithm can support differentiated sensing by extending the work time proportionally to the required redundancy along the time line to adjacent working spaces. However, for each sensor node, its computation load is at least $O(N_g * N_n * log(N_n))$, where N_n is average number of neighbors, and N_g is the number of grids in a sensing area. Furthermore, to reduce energy consumption in exchanging reference points, the algorithm uses a fixed reference point strategy and performs once at the first beginning of system operation. So, the algorithm is only usable in static networks.

Both above algorithms are distributed ones, only using local neighbor information for coordination. Besides them, there are some centralized algorithms in the literature.

In [13], Ko et al presented how to use their previous research results in [12], to solve the node-scheduling problem in the sensing domain of wireless sensor networks. Their algorithm splits time into cycles of K time slots. Each time slot is represented by a kind of color. The algorithms aims to divide nodes into several subsets so that each subset owns one color(representing a time slot) and the sensing coverages among all time slots are balanced. The algorithm does it by initially assigning each node a color at random, then gradually increases the distance of the nodes owning the same color until the algorithm converges. In the algorithm, each time a node changes its color, it has to propagate its change to all the other nodes in the network in order for each node to update its knowledge of the nearest node owning the same or any other different color. Such update leads to an expensive communication overhead. Furthermore, the algorithm has no intent to preserve sensing coverage.

In [71], Slijepcevic et al designed a most-constrained least-constraining heuristic approach to solve a similar problem: how to find K disjointed subsets from a node set C so that each subset can cover the whole monitored area and the cardinality of subsets is maximal. The algorithm first divides the monitored area into a set of multiple fields, denoted as A, with each field covered by an unique subset of sensor nodes. Then it minimizes the coverage of sparsely covered areas within one node subset by using the concept of critical element. A critical element is defined as a member of A covered by the minimal number of nodes in C.

In [69], Meguerdichian et al presented how to use Integer Linear Programming techniques to solve several optimal 0/1 node scheduling problems in wireless sensor networks:1). what is the minimal active node set that can completely cover a sensor field? 2).what is the minimal active node set to ensure that the average sensor field intensity for each region is above a threshold level? 3).how to assign to work at different time slices, without reducing coverage, so that number of active nodes at each time slice is minimal ?

In [47], Chakrabarty et al also leveraged Integer Linear Programming techniques for other coverage and deployment problems: (i) given a grid-based region and types of a sensor set, determine where to deploy these sensors to minimize the overall sensor cost under a certain coverage constraint (for example, each grid is covered by at least k nodes). (ii) how to deploy sensor nodes to ensure that each grid can be covered by an unique node subset, so that target position can be determined by identifying the set of observers.

In [48], Viera et al proposed to use of the well-known "Voronoi diagram" for sensing-scheduling. The algorithm first sorts nodes by the size of the Voronoi cell, and then "removes". the nodes from the network with the smallest Voronoi cell. Each time a node is removed, the Voronoi diagram is updated and a new node is identified until a certain threshold reaches. However, this paper does not provide any clue on how to set the threshold. In the same paper, Viera et al also described a centralized method for the node deployment problem. The method always adds a new node to the center of the largest empty circle, which is identified by using the well-known "largest empty circle" discovery algorithm.

In [10], Srivastava et al seemed to use the well-known min cost flow algorithm to schedule nodes' sensing activity, so that there exists one and just one active path between two fixed points all the time and the overall energy dissipation in the network is minimized. However, we can not figure out the relationship between active paths and typical sensing properties: coverage or detectability, so it is not clean where to use their algorithm.

In [41], Liu et al addressed the problem about how to use physical constraints (e.g. object shape and mobility) to determine the sensor collaboration region. They developed a centralized approach to find sensors that are relevant to object detection. Their approach leverages a dual-space transformation to map the edges of the object (simplified to a half-plane shadow) to a single point in the dual space, and to map the locations of nodes to a set of lines. Thus finding relevant nodes is transformed to tracking the cell that contains a point in the dual space. Their approach can be applied to schedule nodes' sensing activity, i.e. only "frontier" nodes are assigned an active sensing status. However, before the algorithm performs, object shape has been obtained and its mobility can be predicted a prior.

As we can see, the central approaches are supposed to optimize solutions, but they need to collect global information, thus may not be applicable in large-scale or mobile networks. Besides the coverage problem arising in general sensor network applications, exposure (or observation capability or detectability) is another interesting issue existing specially in target tracing application. Exposure is defined as whether and with how much confidence the nodes can detect an object penetrating the network along an arbitrary path [68] [67] [73]. With respect to this definition, two problems need to be solved: the minimal exposure path and the maximal exposure path, which implies system worst and best observation capability, respectively.

In [68], Meguerdichian et al implicitly used a boolean(0/1) sensing model and defined minimal /maximal exposure as the maximal /minimal distance of a target from the closest node. By exploiting the characteristics of the Vornonoi diagram and Delaunay triangulation, they proposed centralized algorithms to find the minimal and maximal exposure paths along the edges of the Vornonoi diagram or Delaunay triangulation, respectively.

In [67], Meguerdichian et al summarized a general sensing model with sensing capability as a function of sensing distance. Based on this model, they redefined exposure as an integral part of sensing model of all sensor nodes. They divided the sensing region into square grids and limited the existence of minimal exposure path within each grid element. Then they proposed to use Djikstra's shortest path algorithm for a heuristic discovery of the minimal exposure path.

In [73], Clouqueur et al described another kind of general sensing model which is a function of distance, energy and noise. Based on this model, they proposed two different approaches to collaborate detections from multiple nodes and to determine target existence. Their solution for the minimum exposure path problem is very similar to that in [67]. Both of them restrict searching space within a grid-based diagram. In [73], Clouqueur et al also proposed a sequential deployment strategy which, instead of deploying all sensors initially , adds nodes little by little until the desired minimum exposure is achieved. Furthermore, they presented an analytical model on how to optimize the number of sensors deployed at each step to minimize the cost sum of deployment and sensors.

In [15] [16], Liu et al did not propose any protocol or algorithm on coverage and detectability for wireless sensor networks. Instead, they defined several fundamental coverage properties: area coverage(the fraction of the area covered by sensors), node coverage(the percentage of sensors that can be removed from networks without reducing area coverage) and detectability(the capability of a network to detect an object penetrating the network). They analyzed these properties and studied coverage behavior in two sensing models(boolean and general) and two node distribution scenarios(two-dimensional infinite plane and two-dimensional strip case). Furthermore, they discussed the implication of analysis results to network planning and protocol performance of wireless sensor networks. To the best of our knowledge,[15] [16] are the earliest and only papers addressing coverage and detectability problems from a theoretical perspective.

Besides sensing redundancy, there is another redundancy in wireless sensor networks: communication. Combining consideration of coverage and connectivity maintenance in a single activity scheduling has been recently another active research issue.

In [78], Wang et al proved that: 1)."the communication range size is at least twice that of the sensing range" is the sufficient condition to ensure that a full coverage of a convex area implies connectivity among active nodes inside the convex area. 2)."the communication range size is at least twice that of the sensing range" is the sufficient condition to ensure that a k-degree full coverage of a convex area implies k-degree connectivity among active nodes inside the convex area. 3).If all the intersection points are k degree covered, then the convex region is k degree covered. Based on these conclusions, they divided consideration into two cases. For the case with "the communication range size is at least twice that of the sensing range", they designed an algorithm to control coverage and connectivity degree. For the case with "the communication range size is smaller than twice that of the sensing range", they described how to combine their algorithm with SPAN [11] to achieve the same goal. In [33], Zhang et al proved the conclusion 1) and 3) independently. They also proposed an algorithm to exploit the conclusion 3) to preserve sensing coverage. However, we argue that the above conclusion 1) and 2) is restricted for use. We explain this point in Chapter 6 and extend their conclusions for general scenarios in the same chapter.

In [32], Sanli et al. proposed a completely different approach to schedule node activity in the communication and sensing domain. They designed a two-phase protocol, which assumes that there are three kinds of nodes in wireless sensor networks: tiny sensor nodes, three or more powerful reference nodes and base stations. The reference nodes can be deployed in a regular pattern, such as in equilateral triangle, and equipped with GPS receivers. The data always are transferred to the reference nodes first, and then forwarded by the reference nodes to base stations. The first phase of the protocol is to build multiple multicast trees rooted at each reference node respectively in a combination way, by considering intermediate nodes' residual energy and the structure of other multicast trees rooted at other reference nodes. Those nodes that are not located at a branch (nonleaf) of any multicast tree are not serving as relay stations. Therefore they can turn their radio off to save energy. After the communication backbone has been established, each node determines whether to turn off its sensing hardware, based on its immediate lower-level neighbor number and its residual energy. The threshold for minimal immediate lower-level neighbor number is derived according to the theoretical analysis on the probability that neighbors can cover the current node's sensing area as a function of the neighbor number.

This protocol conserves energy in two aspects: turning off radio of a subset of nodes and turning off sensing units of another subset of nodes. These two kinds of subset may have overlapping.

2.2.2 On activity scheduling and topology control in ad hoc networks

In the above, we introduce activity scheduling approaches proposed for wireless sensor networks. In ad hoc networks, a similar redundancy problem exists in the communication domain as well. In that context, activity scheduling is how to assign to a subset of nodes a low-power sleep mode, while maintaining network connectivity. A connected dominating set is defined as a connected subset of the vertices of a graph, if every vertex not in the subset is adjacent to at least one vertex in the subset [43] [44] [42]. To reduce system overall energy consumption, only nodes in a dominating set keep active and can relay messages for others. Non dominating set nodes are set to a sleep mode and periodically wake up for sending and receiving packets from the associated dominating set node. In [43], Wu et al proposed a distributed algorithm for the formation of a connected dominating set, which marks a dominating node if two of its neighbors are not directly connected. In addition, they introduced two rules to reduce the size of the dominating set based on node ID. The first rule is as follows: if every neighbor of node v is also a neighbor of node u and $id(v) < id(u)$, then node v is not a dominating node. The second rule is: if every neighbor of node v is also a neighbor of node u or a neighbor of node w and $id(v) < min\{id(u), id(w)\}$, then node v is not a dominating node. In [44], the other two rules replace the ones based on node ID. The first one is to set the node with higher degree as dominating node whenever there is a tie in order to further decrease the size of the dominating set. The second is to consider energy level owned by competitors in order to balance the energy resources among them.

In [11] [80] [3], the other three algorithms for dominating set formation are presented, although the authors did not explicitly mention the concept of dominating set. In [11], a node serves as a coordinator (i.e. dominating node) only if it discovers that two of its neighbors cannot communicate with each other directly or through an existing coordinator. In [80], the network area is divided into square grids. The size of grids is calculated so that any nodes within a square can directly communicate with any nodes in the adjacent square. Thus, within each grid, only one node needs to stay awake to forward packets. The algorithm assumes that each node is aware of its own position. Therefore, it can easily know its associated grid ID. By probing whether there is an active node within its grid, each node can determine its work status. In [3], each node self-determines its participation into multi-hop routing as a relay station by evaluating its contribution to global network connectivity. However, different from the above two algorithms, which rely on positional information or local connectivity for evaluation, each node in [3] assesses connectivity according to the measured operating environment (i.e. local node density and packet loss rate).

In [81], Xu et al. proposed an energy management scheme by adaptively making nodes turn off their radio when they are not involved into data communication (transmitting and receiving). They presented how to leverage node density to increase the duration of its sleeping time. Unlike the algorithms addressed above in this subsection, nodes in this scheme self-configure their work status by only using their own application-level information or neighbor size, without considering global connectivity.

Although these algorithms are initially proposed for ad hoc networks, they are applicable in wireless sensor networks, because sensor networks have communication redundancy as well.

Topology control is an issue about how to determine the transmission power of each node to minimize power consumption while maintaining network connectivity. Instead of transmitting with the maximal possible power, nodes collaboratively determine their transmission range and redefine the network topology by the neighbor relation under certain criteria. Such topology control can be used to optimize network spatial reuse, mitigate MAC-level interference and reduce energy usage of communication [53].

Topology control algorithms usually start from constructing a sparse subgraph from the original unit graph(in which an edge exists if the distance between its endpoints is at most the maximal transmission range). The relative neighborhood graph(RNG) is a geometric and graph theoretic concept proposed by Toussaint [76]. Its definition is as follows: edge (u, v) exits if for all $w \neq u, v : d(u, v) \leq \max(d(u, w), d(v, w))$. Toussaint proved RNG is a connected and planar graph. Each node has on average about 2.5 degrees independent of graph density. The most advantage of RNG is that it can be constructed in total local fashion. In [53], Li et al proposed a localized algorithm to construct a local minimal spanning tree(LMST), which contains a minimal spanning tree(MST) as a subset. The algorithm starts from each node building its local MST consisting of itself and its one-hop neighbors. Then an edge (u,v) is kept in LMST if it exists in both node u and node v's local MST. In [54], Ovalle-Martinez et al suggested to use the longest edge in LMST as a minimal transmission radius shared by all the nodes in network. They proposed a scheme to convert LMST into

MST by using a loop breakage procedure. The procedure makes upwards messages traversing from leaves in LMST to loops along dangling edges, and then breaks loops by eliminating their longest edges until it ends at a single node. In [57], Rodoplu et al propose a distributed algorithm to build a minimal energy topology based on the concept of relay region and enclosure. Suppose node u wishes to transmit to node v, node v is said to lie in the relay region of a third node w, if less power is consumed when node w is a relay station forwarding packets for node u to node v than when node u directly transmits packets to node v. The enclosure of node u is defined as the union of the complement of relay regions of all the nodes that node u can reach by using its maximal transmission power. The sparse graph induced from edges restricted in enclosure is strongly connected and contains a minimal power path. In [49], Li et al improved the algorithm design in [57]. In [79], Wattenhofer et al came up with another approach to form a minimal energy subnetwork by using directional information other than position information. In the algorithm, each node u determines the minimal transmission power to ensure that there is at least a node neighbor can be reached in every cone of α degrees, centered at node u. Wattenhofer et al proved that when α is not larger than $2\pi/3$, the algorithm guarantees a maximum connected node set.

References [57] [82] [53] [34] [35] [72] [64] [18] discuss and explore how to use a sparse subgraph in energy-aware routing, broadcasting and activity scheduling.

2.2.3 On communication protocols in wireless sensor networks

Although routing protocols initially proposed for wireless ad hoc networks can be adapted to wireless sensor networks, unique characteristics of application, node and architecture in wireless sensor networks led to many new communication protocols specially designed for wireless sensor networks. In wireless sensor networks, energy resources are severely limited at sensor nodes. This constraint necessitates energy efficiency in the design of all the layers of a communication protocol stack for wireless sensor networks. In the past few years, application-related issues make routing in wireless sensor networks an active research issue [46].

SPIN [40] is one of the earliest communication schemes designed for wireless sensor networks. It differs from the traditional routing approach in that it aggregates data during data relaying. SPIN introduces several mechanisms in routing. First, meta-data(data descriptors) are used to name data. Then meta-data are exchanged between neighbors before data forwarding in order to suppress any redundant information transmission. Third, sensor nodes monitor their remaining energy resources and can refuse to participate into data dissemination in order to prolong their lifetimes.

Directed Diffusion [36] [26] is another well-known data-centric routing scheme for wireless sensor networks. Many later appearing protocols, such as [24] [21] [84], are built on it. Directed Diffusion combines interest dissemination and data collection into a single paradigm. In it, data collection is triggered by a sink node disseminating interest to all sensor nodes. The interest is described as an attribute-value pair format. Each sensor node records the interest and setups gradient that is used to pull data back to the sink node. When a source node has data that are matched with the interest, it sends its data back to the sink node. Like SPIN, nodes in Directed Diffusion aggregate data on routing. In addition, the sink node in Directed Diffusion reinforces a temporarily best path in order to dynamically adapt to network topology change. In [14], Krishnamachari et al built a model for data-centric routing strategy and investigated the impact of source placement and network density on performance of the data-centric routing strategy in terms of delay, robustness and energy cost. In [24], Ganesan et al improved Directed Diffusion by establishing and maintaining multiple paths in advance, so that routing can recover from a failed node quickly and efficiently. In [21], Intanagonwiwat et al observed that opportunistic data aggregation along lowest-delay tree used in Directed Diffusion is not energy efficient. So they improved it by constructing a greedy incremental tree before data aggregation. In Directed Diffusion, interest is propagated by a sink node to all sensor nodes. However, it is possible that an interest has a location-based attribute to restrict a concerned area. Therefore, in [84], Yu et al proposed an algorithm to support area-dissemination of a query in an energy-efficient way.

In [55], Shah et al proposed to use a single path from prior determined multiple alternatives on the fly in a probabilistic manner to increase system lifetime. The probability of one candidate as the next hop is determined by the energy level of the candidate and the transmission cost through it, in order to reduce energy consumption as well as balance traffic cost.

Usually, a large amount of data are generated and reported as a response to one query (or interest). So Directed Diffusion floods query through the entire network. However, in some cases, there are few events responding to a large number of queries. In [22], Braginsky et al made a compromise between event (data) flooding and query flooding by proposing a routing scheme, called rumor. In this scheme, when an event source detects an event, it tells all the other nodes within a restricted region about the occurrence of this event. After a query is generated by a query source, it walks randomly throughout the network until it reaches the event source or the intermediate node that knows that event.

In [58], Sadagopan et al viewed sensor networks as a distributed database and a query consisting of several sub queries. The proposed query scheme starts with injecting an active query packet into a network that walks along a random (possibly even pre-determined or guided) trajectory. Each node receiving the query tries to respond partially by using its pre-cached information. If the information is not up to date, the node will collect data from all the neighbors within a look-ahead of d hops. As this active query progresses through the network, it gets progressively resolved into smaller and smaller components until it is completely solved. Finally the query is returned back to the querying node as a completed response.

In [27], Ye et al proposed to use a minimal cost path to forward data from nodes to a base station. The protocol consists of two phases. The first phase is to set cost values in all nodes. The process starts from the sink node broadcasting a message, that will diffuse through the entire network. Every node, once receiving the message, adjusts its cost value according to the cost of links and the cost of the message sender. In the protocol, a back-off mechanism is introduced to reduce the communication overhead in the cost setup phase. In the data delivery phase,each node determines whether it is located on the source node's minimal cost path, therefore forwarding the data packet, based on the cost of itself, the receiver and the link. One advantage of this protocol is that it needs no addressing or node identification scheme. Only cost information is used to guide routing. Second, the protocol does not specify the definition of cost. It can be delay, energy, remaining energy or combination of them. So it can be used for any QOS (quality of service) requirement. A similar idea appears in [56]. In [28], Ye et al enhanced the above single minimal cost routing by adding a credit field into each data packet. The credit is used to indicate how much extra amount of energy can be consumed beyond the minimal cost from the source node to the sink node. Each intermediate node, once receiving a copy of data packet, will check if the already consumed cost exceeds the range indicated by the credit. If that is not the case, it will forward the copy even if it may not be located in the minimal cost path of the source node. Such multiple path data forwarding scheme improves delivery reliability.

In [62], De et al also proposeed to use meshed multiple paths for routing in wireless sensor networks. However, different from the meshed routing in [28], their scheme establishes multiple paths between two nodes proactively before data delivery. Furthermore, only one path is used although multiple paths have been setup in [62]. Each intermediate node probabilistically selects its next hop according to their latest link quality.

[63] presented another kind of multi-path routing scheme for wireless sensor networks: spliting a data packet into k small blocks, adding extra information into each block, and then sending different blocks along different paths. This kind of scheme increases the success rate compared with single-path routing because $< k$ blocks are needed to reconstruct the original data blocks.

Besides energy-efficiency and robustness, real-time is an important design requirement for many applications of wireless sensor networks, such as fire detection. In [74], He et al proposed a stateless, real-time communication protocol for wireless sensor networks. The protocol uses a well-known non-deterministic geographic forwarding approach for routing. However, several new features are added in order to mitigate network congestion and avoid a "hop point". First, each node measures the delay of data transmission to one of its neighbors. Statistical results of previous data transmissions can be used to estimate that node's traffic load. During data forwarding, only those nodes that can guarantee the specified delivery speed are selected as the next hop. Second, to further mitigate network congestion, packets may be dropped or buffered locally if the packet loss ratio of the current node is beyond a threshold. The packet loss rate is fedback from the MAC layer. Third, the current node informs its upstream nodes about the congestion so that it upstream nodes can reroute their later-coming packets to another path.

In [60], Bhatnagar et al introduced the concept of service differentiation into routing of wireless sensor networks. They defined service differentiation as that data packets with different important levels should be treated in different ways in order to guarantee delivery requirements for those data packets with a high priority. They identified several performance metrics to evaluate service differentiation, and explained how to use acknowledgement, redundancy and FEC(forward Error Correction) to implement them. They also presented a detailed description for a redundancy-based service differentiation communication protocol.

All the above network protocols deal with how to efficiently deliver data to a single destination. Actually, in wireless sensor networks, it is quite likely that multiple sink nodes have the same interests and are spatially co-located. In this case, sending a data report to multiple sink nodes separately, by using multiple unicast paths, unnecessarily consumes a lot of energy. A better approach is to organize these sink nodes with the same interest into a multicast tree and use this tree to efficiently distribute data. In [51], Mirkovic et al proposed a distributed local routing solution for multiple-sink problem. Their algorithm starts from each sink diffusing its interest within a region confined by a TTL (time to live) value. Then the algorithm selects the best merge point from those nodes that have received multiple interests in the previous phase. After that, the algorithm builds a multicast tree from that merge point to all sink nodes. Once a source node has a data report, it first sends its data report to the closest node in the multicast tree. After that node receives the data report, it will distribute the data to all sink nodes along the multicast tree.

In [61], Bhattacharya et al argued that in the cases where multiple sink nodes request data at different rates, it is inefficient to let each source node distribute its data packets to each sink node separately. A better way is to let the source node distribute multiple copies of its data packet into some intermediate nodes (called data caches) and to let each sink node retrieve data from the closest data cache. Thus, an optimal problem arises: where to place data copies to minimize the communication cost for data distribution and collection. In [61], a heuristic approach was proposed to answer this problem. In brief, the algorithm iteratively builds a complete multicast tree containing all sink nodes. Each time a new sink node is added into the already generated multicast tree, a new data cache is created at the gravity center of that new sink node and its neighbors.

Almost all the network protocols existing in the literature do not make any assumption on lower layer protocols. However, [17] supposes that the underlying MAC (multiple access control) protocol is 802.11. It modifies the RTS message by piggybacking the position information of the destination and the sender into the RTS message. Based on this position information, each receiver can determine whether it should forward the data packet to the destination. That is, if the receiver is closer to the destination than the sender, it responds to the sender via a CTS message. Also, the protocol introduces a waiting time before a node sends the CTS message. By setting the waiting time proportional to the consumed energy of the sender, energy balance among neighboring nodes can be achieved. The main advantage claimed for this protocol is that its routing does not rely on any state information, including neighbor list and routing table.

In [2], Beaufour et al described a different wireless sensor network architecture that consists of a few static sensor nodes scattered over a wide area. Based on such scenario, they proposed a data dissemination protocol that exploits a few mobile devices, called smart-tags, to carry data between disconnected sensor nodes. They implemented smart-tags by modifying the Bluetooth protocol. Their approach is valid when the time constraint is loose and there are a lot of smart-tags roaming around the network.

It may not be appropriate for wireless sensor networks to own an addressing scheme like IP-addresses due to a large number of nodes. Furthermore, routing is often associated with positional information, such as sending a query to a given geographical region or delivering data back to a sink node whose location has been known by all sensor nodes. Therefore, already existing location-based routing approaches, although not specially designed for wireless sensor networks, are well applicable in this new context. There are many location-based routing schemes in the literature. For instance, in [50], Lin et al proposed several location-based multi-path routing schemes. In these schemes, a source node initially sends a message to c best neighbors according to an existing direction-based or location-based single-path routing method (GEDIR, DIR or MFR). Then each intermediate node forwards the first received copy to the best neighbor or forwards the i-th received copy to the i-th best neighbor or forwards any received copy to the best neighbor among those that have never received the message yet. In [37], each intermediate node selects its best neighbor from its 2-hop neighbors, instead of only from 1-hop neighbors. In addition, a local flooding is performed around a concave node to guarantee delivery success in collision-free, connected networks.

Flat network protocols suffer from scalability problem. For example, some nodes such as nodes close to the sink node can be overloaded and be drained out of their energy quickly. Hierarchy is usually considered to cope with such a problem.

The earliest hierarchical protocol proposed for wireless sensor networks is LEACH [31] [30]. The idea of LEACH is to divide the system operation timeline into fixed intervals (rounds). In each round, each sensor node determines its role in a cluster, as cluster-head or non-cluster-head, based on its duty in previous rounds. Data collection starts from each non-cluster-head sending its data report to its cluster-head. Then cluster-heads aggregate all data reports received and send the compressed result directly to the base station. In LEACH, the role of cluster-head is rotated among all sensor nodes to prevent energy draining of a single node. However, LEACH is based on the assumption that each node in the network can directly communicate with each other node. Therefore, it is not scalable for large-region networks.

The idea of LEACH has inspired many other protocols. TEEN [5] is designed for time-critical applications, in which quick response is an important metric for protocol performance. TEEN also uses a hierarchical architecture. However, two thresholds(soft or hard) are broadcasted to nodes to trigger their data transmission. TEEN does not work well for those applications where periodic data collections are needed since sink node may not get any data if the thresholds are not reached. Manjeshwar et al extended TEEN by combining data collection strategies in LEACH and TEEN. In APTEEN [6] [7], sensors immediately respond to sudden environmental changes whenever their sensed data exceed some threshold values. Also, they send periodically reports to the base station even if the sensed data do not reach the threshold values.

In [66], Lindsey improved LEACH by organizing sensor nodes into a chain. Data collection starts from the sensor nodes located at the farthest end point of the chain sending their data packets to their immediate neighbors in the chain. Then those neighbors aggregate their own data with the received data and forward the compressed data to the next hop in the chain; the process is going on in both directions, until data packets from both directions arrive at the same sensor node which is supposed to transmit the final result to the remote base station. The algorithm needs less energy consumption for transmission than LEACH in that each sensor node transmits and receives not more than once. However, at most two sensor nodes can transmit data at any time instant. Therefore, excessive delay is introduced for distant nodes on the chain. Furthermore, bottleneck exists in a single node, that finally sends data to the base station.

In [16], Liu et al modified LEACH to support networks where nodes have mobile capability. First, in clustering form phase, the algorithm predicts the distance of two nodes in the future and assigns each non cluster-head node to the cluster-head which may have a minimal distance to the concerned node. Second, Liu et al observed that LEACH suffers from significant energy consumption when there are no cluster-heads selected in some rounds. They provided two approaches to avoid this situation. One is to skip the round which has no cluster-head elected. The other is to count and fix the cluster-head number in each round.

In [29], Gupta et al suggested to use a few powerful gateway nodes to organize tiny sensor nodes into clusters. However, because these gateway nodes take a critical role in the network, the system is more sensitive to failure of them. Therefore, they enhanced the existing clustered-based data collection approach such as LEACH by quickly detecting a failed gateway node and assigning its members to other alive ones.

In [59], Bandyopadhyay et al proposed to organize sensors into a multiple-layered clustering architecture for data collection. The most impressive part of their work is that they derived an optimal value of the cluster number at each layer and hop number in each cluster in order to minimize the total energy cost for data collection.

In [83], Ye et al described another different kind of hierarchial data dissemination protocol for wireless sensor networks. This protocol consists of two tiers and aims to enable mobile sink nodes to continuously receive data. Data dissemination starts from a source node actively building a grid-structure, once it detects a stimulus. The protocol assumes that sensor nodes are stationary and location-aware. Therefore, each source node can maintain the grid-structure with a little cost. With the grid structure in place, query flooding is restricted in the local grid cell where the sink node is located. Once the query reaches a sensor node located at the backbone of the grid-structure, it follows the grid structure upstream to the source node, pulling its data back to the sink node. As long as the sink node does not move out of the cell, it can continue to receive data without any stop.

From the above introduction, we find the communication protocols designed for wireless sensor networks show a lot of diversity due to different assumptions on applications, nodes and architecture. However, energy-efficiency is a common goal shared by all the protocols.

2.2.4 Other related issues

In [70], Ni et al analyzed three severe problems caused by a flooding-based broadcasting scheme: redundancy, contention and collision, named as the broadcast storm problems. To mitigate the harm of these problems, they developed several schemes, following the direction of reducing redundant rebroadcast, based on probabilistic, counter, distance, location and cluster, respectively. Their work inspired us in location-free node scheduling as explained in Chapter 5.

In [20], C.Florens et al proposed a node-scheduling algorithm for wireless sensor networks, but to solve a complete a different problem. The problem they studied is how to schedule data transmission in order to minimize the time expense for data collection from all nodes to a base station in tree-like sensor networks with directional antennas. They defined a mathematical model to analyze this problem and proposed an optimal scheduling strategy to solve it. They verified that, by using directional antennas instead of omni-directional antennas, the duration needed for data collection can be dramatically decreased. The scheduling scheme they proposed is centralized. Before performing it, the base station has to know the network topology as well as the amount of data each node needs to report.

As we recall, some node-scheduling algorithms are based on the assumption that a central node knows global network topology. In [23], Deb et al studied how to efficiently discover global topology. The algorithm starts from a central node initiating a topology discovery request message. Taking advantage of the broadcasting characteristics of wireless communication, each node can obtain the neighbor list when the topology discovery request message diffuses throughout the network. To reduce the communication overhead when nodes report their neighbor list to the central node, the algorithm organizes the network into clusters. In each cluster, only the cluster head is responsible for sending a response message back to the central node. In [23], Deb et al proposed and studied two coloring approaches to form clusters and to build a cluster tree when the request message propagates through the network.

Location awareness is an important issue for wireless sensor networks, because event detection is meaningful only when it is associated with a location. Furthermore, many protocols and algorithms, including that in Chapter 4, assume nodes have knowledge of their own position. However, it is not feasible to equip each sensor node with a GPS system due to its constraint in size, cost and energy. Sensor networks may work in the region where satellite signals are unavailable. Therefore, a wireless sensor network has to involve an energy-efficient scheme that can obtain an accurate position estimation based on a known location of a small set of nodes even in highly dense networks. In [25], Doherty et al describe an approach for estimating unknown node positions based on connectivity constraints between nodes. The basic idea is to formulate node-to-node radial constraints and angular constraints as a set of linear matrix inequalities or scalar equalities, and then solve these inequalities or equalities through linear or semi-definite programming to get the approximate values of unknown node positions. The approach has no strict restriction on the number of nodes with known positions. However, it is centralized which implies nontrivial communication overhead for information collection. [9] presents a distributed location discovery algorithm which consists of two phases. In the first phase, each node estimates its distance from its neighbors. In the second phase, some nodes with unknown location can use the ranging and known location information of their ambient reference nodes to estimate their positions. Once a node gets its location, it becomes a new reference node and broadcasts its location to help other nodes for localization. In [9], Savvides et al suggested to use distance determination technique based on the Arrival Time of the RF signal and ultrasonic signal other than RF signal strength, because the former is less sensitive to physical effects. In [38], Albowicz et al proposed a very similar approach to that in [9]. However, in [38], each undiscovered node first selects those nodes with the smallest estimation error among all available reference nodes and then uses their location knowledge to determine its own position. Thus, accuracy of position estimation will not degenerate with the distance from the initial reference nodes. In [52], an undiscovered node's position is estimated as the centroid of those reference nodes within its communication range. The advantage of such approach is that it unnecessitates any distance estimation technique. In [8], Nasipuri et al assumed that there are three or more powerful beacon nodes in the network. These beacon nodes have a long transmission range to cover the whole network area. In addition, they are equipped with a radar transmitter or smart antenna to generate a narrow directional beam instead of an omni directional signal. Furthermore, each beacon node can rotate their directional beam at a fixed speed with different initial phase. Beacon nodes broadcast beacon signals periodically to all sensor nodes. Each sensor node records the time when it receives a beacon signal from a beacon node. Based on the time difference of beacon arrival and the known position of beacon nodes, node position can be determined through triangulation. This approach does not introduce any extra communication overhead at sensor nodes except for receiving beacon signals. In addition, this approach enables quick estimation because the others estimate unknown positions iteratively.

Besides position estimation, time synchronization is another essential issue in wireless sensor networks. Many algorithms and protocols, such as [31], require nodes to operate in a synchronized way. Event classification and identification also relies on time synchronization because duplicate detection of the same event from multiple sensor nodes is recognized through time stamps. In addition, time synchronization has an impact on accurate position estimation, because some position estimation algorithms need to estimate the distance between two nodes by using the Time of Arrival approach (i.e. distance is propagation speed multiplied by propagation delay). Elson et al argued that in [39] that traditional time synchronization approaches are not applicable for wireless sensor networks because sensor nodes are powered down most of the time for energy saving purposes. Therefore, they proposed a light-weighted algorithm, which attempts to achieve time synchronization by an instantaneous phase correction after a stimulus is recorded by multiple nodes based on a common synchronization signal sent by a "third party" beacon node. To reduce skew of the receivers' local clock, they suggested to use NTP (Network Time Protocol) to discipline each node's oscillator frequency.

2.3 Summary

This chapter provided background knowledge of wireless sensor networks. It reviewed the recent advances on studying coverage and detectability problem in wireless sensor networks, and covered all the existing distributed or centralized node activity scheduling algorithms in the sensing domain, to the best of our knowledge. It also introduced several scheduling schemes to remove communication redundancy, which, although originally proposed in the context of wireless ad hoc networks, are applicable in wireless sensor networks as well. Then it listed the communication schemes of wireless sensor networks, which well show the main concerns during algorithm design. Finally, it discussed other important issues related to our work, such as time synchronization and position estimation.

Chapter 3

Preliminary

To facilitate the future description, we first define some terms and notations that will be used in the next three chapters. We view a wireless sensor network from three perspectives: location, communication and sensing.

3.1 Location Model

We consider wireless sensor networks consisting of a large number of sensors placed in a vast two-dimensional geographical region. We define $\mathring{A}$ as the region where sensors are initially deployed in and are supposed to monitor afterwards. We assume that the locations of sensors are uniformly and independently distributed in the region $\mathring{A}$. Such a random deployment strategy is claimed as one of the main advantages of wireless sensor networks over traditional wired sensor networks. It can be easily performed by dropping sensors from aircrafts. Based on this consumption, the locations of sensors can be modeled as per a homogeneous spatial Poisson process of intensity λ in a 2-dimensional space. This model is widely adopted in the literature, such as [59] [16] and [15]. In other words, the probability that the number of nodes in the area of s, denoted as $N(s)$, is equal to n is given by:

$$P[N(s) = n] = \frac{(\lambda s)^n e^{-\lambda s}}{n!}, \text{ for } s \geq 0 \text{ and } n = 0, 1, 2, ... \tag{3.1.1}$$

3.2 Communication Model

We assume that two sensors can directly exchange messages if their Euclidean distance is not larger than a communication range R and the communication range is homogeneous among all the sensors in a network. We characterize a wireless sensor network as a communication graph $G(V, E)$, where each element in the node set $V(G)$ represents a sensor in the network. For any pair of nodes v and u in $V(G)$, the edge (v, u) is in the edge set $E(G)$ if and only if the Euclidean distance between v and u, denoted as $d(v, u)$, is not larger than the communication range R. We call two nodes adjacent if they are incident to a common edge. The neighbor set of node v, denoted as $N(v)$, is the set of those nodes that are adjacent to node v. In other words, $N(v)$ contains all the other nodes that are located within the communication range R away from node v. Each node in $N(v)$ is called a neighbor of node v. We define a path P in $G(V, E)$ as an alternating sequence of nodes in $V(G)$ and edges in $E(G)$, with each edge being incident to the nodes immediately preceding and succeeding it in the sequence. P can be represented by its complete edge sequence as $P = (n_1, n_2)(n_2, n_3)$ $...(n_{l-2}, n_{l-1})(n_{l-1}, n_l)$ or simplified as $n_1 \longleftrightarrow_{(P)} n_l$. We call two nodes v and u in $V(G)$ connected, if there is a path P in $G(V, E)$, whose ends are the given nodes, and the whole graph $G(V, E)$ connected,if every pair of nodes in $V(G)$ is connected.

3.3 Sensing Model

In the literature, there are two different sensing models used.

The first model captures the common property of sensing devices with varying complexity and features, and assumes that sensing ability diminishes as distance increases. In the model summarized in [16], the sensing intensity of a node v at a point P is given as:

$$S(v,P) = \begin{cases} \frac{\alpha}{d(v,P)^{\beta}} & , A \leq d(v,P) < B \\ 0 & , \text{otherwise.} \end{cases} \tag{3.3.1}$$

where α is the energy emitted by the event occurring at point P. $d(v, P)$ is the Euclidean distance between sensor v and point P. A and B are range-related parameters of sensor v. β is decaying factor, ranging from 2.0 to 5.0. Whether an event is detectable is determined by a data fusion of N sensors with nearest distance to point P: $I_P = \Sigma_{i=1}^{N} S(s_i, P)$, where Σ can be sum or any other data fusion function used by the center control node. When $I_P \geq \theta$, the event at point P can be be observed.

The second model is a special case when $N = 1$. We call it boolean or 0/1 sensing model. That assumes each sensor can do 360^o degree observation. Sensing area $S(v)$ of node v is the maximal circular area centered at sensor v that can be well observed by sensor v. Note that although the maximal observation region of a sensor may be irregular, but an irregular region can always contain a circle centered at that sensor. The radius of $S(v)$ is called sensor v's sensing range $r(v)$.

The second model is simple, while the first one is general. Both models are widely adopted in the literature. Currently, the first one is mainly used in detectability issues in wireless sensor networks, while the second mainly in activity scheduling algorithms.

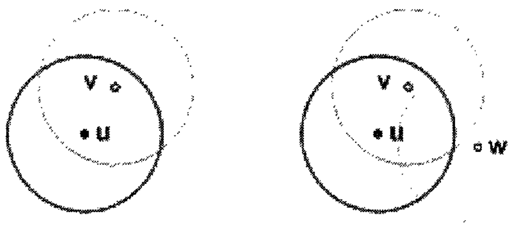


Figure 3.1: Auxiliary observable area (AOA)

Our work focuses on the activity scheduling problem and is based the second model. In the future, we will consider to transform our results to the general sensing model.

Now, we introduce the concept of sensing coverage into a network graph $G(V, E)$. We assume that all sensors have the same sensing range r in the next three sections, except that in section 4.1.2, where a special case with heterogeneous sensing range is discussed. We also define the coverage $C(X)$ of a node set X, as the union of the sensing areas covered by each node in X, i.e. $C(X) = \bigcup_{v \in X} S(v)$. The overall sensing coverage of a network graph $G(V, E)$ is $C(V(G)) = \bigcup_{v \in V(G)} S(v)$. Like in the communication model, we also define neighbors in the sensing domain. We call two nodes sensing adjacent, if their Euclidean distance is not larger than the sensing range r. The sensing neighbor set of node u is defined as $SN(u) = \{v \in V(G) | d(u, v) \leq 2r, u \neq v\}$. As shown in Figure 3.1, sensing adjacent nodes u and v share a common observed area $S(u \cap v)$. We name this area in $S(u)$ as auxiliary observable area (AOA) of node u by node v, denoted as $AOA_{v \to u}$, because it is within node v's sensing area and thereby can be monitored by node v, if node u stops working but node v is still vigilant. The AOA of node u by its sensing neighbor set is $\bigcup_{v \in SN(u)} S(u \cap v)$. The counterpart of AOA is the auxiliary unobservable area (AUA). $AUA_{v \to u}$ is defined as $S(u) - AOA_{v \to u} = S(u) - S(u \cap v)$.

3.4 Topology After sensing Scheduling

The main topic of this thesis is to study node activity scheduling schemes in the sensing domain of wireless sensor networks. Such schemes are used to assign each sensor a sensing status: active or inactive. Inactive nodes do not participate in sensing tasks and keep in a low-power sleep mode in order to save energy. After scheduling, the node set $V(G)$ in the initial network graph $G(V, E)$ is divided into two parts: $V_{active}(G)$ and $V_{inactive}(G)$. $V_{active}(G)$ consists of all the nodes obtaining an active status after scheduling. $V_{inactive}(G)$ contains the others, i.e. $V_{inactive}(G) = V(G) - V_{active}(G)$. We are interested in a subgraph of $G(V, E)$, denoted as G_{active}, which is induced by the node set $V_{active}(G)$, with the edge set as the subset of $E(G)$ consisting of those edges whose both ends are in $V_{active}(G)$. We also define an active path as a path in $G(V, E)$, whose all intermediate nodes are active ones. The counterpart is the inactive path, owning at least one inactive intermediate node. If nodes u and v in $V(G)$ can communicate through an active path, we call these two nodes actively connected, denoted as $u \longleftrightarrow_{active} v$.

3.5 Summary

Table 3.1 lists the terms and notations used in the next sections.

Table 3.1: List of terms or notations

$G(V, E)$	network graph
$V(G)$	node set of network graph $G(V, E)$
$E(G)$	edge set of network graph $G(V, E)$
$d(u, v)$	Euclidean distance between node v and node u
v	node or sensor v
$N(v)$	neighbor set of node v
$\overline{N(v)}$	the set of node in $V(G)$ that is not in $N(v)$
P	a path in $G(V, E)$
(u, v)	an edge in $E(G)$, which connects node u and node v
A	sensing field or monitoring area
$S(v)$	node v's sensing area
$r(v)$	node v's sensing range
r	sensing range shared by all the nodes in a network
$C(X)$	coverage of node set X
R	communication range shared by all the nodes in a network
$u \longleftrightarrow_{(P)} v$	path P ends at node u and node v
$V_{active}(G)$	set of nodes with active status in network graph $G(V, E)$
$V_{inactive}(G)$	set of nodes with inactive status in network graph $G(V, E)$
G_{active}	subgraph of $G(V, E)$ induced by $V_{active}(G)$
$u \longleftrightarrow_{active} v$	an active path[1] exists between node u and node v
$SN(u)$	node u's sensing neighbor set
$S(u \cap v)$	the common observed area of node u and node v
$AOA_{u \to v}$	auxiliary observable area of node u by node v
$AUA_{u \to v}$	auxiliary unobservable area of node u by node v

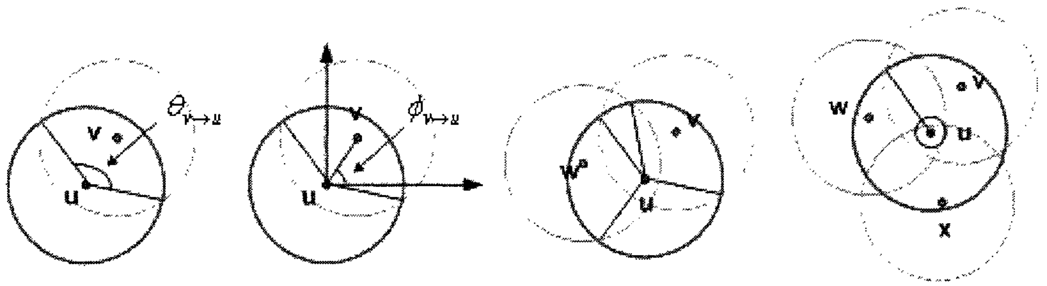
[1] all the intermediate nodes in the path are in G_{active}

Chapter 4

Location-based node-scheduling Scheme

In this chapter, we present a novel node activity scheduling scheme, which is used to configure nodes' sensing status and to schedule their *on-duty* time in large-scale wireless sensor networks. Our design was driven by the following requirements. First, since it is inconvenient or impossible to manually configure sensors after they have been deployed in a hostile or remote working environment, self-configuration is mandated. Second, the design has to be fully distributed and localized, because a centralized algorithm needs significant overhead for global synchronization and is unscalable to large-populated networks. Third, the algorithm should allow as many nodes as possible to be off duty in most of the time. At the same time, it should preserve the initial sensing coverage with minimal "sensing hole", or "blind points". It is ideal, if the working nodes can cover the same monitored area as the original one. Fourth, certain redundancy is still needed against node failures, therefore the scheduling scheme should be able to maintain certain sensing reliability.

This chapter is organized as follows. A node activity scheduling scheme involves two problems. The first is how nodes determine their work status. The second is when



(a) $S_{u\to v}$ and $\theta_{v\to u}$ (b) $\phi_{v\to u}$ (c) $S_{v\to u} \bigcup S_{w\to u}$ (d) $S_{\{v,w,x\}\to u} = S(u)$

Figure 4.1: Location_based node-scheduling scheme: basic *off-duty* eligibility model

nodes make such decision. Section 4.1 and section 4.2 address these two problems separately. In section 4.3,we present the experimental and simulation results. In the last section, we summarize this chapter.

4.1 Off-duty Eligibility Model

4.1.1 Basic Model - homogeneous sensing range

As addressed above, the proposed algorithm aims to remove as many as possible redundant sensor nodes, while preserving the original system coverage. In order to achieve this goal, an intuitive approach is to calculate each node's sensing area and then compare it with its neighbors' ones. If a concerned node's sensing area is covered by its neighbors, it can fall asleep safely without any loss of system overall coverage. In this section, we will explain how a node determines that its neighbors can cover its sensing area based on its own and its neighbors' location information.

As mentioned in section 3.3, we assume that each node in wireless sensor networks can perform a 360^o monitoring and its maximal circular observation region is its sensing area. In the basic model, we assume that all sensor nodes have the same sensing range r. The *off-duty* eligibility rule can be expressed as $\bigcup_{v\in SN(u)} S(u\cap v) \supseteq S(u)$. By observation, we know that the auxiliary observable area of node u by node v is a crescent-shaped region. The size of this region is easy to get when the location information of both nodes is given. But direct calculation of AOA offered by multiple sensing neighbors is rather difficult. Some schemes, such as [77] divide AOA into small grids and simplify the problem by investigating multiple grid points, instead of an area. However, the granularity of a grid is proportional to the level of coverage loss and inversely proportional to computation's complexity. In our scheme, we consider another approach.

We observe that AOA by a single sensing neighbor always contains a sector, the shaded area in Figure 4.1(a), if the distance of two nodes is not more than the sensing range r. Although the sector is smaller than the crescent, it is much easier to calculate the size of the sector than that of the crescent, because the area of a sector can be represented by its central angle accurately and the union of multiple sectors can be transformed to merging their central angles, as illustrated in Figure 4.1(c). Therefore, although node v can offer a crescent-shaped auxiliary observable area, to facilitate quick and simple calculation, node u only "admits" that node v can help it to monitor part of AOA if node u is off work. This sector-shaped region is denoted as $S_{v\rightarrow u}$. The size of its central angle (called magnitude of $S_{v\rightarrow u}$) is denoted as $\theta_{v\rightarrow u}$, as shown in Figure 4.1(a). The directional angle of node v referred to node u (called phase of $S_{v\rightarrow u}$) is denoted as $\phi_{v\rightarrow u}$, as illustrated in Figure 4.1(b). According to trigonometry, the central angle and the directional angle are given by

$$\begin{aligned}\theta_{v\rightarrow u} &= 2\times\arccos(\frac{d(u,v)}{2r})\\ &= 2\times\arccos(\frac{\sqrt{(x_u-x_v)^2+(y_u-y_v)^2}}{2r})\end{aligned} \tag{4.1.1}$$

and

$$\phi_{v\rightarrow u} = \arctan(\frac{y_v - y_u}{x_v - x_u}) \tag{4.1.2}$$

.

$\because S(u \cap v) \supseteq S_{v\rightarrow u}$,

we can get the following conclusion that:

if $\bigcup_{v\in SN(u) \text{ and } d(u,v)\leq r} S_{v\rightarrow u} \supseteq S(u)$, then $\bigcup_{v\in SN(u)} S(u \cap v) \supseteq S(u)$

In other words, investigating whether the auxiliary observable area offered by sensing neighbors covers a concerned node's sensing area can be transformed to examining whether the union of the "admitted" auxiliary observable area offered by the sensing neighbors within the sensing range r contains the concerned node's sensing area. The latter, in turn, is equivalent to calculating whether the union of their central angles can cover the whole 360^o as illustrated in Figure 4.1(d). It is easy to merge central angles based on their $\theta_{v\rightarrow u}$ and $\phi_{v\rightarrow u}$. and $\theta_{v\rightarrow u}$ and $\phi_{v\rightarrow u}$ have been formulated in Equations 4.1.1 and 4.1.2. Therefore, our approach only involves simple arithmetic calculation.

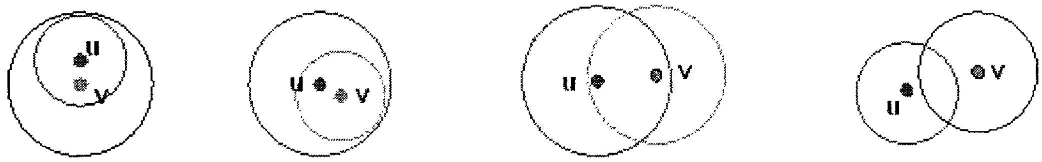
The above model is used to assign nodes a working status: *on-duty* or *off-duty*. Compared to normal *on-duty* mode, *off-duty* is a power-saving sleep mode. It may be implemented by powering off the sensing unit and communication unit of a sensor node, powering off the sensing unit, or just ignoring any event in the surrounding area without powering off any unit. Among the three forms, the first one saves the most energy and the last one does the least. Therefore, the first one is the most desirable from energy conservation's perspective. However, which method is used in practice depends on the intelligence owned by sensors and the cost for sleeping and waking up the sensing unit and communication unit.

If an *off-duty* node has both its sensing unit and communication unit powered off, the network connectivity has to be considered during scheduling as well. When the transmission range is large compared with the sensing range, network connectivity can still be ensured even after many nodes are in inactive status. In chapter 6, we will prove that $R \geq 2r$ is the sufficient condition and tight lower bound to guarantee that connectivity is maintained among active nodes after a coverage-preserving scheduling scheme is executed. However, when the transmission range is smaller than twice of the sensing range, the connectivity of the original network may be ruined after activity scheduling. To prevent it, each node can examine connectivity of its neighbors by using their location information. If two neighbors of a node cannot reach each other directly, the node should keep on duty, even if it satisfies the eligibility rule described above.

4.1.2 Extended Model - heterogeneous sensing range

In the previous section, we assume that each node has the same sensing range r. In this part, we will extend the basic model and provide a solution for the case that nodes have different sensing ranges.

Two reasons may cause different sensing ranges at different nodes. First, nodes have different initial sensing ranges. Second, a node's sensing range changes during its lifetime. For instance, the power level of a sensor node may have an impact on its sensing range and the energy resource is consumed over time. We denote node u's and its neighbor node v's current sensing range as $r(u)$ and $r(v)$, respectively. There are many different cases how sensing adjacent nodes'sensing areas are laid out. For instance, Figure 4.2 presents four of them.



(a) $S(v) \supset S(u)$ (b) $S(v) \subset S(u)$ (c) $(S(v) \bigcap S(u)) \supseteq S_{v \to u}$ (d) $S_{v \to u} \not\subset (S(v) \bigcap S(u))$

Figure 4.2: Location_based node-scheduling scheme: layout of neighboring nodes with different sensing ranges

(a) $S(v) \supset S(u)$ (b) $S(v) \bigcap S(u) \supseteq S_{v \to u}$

Figure 4.3: Location_based node-scheduling scheme: extended model with heterogeneous sensing range

In order to still be able to use central angles to calculate the auxiliary observable area (AOA), we consider two cases shown in Figure 4.3(a-b) in a conservative way.

Case 1: node v's sensing area completely contains node u's sensing area, which happens whenever $r(u) + d(u,v) \leq r(v)$ holds. In this case, node u can fall asleep without further calculation.

Case 2: The sensing areas of both nodes touch at two points, and the intersection area includes a sector centralized at node u. This case happens whenever both $d(u,v) \leq r(v)$ and $r(u) - r(v) \leq d(u,v)$ are true. In this case, the central angle is

$$\theta_{v \to u} = 2 \times \theta = 2 \times \arccos\left(\frac{d(u,v)^2 + r(u)^2 - r(v)^2}{2 \times r(u) \times d(u,v)}\right) \qquad (4.1.3)$$

In summary, when nodes have different sensing ranges, the neighbor set involved into calculation of AOA is modified as

$$\{v \in SN(u) | ((d(u,v) \leq r(v)) \wedge ((r(u) - r(v)) \leq d(u,v))) \vee ((r(u) + d(u,v)) \leq r(v)), v \neq u\} \tag{4.1.4}$$

Obviously, the basic model described previously is a special case of this extension model when $r(u) = r(v) = r$.

4.1.3 Extended Model - exploiting direction information

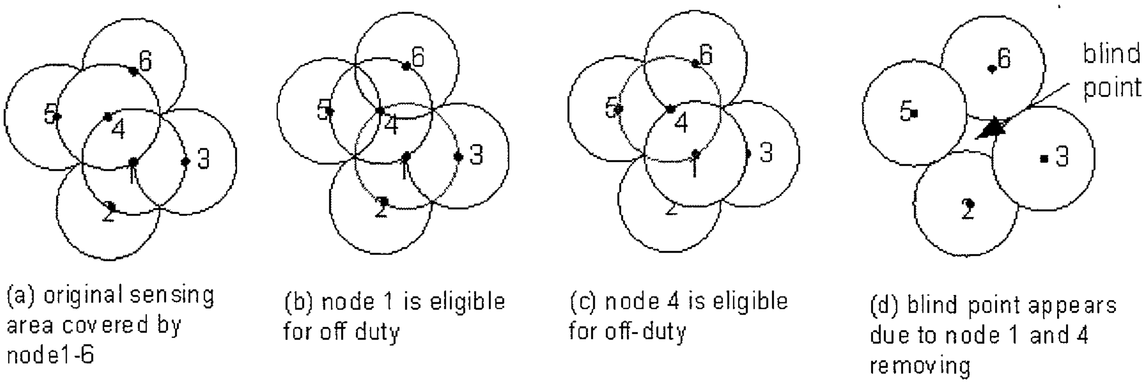
In this section, we will present another extended model that may be feasible in the cases when location information is unavailable for sensor nodes. The restriction of this extended model is that every node has the same sensing range r. As illustrated in the Figure 4.1 (b-d), in order to merge two central angles, we need to calculate their magnitude $\theta_{v \rightarrow u}$ and phase $\phi_{v \rightarrow u}$ first. The formulas of $\theta_{v \rightarrow u}$ and $\phi_{v \rightarrow u}$ are given in expressions 4.1.1 and 4.1.2. Like in the basic model, only those neighbors that are located within a distance of r away (i.e. $0 < d(u,v) \leq r$) are involved into the calculation of the auxiliary observable area (AOA). From the expression 4.1.1, we know that the value range of $\theta_{v \rightarrow u}$ is $120^o \leq \theta_{v \rightarrow u} < 180^o$. If we take the lower bound, i.e. 120^o, as the safe value, it is not necessary to calculate the actual value of $\theta_{v \rightarrow u}$. In other words, if we know the value of $\phi_{v \rightarrow u}$, we can determine the lower bound of the auxiliary observable area (AOA). Techniques to estimate direction $\phi_{v \rightarrow u}$ from incoming signals have already been discussed in the IEEE antennas and propagation community as the Angle-Of Arrival problem. This can be accomplished by using more than one directional antenna, such as addressed in [79]. If the radio communication units in sensor nodes have such direction estimation capability, a node's off-duty eligibility can be determined by simply setting all $\theta_{v \rightarrow u}$ values as 120^o. Obviously, this new extended model does not use any location information.

4.2 Node Scheduling Scheme

In this section, we describe the node-scheduling algorithm based on the eligibility model presented in section 4.1.1.

In our algorithm, the operational timeline is divided into duty cycles. Each duty cycle begins with a self-scheduling phase, followed by a working phase. In the self-scheduling phase, each node investigates its *off-duty* eligibility and determines its sensing status (*off-duty* and *on-duty*) based on it. Non-eligible nodes are responsible for monitoring the environment in the working phase, collecting data from the environment and delivering data reports to the sink node. While eligible nodes become inactive in the working phase and may power off themselves to save energy. To minimize the energy consumption in the self-scheduling phase, the working phase should be long compared to the self-scheduling phase. How *on-duty* nodes collect and deliver data to the sink node is the issue of the data collection and transmission protocols and is out of the scope of our algorithm.

Before self-scheduling is executed, nodes need to collect neighbors' location information. To implement it, a straightforward, simple approach is to let each node broadcast a Position Advertisement Message (PAM), which contains its ID and its current location, at the beginning of each duty cycle. Each neighboring node adds an entry into its neighbor list after receiving a PAM message. To reduce the energy consumption in this step, some technologies can be used if they are available. For instance, because only neighbors within a node's sensing range away are involved into calculation of auxiliary observable area (AOA) according to the model, each node can



(a) original sensing area covered by node1-6

(b) node 1 is eligible for off duty

(c) node 4 is eligible for off-duty

(d) blind point appears due to node 1 and 4 removing

Figure 4.4: Location_based node-scheduling scheme: simultaneous removal

transmit its PAM message with the minimum power as long as the message reaches its sensing range. Thus only those nodes within the sender's sensing range can receive its PAM message.

Of course, such step for neighbor information collection is not mandatory. If we let all the messages piggyback sender's location information, nodes can maintain their neighboring list by overhearing the messages transmitted in the previous duty cycles.

As we mentioned, in self-scheduling phase, each node needs to investigate its *off-duty* eligibility and to determine its work status (*off-duty* and *on-duty*) in the following work phase, based on the model described in the section 4.1.1. However, if all nodes make decision simultaneously, blind points may appear. As illustrated in Figure 4.4, node 1 finds its sensing area can be covered by nodes 2, 3 and 4. According to the *off-duty* eligibility rule, node 1 falls asleep. However, at the same time, node 4 finds its sensing area can be covered by nodes 1, 5 and 6. Believing node 1 is still active, node 4 takes *off-duty* status as well. In the result, the system's sensing coverage is reduced, as shown in Figure 4.4(d). To solve such simultaneous removal problem, we introduce a random back-off delay before each node self-determines its status. Moreover, we let those nodes eligible for *off-duty* broadcast a *Status Advertisement Message (SAM)* to inform their neighbors of status change. Initially, all neighboring nodes are assumed to have a default *on-duty* status. If a node receives a *SAM* message, it will mark the sender as an *off-duty* one and remove that node from its active neighbor list. The nodes, which have a longer back-off delay, will not consider those neighbors that have marked *off-duty* before. Thus, as long as node 1 and node 4 select different back-off delay, the blind point shown in Figure 4.4(d) can be avoided. However, there is still a chance that node 1 and node 4 may delay the same time. To preserve system coverage further, we let each *off-duty* candidate wait for a short time period after sending *SAM* out (instead of changing its status to *off-duty* immediately). If a *SAM* is received during this "ready-to-off" period, the candidate will re-investigate its *off-duty* eligibility. If the eligibility doesn't hold any more, the candidate returns its status to *on-duty*. Otherwise, the candidate sets its status as *off-duty* after timeout. Note that the nodes, which have decided to serve as *on-duty* ones, won't re-evaluate their eligibility once the decision has been made. For instance, in Figure 4.4, node 1 and node 4 select the same random delay. Node 1 finds it is eligible for *off-duty*, so it broadcasts a *SAM* to its neighbors. The same thing happens at node 4. Thus both nodes 1 and 4 are ready to be off duty and wait to enter *off-duty* status. Before timeout, the *SAM* message sent by node 1 is received by node 4. Node 4 finds that it can't fall asleep without node 1's aid. So node 4 sets its status as *on-duty*. So does node 1. Finally, both of them have an active status. This result is not perfect, since either node 1 or node 4 (not both) falling asleep won't degrade the sensing area. The *off-duty* node number is not minimized. However, to avoid process repeating and extra traffic, we do not let a node send a *SAM* if its status changes from *ready-to-off*

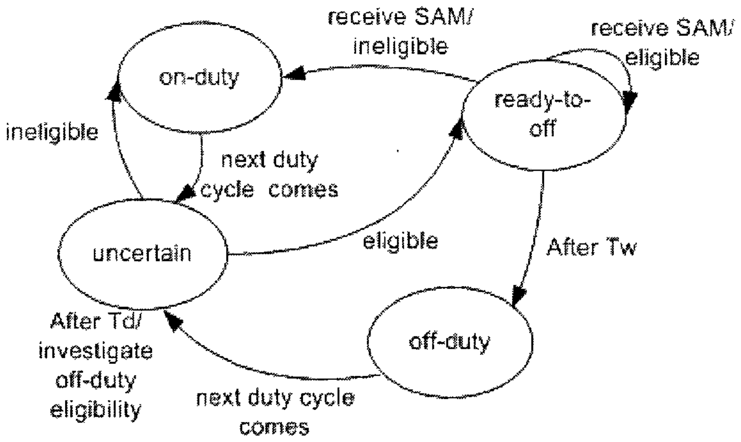


Figure 4.5: Location_based node-scheduling scheme: FSM

to *on-duty*. The status transition graph of this step is presented in Figure 4.5.

How each node performs self-scheduling is formally described below:

1. Generate a random back-off time T_d.
2. During T_d, listen to the channel. If receiving a *Status Advertisement Message (SAM)*, delete the sender from the active neighbor list.
3. After time T_d, investigate *off-duty* eligibility. If ineligible, set status as *on-duty* and enter step 6. Otherwise, set status as *ready-to-off*, broadcast a *Status Advertisement Message* and perform step 4.
4. Wait for a short time T_w. During T_w, listen to the channel. If receiving a *Status Advertisement Message (SAM)*, delete the sender from the active neighbor list and re-investigate the *off-duty* eligibility. If ineligible, set status as *on-duty* and enter step 6. Otherwise, continue to perform step 4.
5. After T_w, set the status as *off-duty* and fall asleep. Perform step 6.

6. Self-scheduling has been done. Wait to enter working phase

Besides the random backoff delay described above, network topology or energy factor can be considered in the derivation of the backoff delay. In non-uniform network topologies, nodes have different neighbor number. The nodes with more neighbors have a higher probability to be identified as a redundant node. However, if they evaluate their *off-duty* eligibility too early, their neighbors that do it later will have less chance to become *off-duty* nodes. Therefore, we can let those nodes having more neighbors take a longer backoff delay. In this way, more off-duty nodes can be identified in the final. The experimental result verifies that setting delay in the ascending order of neighbor number generates more *off-duty* nodes than using a random delay. Another consideration is that of unequal energy left at each node. Since the *on-duty* nodes consume more energy than off-duty ones, to balance energy load, those nodes with less energy resource should be more reluctant to keep *on-duty* and therefore should select a shorter backoff delay (because nodes with a longer delay will have less opportunity to escape from working).

4.3 Performance Evaluation

In this section, we present some experimental and simulation results as performance evaluation of the proposed scheme. We divide the evaluation into two parts. The first is to investigate the capability of *off-duty* eligibility model in terms of maintaining system original sensing quality. The second is to study its efficiency in terms of energy savings.

4.3.1 Performance Evaluation of Eligibility Model

To evaluate the validity of the off-duty eligibility model, we designed and developed a simplified simulator in Java, as described below. It does not introduce any lower layer protocols and assumes that the high layer at two neighbors can exchange messages directly, i.e. it assumes perfect wireless communication and does not account for any communication overhead. Such simulator shortens the time for protocol implementation, enables us to quickly get experimental data through a large number of simulation setting. The most important, it can provide similar coverage performance results to ns-2 [1], when used to evaluate coverage properties.

In the experiments, we deploy N nodes to random positions within a fixed square space ($50m \times 50m$). N can be 50 to 300 with an increment of 50. Each node has a sensing range of r meters and a communication range as $r \times \kappa$ meters. Also, each node has already known neighbor position information. During the scheduling, we let all the nodes decide their work status in a random sequence. The decision of each node is visible to all the other nodes. In other words, the nodes, which make decisions later, know the nodes that have taken *off-duty* status before and thereby will not consider those nodes in their turns. After all the nodes have made decisions, the number of the off-duty nodes is counted and then is divided by the initial deployed node number N to get the *off-duty percentage* (one of metrics we are interested in). Another metric measured is *coverage loss*. To calculate the metric, we divide the space into $0.1m \times 0.1m$ unit cells. We assume an event occurs in each cell, with the event source located at the center of the cell. We count how many nodes can detect each event. The system original sensing coverage consists of all the cells containing an event that can be observed by at least one node. If an event cannot be detected by

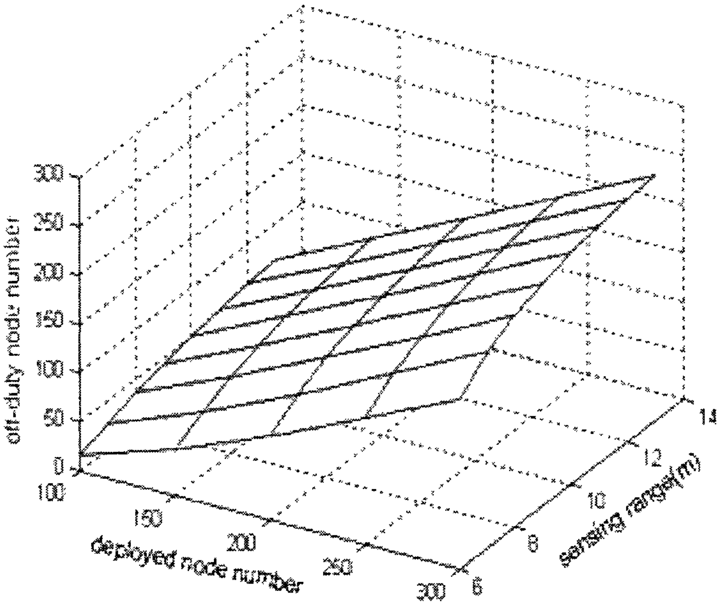


Figure 4.6: Location_based node-scheduling scheme: off_duty node number vs. node density

any *on-duty* node, but is within the range of the original sensing coverage, we call the event source cell a "blind point". The coverage percentage loss is calculated as the ratio of the number of blind points to the original sensing coverage. We also compute the *average sensing degree* as the sensing degree sum of all the events divided by the total event number. Here, sensing degree is defined as the number of active nodes that can simultaneously detect and report a common event.

All tables and graphs presented in this section represent the mean values from 100 random network topologies with $\kappa \geq 2$.

Figure 4.6 shows a 3D surface plot of the off-duty node number in different sensing range and deployed node number. As we can see, when the original deployed node number and the sensing range increase, more *off-duty* nodes are generated. This is consistent with our expectation. However, as illustrated in Figure 4.7, *on-duty* node

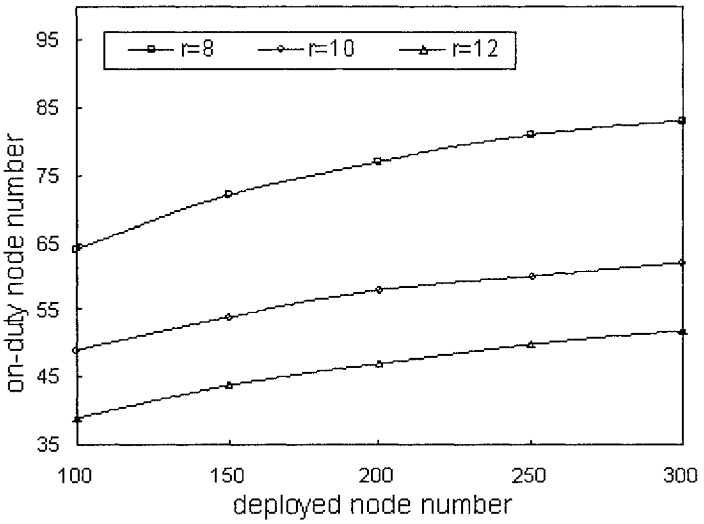


Figure 4.7: Location_based node-scheduling scheme: on_duty node number vs. node density

number does not remain constant over different deployed node number when the sensing range and the deployed area are fixed. Instead, it increases with the deployed node number. There are two reasons contributing to this effect. First, the more nodes the network has, the more nodes are located at the boundary of the deployed area. According to our *off-duty* eligibility model, edge nodes have no chance to be *off-duty* because all the other nodes are located on one side of them. Second, the increase of redundant nodes inside the deployed area is not always proportional to the increase of deployed nodes, because some new added nodes are used to fill sensing holes that originally exist in the network due to insufficient node density. In spite of this, the experimental result shows that the *off-duty* eligibility model effectively restricts the number of *on-duty* nodes. When the deployed nodes' number is increased from 100 to 300, the on-duty node number just increases about 30%.

Next, we investigate the change of average sensing degree over node density. As

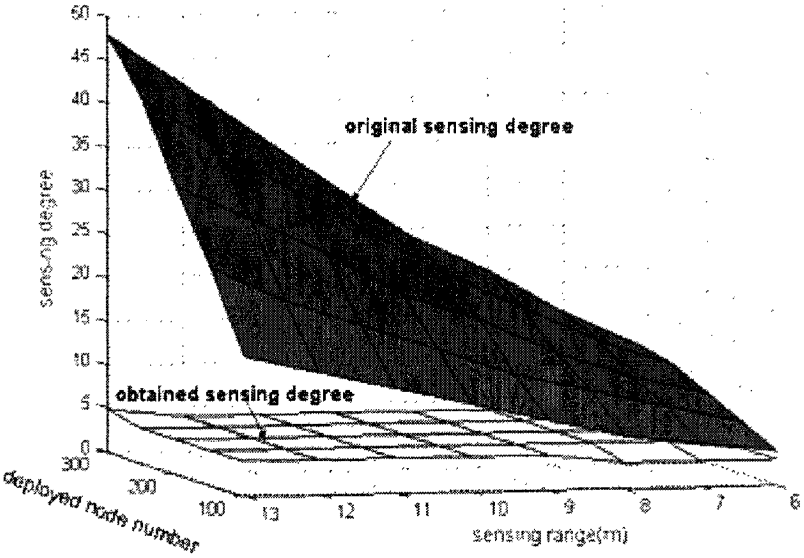


Figure 4.8: Location_based node-scheduling scheme: sensing degree vs. node density

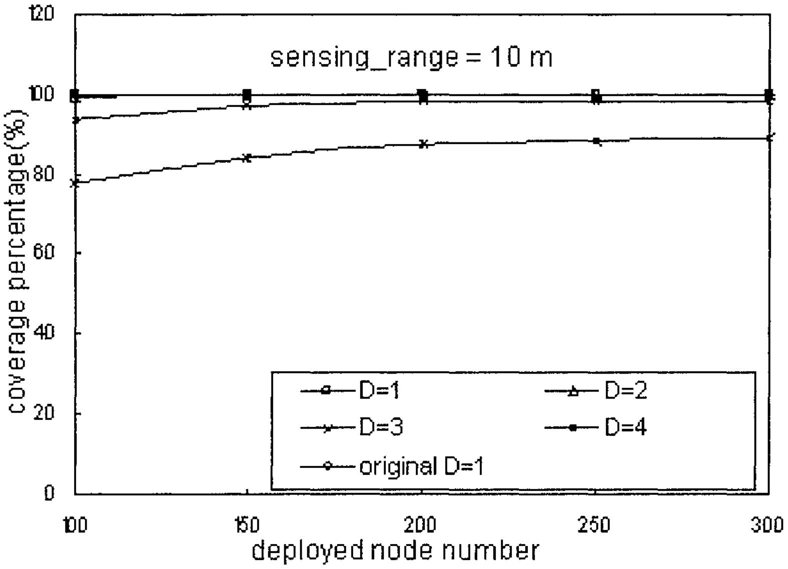


Figure 4.9: Location_based node-scheduling scheme: coverage percentage vs. node density

shown in Figure 4.8, although the range of the initial sensing degree is varied from 3 to 48, the obtained sensing degree is stable at 3 or 4 in most test cases. Obviously, the *off-duty* eligibility model can effectively control the network redundancy, at the same time remaining a level of sensing reliability. Figure 4.9 presents the same effectiveness as in Figure 4.8, but from the different view: the percentage of the deployed area that can be monitored by at least D *on-duty* nodes. Specially, it implies the capability of the model in preserving the original sensing coverage. We still divide the space into $1m \times 1m$ unit cells as mentioned before. An event occurs in each cell, with the event source located at the center of the cell. We investigate the ratio of the cell number reached by at least D *on-duty* nodes to the total number of cells when sensing range is 10 meters. As we can see, most of the area, above 93%, can be covered by at least 3 *on-duty* nodes. Almost 100% cells can be reached by at least one *on-duty* node. And about 99% cells can be monitored by at least 2 *on-duty* nodes. Furthermore, the two curves ($D = 1$, original $D = 1$) are exactly the same in the figure. When we change the sensing range to 8 meters and 12 meters, the same behavior is observed by us. This implies that the *off-duty* eligibility model has the capability of completely preserving the original sensing coverage.

The results presented above are got from an ideal running environment (i.e. there is no location error, packet loss and node failure in the network).

Practically, location errors may be caused by either imprecise measurement from GPS or inaccurate estimation from a localization system. To investigate the sensitivity of our scheme to location error, we artificially introduce location errors at each node. We modeled the error by randomly recoding the location of each node in the range $[x - e,\ x + e]$ and $[y - e,\ y + e]$. Simulation results show that our scheme is

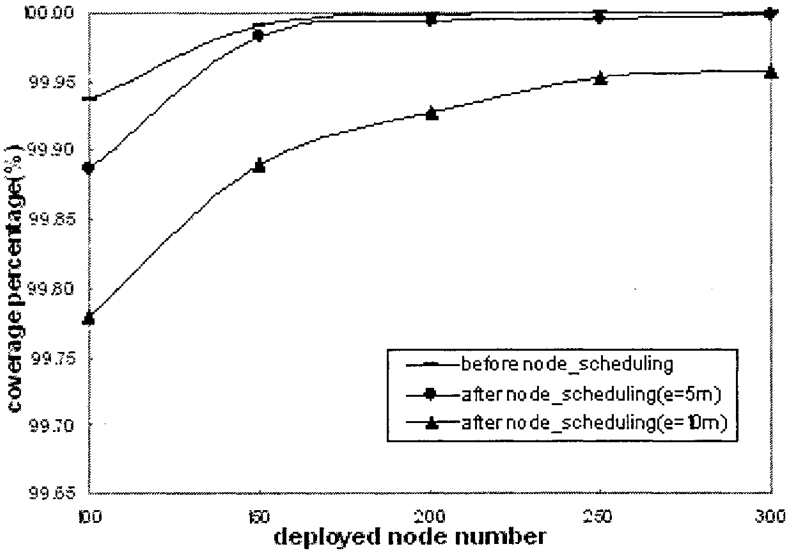


Figure 4.10: Location_based node-scheduling scheme: sensitivity to location error($\mathring{A} = 50m \times 50m, r = 10m$)

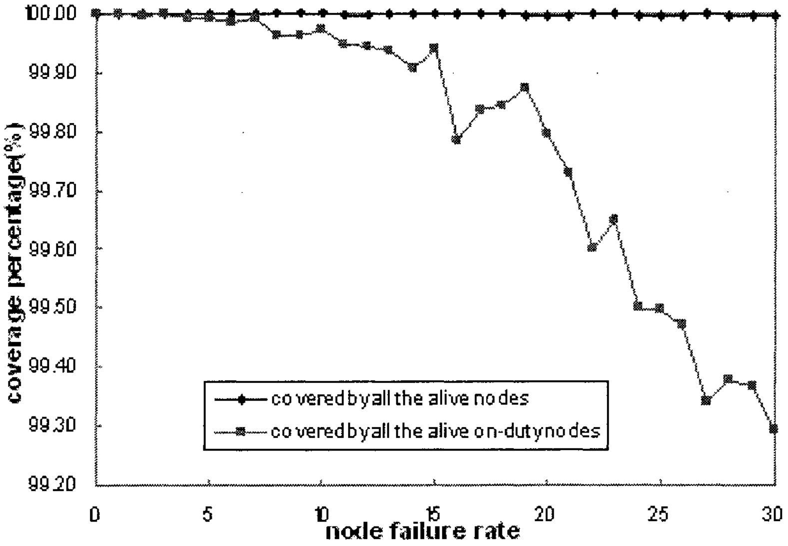


Figure 4.11: Location_based node-scheduling scheme: sensitivity to node failure($\mathring{A} = 50m \times 50m, r = 10m$)

not sensitive to location errors. For example, Figure 4.10 plots the sensing coverage changes (original $D = 1$ and $D = 1$) when e is set as 5 meters or 10 meters. When e is 5 meters, the maximal loss percentage of sensing coverage is less than 0.051%. If we introduce a randomized error [-10m, +10m] in each node's x- and y- coordinates, the simulation results show 0.16% loss. The reason of such insensitivity is that there is still enough redundancy after performing our node-scheduling algorithm. Although inaccurate position information "shifts" a single node's sensing area during calculation, the overall sensing coverage will not be affected too much, because the empty area, which was expected to be covered by this single node, can be filled up by "shifting" of other *on-duty* nodes. Another observation in Figure 4.10 is that the loss decreases as more nodes are initially deployed. That is because more *on-duty* nodes are generated when the initially deployed node number increases.

In wireless sensor networks, sensors are prone to failure. Our node-scheduling algorithm requires nodes to maintain their neighbor information periodically. Therefore, the impact of node failure on system coverage will not last for too long. The more frequently the neighbor information update is performed, the shorter the impact lasts. To investigate how much node failure affects system coverage before next neighbor information update, we set all sensor nodes as "alive" initially. After node scheduling is executed, we randomly select some nodes from all deployed nodes and mark them as "dead". The ratio of node failure rate is changed from 0% to 30% with an increment of 1%. Then we measure the sensing coverage covered by all alive nodes and by all alive *on-duty* nodes, respectively. We find that although the degradation increases as the node failure rate increases, it is still confined within an acceptable range. As shown in Figure 4.11, in the networks with 200 deployed nodes and 10m

Table 4.1: Location_based node-scheduling scheme: result of sensitivity to packet loss rate($|\aleph| = 200$ nodes, $\mathring{A} = 50m \times 50m, r = 10m$)

Packet loss rate(%)	Original coverage percentage (original D=1)	Coverage percentage (D=1)	*off-duty* node rate(%)	Obtained sensing degree
0	100	100	71.5	4
5	100	100	71.5	4
10	100	99.99	71.5	4
15	100	99.97	71.5	4
20	100	99.86	71	4
25	100	99.74	70.5	4
30	100	99.69	69	4

sensing range, the average degradation is only 0.7% when the node failure rate is 30%. The sensing degree drops a little, from 4(when node failure rate is 0%) to 3 (when node failure rate is 30%).

In the next experiment, we investigate the sensitivity of the algorithm to packet loss. We introduce a packet loss rate in each transmission of both *PAM* and *SAM* messages. Table 4.1 lists the results corresponding to different packet loss rates. From the table, we can see that the sensing coverage does not decrease dramatically as the packet loss rate increases. When the packet loss rate is 30%, the sensing coverage reduction is 0.31%. Furthermore, the *off-duty* node number and obtained sensing degree remain almost unchanged against packet loss. Such observation is the combination effect of two counter-forces. On one hand, loss of *PAM* messages may make some nodes invisible to their neighbors. With incomplete neighbor list, some nodes eligible for *off-duty* may make *on-duty* decisions. On the other hand, loss of SAM messages isolates some *off-duty* nodes from their neighbors and causes them

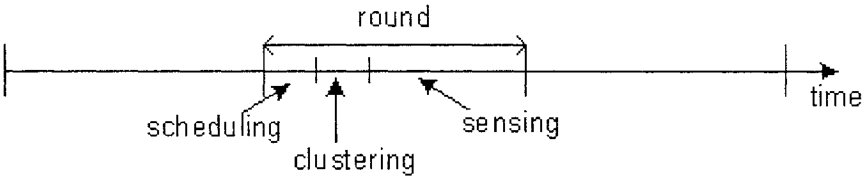


Figure 4.12: Location_based node-scheduling scheme: timeline of LEACH with extension

make wrong *off-duty* decisions.

4.3.2 Performance Evaluation of Node Scheduling Scheme

We implement the complete scheme as an extension of an existing data gathering protocol, LEACH [31] in ns-2 [1]. Although the proposed scheme can be combined with any other data collection approaches, we select LEACH because it has a similar timeline as the proposed algorithm.

The operation of LEACH is divided into rounds. Each round is composed of a cluster set-up phase (when the clusters are formed), and a steady-state phase (when sensors collect data from the environment and transfer data reports to the cluster-heads and then to the base station). To extend LEACH with our node-scheduling algorithm, a straightforward way is to insert a self-scheduling phase before LEACH's cluster set-up phase. More specifically, at the beginning of each round, all sensor nodes determine their work status (*off-duty* or *on-duty*). Off-duty nodes fall asleep and don't participate in the cluster forming and data collection followed. Such extension approach enables seamless embedding of our node-scheduling algorithm into LEACH without any modification of LEACH's original workflow. The timeline of the implementation is illustrated in Figure 4.12.

The simulation is carried out in a network with 100 nodes, each with a sensing range of 10 meters. Nodes are placed randomly in a rectangular region which area is $50m \times 50m$. The remote base station (or sink node) is located at the low left corner, i.e. origin point (0,0). The initial energy of all nodes is $2J$. Each sensor sends a 2000-bit report message to the base station with a 0.5 time interval. The time duration of each round is 10 seconds. That means, the number of data collections per round, denoted as N_g, is $10s/0.5s = 20$. The simulation uses the same energy parameters and radio model as discussed in [31]. Among all energy consumptions for processing, we only consider that for data aggregation, i.e. 5nJ/bit/signal. Off-duty nodes don't get involved into cluster forming. In addition, they do not generate, send and receive data reports. So, the energy consumed by them after self-scheduling phase is negligible. Note that we let all the nodes broadcast their position information and update their neighbor lists at the beginning of each duty cycle, although the period of neighbor information maintenance can be longer than that of node-scheduling or a passive,economic neighbor information exchange approach may be used instead, as we have explained in section 4.2. Finally, all graphs presented in this section represent the mean values from 5 random network topologies with $\kappa \geq 2$.

Figure 4.13 illustrates the energy dissipation curve per node in the original LEACH and the extended LEACH in a random network topology. As we can see, the energy dissipation in the extended LEACH is slower than the original one. Figures 4.14 and 4.15 show an increase of the system lifetime, in the same parameter setting. Here, we use two metrics to evaluate the system lifetime: the total number of nodes alive over time and the system overall sensing coverage over time (the ratio of the area monitored by *on-duty* nodes to the deployed region). As illustrated in Figure 4.14

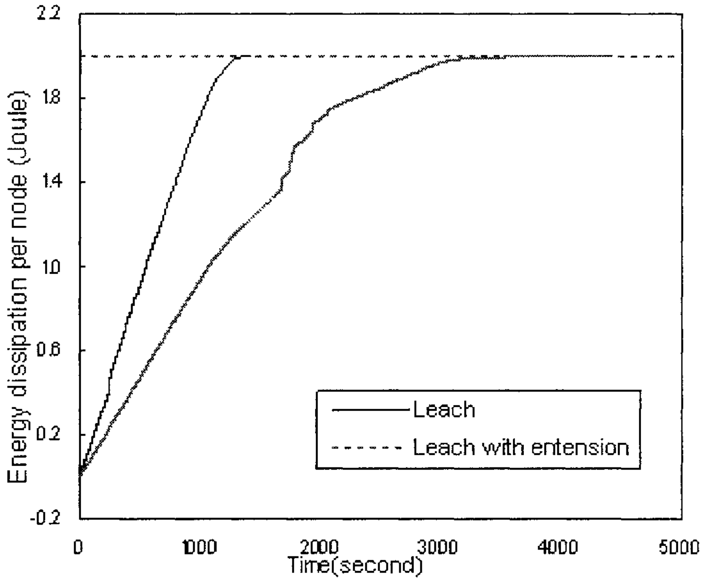


Figure 4.13: Location_based node-scheduling scheme: average energy dissipation per node over time ($|\aleph| = 100, \mathring{A} = 50m \times 50m, r = 10m, N_g = 20$)

and 4.15, although the extended LEACH does not outperform the original one in term of first node dead time, the number of nodes alive and the system sensing coverage drop more slowly in the extended LEACH than in the original one. In the result, it takes approximately 4378 seconds for the last node to die in the extended LEACH, while 1412 seconds in the original LEACH. And it takes approximately 2055 seconds for the sensing coverage to drop 20% (reach 80%) in the extended LEACH, while 1285 seconds in the original one.

We also change the number of data collections in each duty cycle from 4 to 20 with the increment of 4, and compare the system lifetime (represented by the time when system coverage drops below 80%) of the original and extended LEACH. Figure 4.16 shows that the increase of system lifetime in the extended LEACH has no dramatic change over N_g.

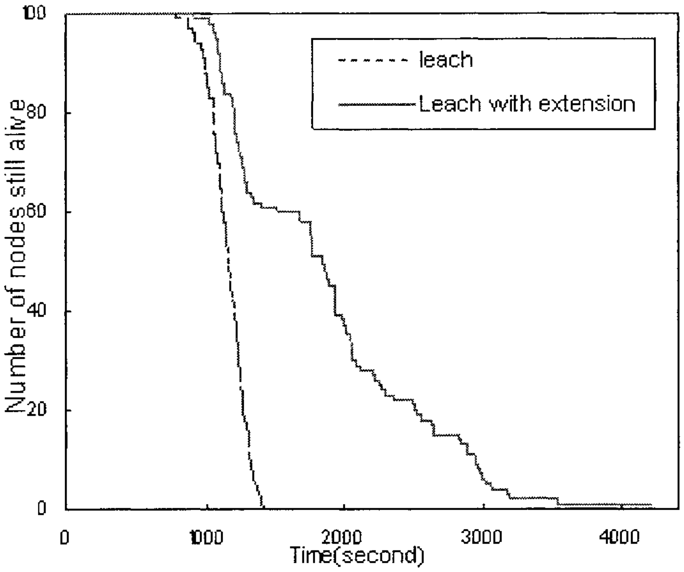


Figure 4.14: Location_based node-scheduling scheme: number of nodes alive over time ($|\aleph| = 100, \mathring{A} = 50m \times 50m, r = 10m, N_g = 20$)

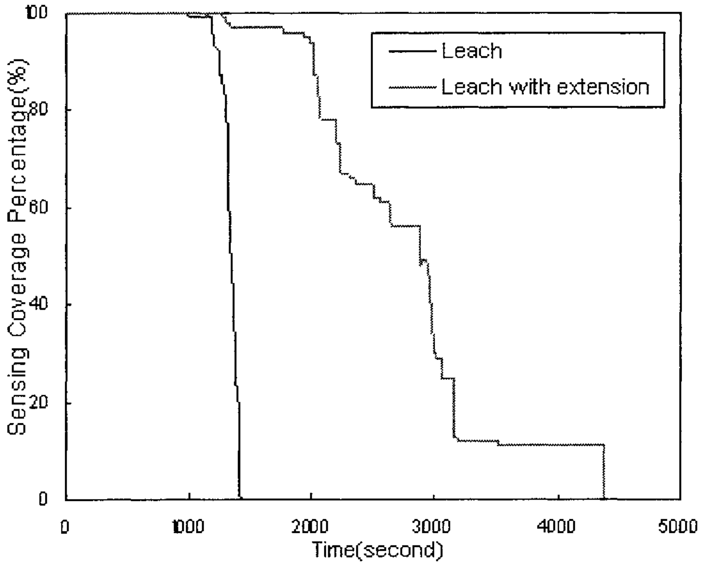


Figure 4.15: Location_based node-scheduling scheme: sensing coverage percentage over time ($|\aleph| = 100, \mathring{A} = 50m \times 50m, r = 10m, N_g = 20$)

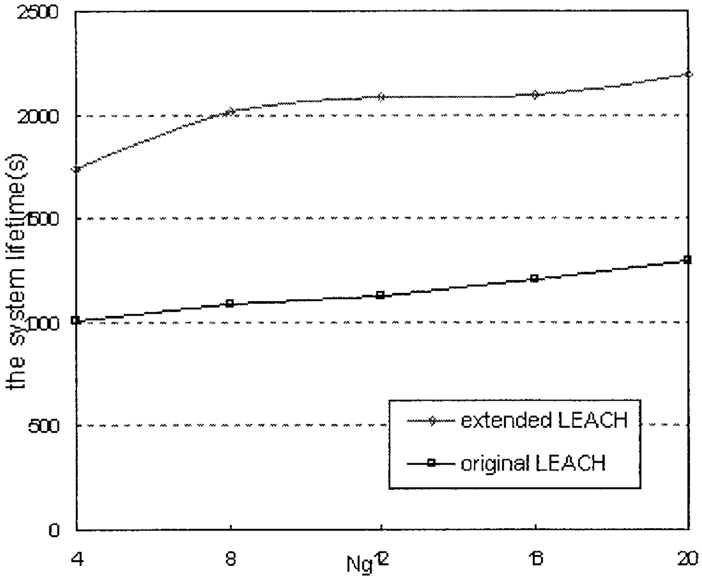


Figure 4.16: Location_based node-scheduling scheme: system lifetime(coverage capability) over N_g ($|\aleph| = 100, \mathring{A} = 50m \times 50m, r = 10m, N_g = 20$)

Energy saving analysis

Next, we simply analyze the factors that affect energy saving in the extended LEACH.

Figure 4.17 shows how the average energy dissipation per node during the first 100-s simulation time(i.e. no node has drained out of its energy resources) changes over different deployed node number. From it, we find that decreasing of the average energy consumption falls too slowly compared with the increasing of the deployed node number. This result is counterintuitive because, although the increase of the deployed node number leads to that of the on-duty node number, the change is not dramatic, as illustrated in Figure 4.7. To understand the reason, we make a simple mathematical analysis. We define the energy saving factor as

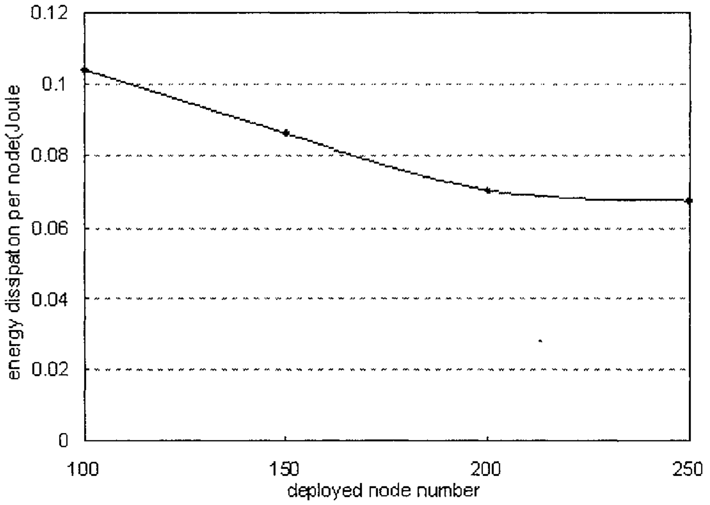


Figure 4.17: Location_based node-scheduling scheme: energy dissipation per node vs. node density($\mathring{A} = 50m \times 50m, r = 10m, N_g = 20$)

$$\alpha = \frac{\frac{E_{on}}{N_{on}} - \frac{E'_{on+off}}{N_{on+off}}}{\frac{E_{on}}{N_{on}}}$$

where N_{on+off} is the deployed node number, E'_{on+off} is the overall energy consumed by N_{on+off} nodes per duty cycle in the extended LEACH. After node-scheduling, N_{on} nodes are marked as on-duty ones and participate in the following operations(clustering-forming and environment-monitoring). E_{on} is the overall energy consumption per duty cycle if these N_{on} nodes are performing the original LEACH. We define the redundancy factor η as N_{on+off}/N_{on}. Ideally, E'_{on+off} equals to E_{on} if energy consumption in node-scheduling is negligible. Thus we have that $\alpha_{ideal} = 1 - \frac{1}{\eta}$. However, when $E'_{on+off} = E_{on} + E'(s)$, with $E'(s)$ denoted as the energy consumption in scheduling, we have the energy saving factor as

$$\alpha = 1 - \frac{1}{\eta} - \frac{E'(s)}{\eta E_{on}}$$

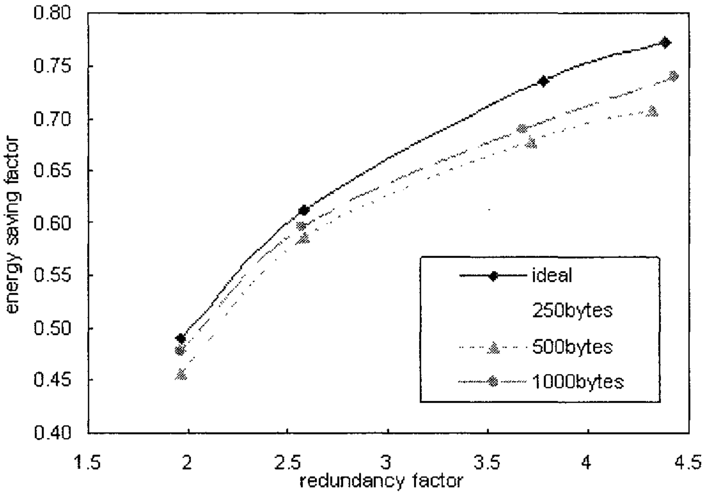


Figure 4.18: Location_based node-scheduling scheme: energy saving factor vs. redundancy factor($|\aleph| = 100 - 250, \mathring{A} = 50m \times 50m, r = 10m$)

which is smaller than α_{ideal} by $\frac{E'(s)}{\eta E_{on}}$. When the ratio of energy consumption in scheduling to that in clustering-forming and data-gathering decreases, the difference between α and α_{ideal} decreases. In the simulation, we increase the size of report message from 2000 bits to 8000 bits, i.e. increasing E_{on} correspondingly and maintaining the same $E'(s)$. As shown in Figure 4.18, the increase of the report size leads to the decrease of the difference α and α_{ideal}.

In addition, we know that the main energy dissipated in $E'(s)$ s is from transmission and reception of PAM and SAM. In the simulation, each node transmits PAM message once and N_{off} nodes transmit SAM message once per duty cycle. Each PAM or SAM is received by approximate n nodes, where n is the average neighbor number. Therefore

$$\begin{aligned} \alpha &= 1 - \frac{1}{\eta} - \frac{E'(s)}{\eta E_{on}} \approx 1 - \frac{1}{\eta} - \frac{N_{on}((N_{on+off} + N_{on})E_{Tx} + n(N_{on+off} + N_{on})E_{Rx})}{N_{on+off}E_{on}} \\ &= 1 - \frac{1}{\eta} - \frac{N_{on}E_{Tx}(1+\eta)}{\eta E_{on}} - \frac{nN_{on}E_{Rx}(1+\eta)}{\eta E_{on}} \end{aligned} \tag{4.3.1}$$

where, E_{Tx} is the energy consumed from transmitting a PAM or SAM message, E_{Rx} is the energy for receiving it. Equation 4.3.1 tells us that with fixed deployed area, sensing range and energy parameters, if we increase the deployed node number, n is the only coefficient in equation 4.3.1 which increases significantly. Therefore, the more nodes we have, the larger the difference α and α_{ideal}. This explains why the change speed of energy consumption is not as quick as the that of the redundancy factor in Figure 4.17. Equation 4.3.1 also tells us that to further energy saving, energy consumption for information exchange during scheduling must be reduced by increasing the frequency of neighbor position maintenance or using passive information obtain approach or allowing some coverage loss.

4.4 Summary

In this chapter, we proposed a coverage-preserving node-scheduling scheme, which can reduce energy consumption, therefore increase system lifetime, by identifying redundant nodes in the sensing domain. We presented a basic model for coverage-based off-duty eligibility rule and then extended it for the cases when nodes have different sensing ranges and directional information is available to be used. This kind of off-duty eligibility rule guarantees that the original sensing coverage can be completely preserved in an ideal simulation environment. And sensing coverage degradation due to location error, packet loss or node failure is not more than 1% . Experimental results also show that certain redundancy still remains after node-scheduling. We design a back-off based node scheduling algorithm to solve simultaneous removing problem. We implemented this node-scheduling scheme as an extension to the famous LEACH protocol. We compared the energy consumption in both LEACH and analyzed the effectiveness of our scheme in terms of energy saving. Preliminary simulation results in the radio model and energy parameters proposed by the LEACH designer show noticeable energy saving and system lifetime increase.

Chapter 5

Location-free Node-scheduling Schemes

In a previous chapter, we proposed a node-scheduling scheme, which aims to completely preserve system sensing coverage without any loss. However, the scheme assumes that each node has knowledge of its own and its neighbors'location information. Based on such location information, each node evaluates its off-duty eligibility to determine its working status. In this chapter, we propose and study several alternative node-scheduling schemes, which may not completely preserve the original system coverage, but are location-free, flexible and light-weighted.

The rest of this chapter is organized as follows. section 5.1 explains the off-duty eligibility models. In section 5.2, we describe how these schemes operate. In section 5.3, we present our experimental results to compare their performance. Finally, we give a summary and conclude the chapter.

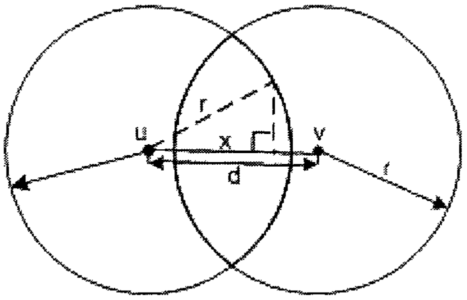


Figure 5.1: Nearest-neighbor-based off-duty eligibility model: intersection of two nodes' sensing areas

5.1 Off-duty Eligibility Model

5.1.1 Nearest-Neighbor-based Off-duty Eligibility Model

Consider the simple scenario in Figure 5.1, where node u has a neighbor v. As described in the previous chapter, $S(u)$ and $S(v)$ denote the circle sensing areas covered by node u and v, respectively. The part of node u's sensing area that can also be covered by node v is the shaded region, denoted as $S(u \cap v)$. The radii of both $S(u)$ and $S(v)$ are r. d is the distance between nodes u and v.

It is easy to derive that the area of $S(u \cap v)$ equals to

$$|S(u \cap v)| = INTC(d,r) = \begin{cases} 4\int_{d/2}^{r} \sqrt{r^2 - x^2}dx & \text{,for all } d \leq 2r, \\ 0 & \text{,otherwise.} \end{cases} \tag{5.1.1}$$

where $INTC(d,r)$ is the intersection area of the two circles with the radii as r and centered at two center points distanced by d. When $d = 0$, the area $|S(u \cap v)|$ is the largest, which equals to πr^2. When $d \geq 2r$, the area $|S(u \cap v)|$ is the smallest, which equals to 0. Resolving the integral in Equation 5.1.1, $|S(u \cap v)|$ can be expressed as:

$$|S(u \cap v)| = INTC(d,r) = \begin{cases} \pi r^2 \times [(2\arccos \frac{d}{2r} - \frac{d}{\pi r}\sqrt{1 - \frac{d^2}{4r^2}})] & \text{,for all } d \leq 2r, \\ 0 & \text{,otherwise.} \end{cases} \tag{5.1.2}$$

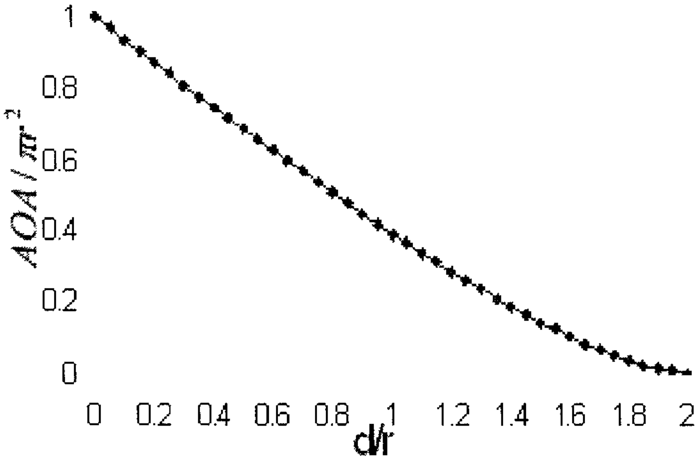


Figure 5.2: Nearest-neighbor-based off-duty eligibility model: relationship of intersection area and relative distance between two nodes

Figure 5.2 graphically represents the relationship between the intersection area of two circles and the relative distance of their center points. As mentioned before, in the context of this thesis, we name the intersection area as "auxiliary observable area (AOA)", because it is within active node v's sensing area and thereby can still be observable if node u is off work . Correspondingly, the leftover area $S(u) - S(u \cap v)$ is named as "auxiliary unobservable area (AUA)". As can be seen, when the distance of node u and v is equal to half of their sensing range r, the auxiliary observable area is about 68.5% of node u's sensing area. That implies that if node u is aware that one of its neighbors is active and within $r/2$ distance away, it can deduce that the loss of its sensing area caused by its leaving will not be more than 31.5%.

Such relationship can be used in node scheduling, if nodes have knowledge of the relative distance to their neighbors. Intuitively, if the maximal percentage loss of sensing area at each individual node, denoted as AUA_{max}, is selected, we can determine the maximal distance D between two nodes to guarantee this threshold.

During node scheduling, a node examines if the distance to its nearest neighbor, denoted as d_{min}, is less than or equal to the value of D. If affirmative, the node will set its status as *off-duty*. For instance, if the threshold AUA_{max} is set as 20%, the corresponding D is $0.315r$. So a node can be *off-duty*, if the distance to its nearest neighbor is not more than $0.315r$. If the threshold AUA_{max} is set as 10%, the corresponding D is $0.157r$. The value of D can be solved in two ways. One is direct calculation by using numerical method. The other is to retrieve from a table storing data-pair in Figure 5.2.

Note that the threshold AUA_{max} is a restriction of the area loss observed by an individual node. This does not mean that the percentage loss of the whole system coverage is equal to this value. In fact, the loss of the whole system coverage is always less than the threshold, because the losses at individual nodes are overlapped with each other. This point has been verified through our experimental results. It is an open problem to determine the accurate or approximate relationship between the percentage loss at individual nodes and that of the whole system.

5.1.2 Neighbor-Number-based Off-duty Eligibility Model

Next, we would like to know the average value of $|S(u \cap v)|$ if a node u is unaware how far a neighbor v is away from it. Supposing that node v can randomly locate in any place within node u's transmission range R, the average value can be obtained by integrating the value of $INTC(d, r)$ over the circle of radius y centered at node u for y in $[0, R]$:

$$INTC(d,r)_{avg} = \int_0^{2r} \frac{2\pi y[INTC(y,r)]}{\pi R^2} dy, R \geq 2r \qquad (5.1.3)$$

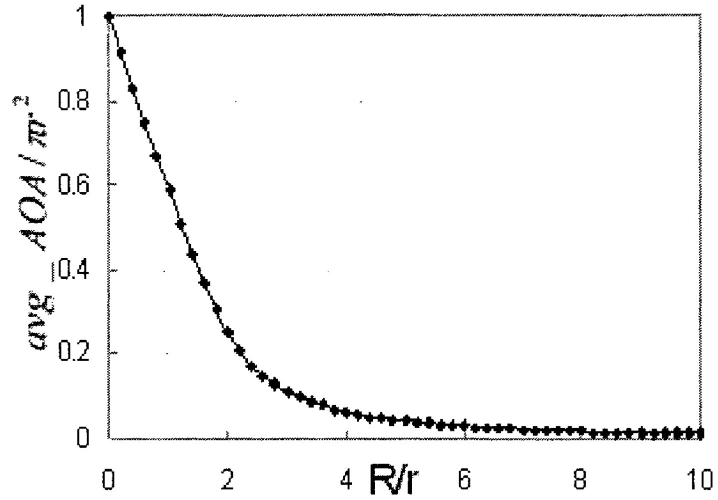


Figure 5.3: Neighbor-number-based off-duty eligibility model: relationship of average auxiliary observable area offered by one neighbor and the ratio of transmission range to sensing range

$$INTC(d,r)_{avg} = \int_0^R \frac{2\pi y[INTC(y,r)]}{\pi R^2} dy, R < 2r \tag{5.1.4}$$

Substituting Equation 5.1.2 for $INTC(d,r)$ in Equation 5.1.3, we get the average value of $|S(u \cap v)|$ when $R = 2r$ as follows:

$$\begin{aligned} &\int_0^{2r} \frac{2\pi y(2r^2 \arccos \frac{y}{2r} - y\sqrt{r^2 - \frac{y^2}{4}})}{4\pi r^2} dy \\ &= \int_0^{2r} y \arccos \frac{y}{2r} dy - \int_0^{2r} \frac{y^2}{2r^2}\sqrt{r^2 - \frac{y^2}{4}} dy \end{aligned} \tag{5.1.5}$$

Using the composite Simpson's rule (numerical integration method) to approximate the value of expression 5.1.5, we obtain the average value of $|S(u \cap v)| \approx 0.25\pi r^2$. This means that the average auxiliary observable area (denoted as *avg_AOA*) offered by one neighbor is only 25% of the concerned node's sensing area, when the transmission range R is double the sensing range r. Similarly, we calculate the value of

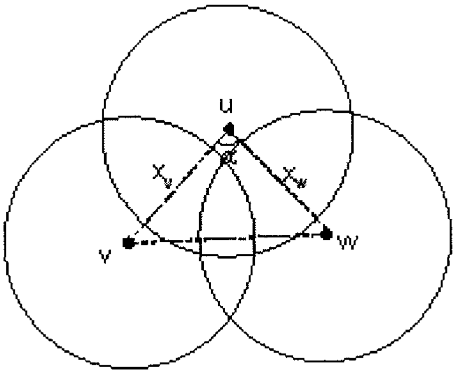


Figure 5.4: Neighbor-number-based off-duty eligibility model: auxiliary observable area offered by two neighbors of node u

avg_AOA at any ratio of R to r by substituting expression 5.1.2 into expression 5.1.3 or 5.1.4. The curve obtained is shown in Figure 5.3.

Now, let us look at the scenario when u has two neighbors v and w. Let x_v be the distance between node u and v. Let x_w be the distance between node u and w. The angle, which segment $\overline{uv}$ makes with segment $\overline{uw}$ counter-clockwise, is α as shown in Figure 5.4.

The part of u's sensing area that can be covered by node v or node w is the shaded region:

$$|S(u \cap v) \cup S(u \cap w)| = |S(u \cap v)| + |S(u \cap w)| - |S(u \cap v \cap w)| \tag{5.1.6}$$

where $|S(u \cap v \cap w)|$ is a function of α, x_v , x_w and r, denoted as $g(\alpha, x_v, x_w, r)$.

Also, we would like to know the average value of $|S(u \cap v) \cup S(u \cap w)|$. Supposing that neighbors v and w can randomly locate in any place within node u's transmission range R, and the angle α can be randomly selected from the range $[0, 360^o]$, the average value can be obtained by integrating the expression 5.1.6 as:

$$\int_0^{2r} \frac{2\pi x_v}{\pi R^2}\left(\int_0^{2r} \frac{2\pi x_w}{\pi R^2}\left(\int_0^{2\pi} \frac{INTC(x_v, r) + INTC(x_w, r) - g(\alpha, x_v, x_w, r)}{2\pi} d\alpha\right) dx_w\right) dx_v \quad , R \geq 2r \tag{5.1.7}$$

$$\int_0^{R} \frac{2\pi x_v}{\pi R^2}\left(\int_0^{R} \frac{2\pi x_w}{\pi R^2}\left(\int_0^{2\pi} \frac{INTC(x_v, r) + INTC(x_w, r) - g(\alpha, x_v, x_w, r)}{2\pi} d\alpha\right) dx_w\right) dx_v \quad , R < 2r \tag{5.1.8}$$

When we increase the number of neighbors to $k(k > 2)$, the expression of the average auxiliary observable area offered by k neighbors becomes more and more complicated. Instead of trying to do cumbersome calculations to get the accurate value of avg_AOA, we estimate it through simulation. We randomly generated k neighbor's coordinates within node u's transmission range R, divided node u's sensing area into small grids and measured how many grids can be covered by at least one of its neighbors. The simulation results when $k = 1$ to $k = 10$ are shown in Figure 5.5. As we can see, the simulation result when $k = 1$ is the same as the derivation from the Equation 5.1.5, as shown in Figure 5.3. This implies the correctness of our simulation code. Another observation from Figure 5.5 is that the average auxiliary observable area offered by k neighbors increases with k and decreases with the ratio R/r.

The above analysis leads to another rule for node scheduling. If a threshold of maximal auxiliary unobservable area AUA_{max} is chosen a priori for each individual node, we can determine the minimal neighbor number K to guarantee this threshold from Figure 5.5. During node scheduling, a node examines if its neighbors' number is equal to or greater than the value of K. If affirmative, the node will set its status as off-duty. Note the value K is dependent on not only the threshold but also the ratio

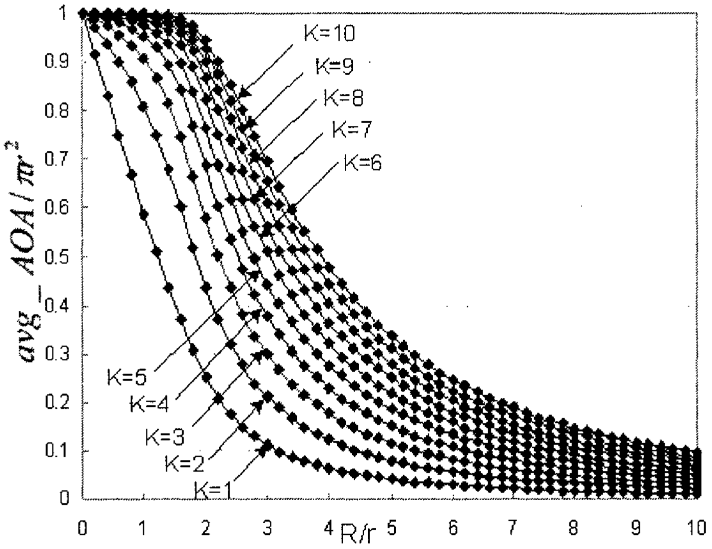


Figure 5.5: Neighbor-number-based off-duty eligibility model: relationship of avg_AOA offered by k neighbors and the ratio of communication range to sensing range

of communication range to sensing range. For instance, when the threshold is set as 20%, the corresponding K is 2 if $R/r = 1$. If $R/r = 2$, however, K should be 6.

Note that when R/r is larger than 10, Figure 5.5 may need to be extended through more simulations to present the given threshold.

5.1.3 Probability-based Off-duty Eligibility Model

The above two rules need nodes to collect neighbor information (neighbor number or distance) before node scheduling. Next, we will consider another off-duty eligibility rule, which does not count on any neighbor information. It is a probability-based one. In Chapter 3, we assume that the sensors in wireless sensor networks are distributed as per a homogeneous spatial Poisson process of intensity λ in a 2-dimensional space.

In other words, the probability that the number of nodes in the area of s, denoted as $N(s)$, is equal to n is given by:

$$P[N(s) = n] = \frac{e^{-\lambda s(\lambda s)^n}}{n!}, \text{ for } s \geq 0 \text{ and } n = 0, 1, 2, ... \tag{5.1.9}$$

According to the properties of the Poisson Process, the probability of n nodes in the area s around an origin but excluding the origin can be given by:

$$P[N(s - origin) = n] = \frac{e^{-\lambda(s-origin)[\lambda(s-origin)]^n}}{n!} \approx \frac{e^{-\lambda s(\lambda s)^n}}{n!} \tag{5.1.10}$$

The probability that {the distance d_{min} to the nearest neighbor is longer than x } is equivalent to the probability that {there are no other nodes with a distance less than x from the concerned node}, which is given by $P[d_{min} > x] = P[N(\pi x^2 - origin) = 0] = e^{-\lambda \pi x^2}$. Therefore, the probability that {the distance d_{min} to the nearest neighbor is less than or equal to x } is:

$$P[d_{min} \leq x] = 1 - e^{-\lambda \pi x^2} \tag{5.1.11}$$

As described in section 5.1.1, for a given threshold of maximal auxiliary unobservable area, there is a corresponding D. As long as the distance to the nearest neighbor is not more than D, the node will take an off-duty status. By using equation 5.1.11, we can estimate the off duty probability corresponding to a given D as:

$$\rho(D, \lambda) = P[d_{min} \leq D] = 1 - e^{-\lambda \pi D^2} \tag{5.1.12}$$

Next, we would like to know the probability of at least k neighbors near a given node. Equation 5.1.10 tells us that the probability of exactly i nodes in the area s around an origin is about $\frac{e^{-\lambda s(\lambda s)^i}}{i!}$. The probability of at least k neighbors near one node is equivalent to the probability that {there are at least k nodes with the distance of communication range R from the node}, which is given by:

$$\rho(k, \lambda) = P[N(\pi R^2) \geq k] = 1 - \sum_{i=0}^{k-1} \frac{e^{-\lambda \pi R^2} (\lambda \pi R^2)^i}{i!} \tag{5.1.13}$$

Similarly, for each given threshold of maximal auxiliary unobservable area, there is the corresponding K. In turn, for a given K, we can estimate the corresponding ρ as a function of K:

$$\rho(K, \lambda) = P[N(\pi R^2) \geq K] = 1 - \sum_{i=0}^{K-1} \frac{e^{-\lambda \pi R^2} (\lambda \pi R^2)^i}{i!} \tag{5.1.14}$$

Equations 5.1.12 and 5.1.14 are both functions of λ. However, λ is not a constant during node scheduling. It decreases over time in that nodes are scheduled sequentially and the successive nodes will not consider those neighboring nodes that have taken *off-duty* status before(more detailed explanation is presented in section 4.2). λ reaches its maximal value λ_{max} when the system is initially deployed and can be estimated as the ratio of the total node number N to the deployed area $\mathring{A}$. We denote the intensity λ_n when the system has n active nodes (including those nodes that have already selected *on-duty* status and that have not made a decision yet) as $\lambda_n = \frac{n}{\mathring{A}}$. Initially, $\lambda_{max} = \lambda_N = \frac{N}{\mathring{A}}$. Assume that node scheduling is in a strict sequence, i.e. without two nodes scheduling at the same time.

The probability of the first node taking *off-duty* status is:

$$P(\text{1st node off}) = P(\text{off with } N \text{ active nodes})) = \rho(\lambda_N)$$

The probability of the second node taking *off-duty* status is:

$$\begin{aligned} P(\text{2nd node off}) &= P(\text{1st node on}) \times P(\text{off with } N \text{ active nodes})) \\ &+ P(\text{1st node off}) \times P(\text{off with } N-1 \text{ active nodes}) \\ &= (1 - \rho(\lambda_N)) \times \rho(\lambda_N) + \rho(\lambda_N) \times \rho(\lambda_{N-1}) \end{aligned}$$

The probability of the third node taking *off-duty* status is:

$$\begin{aligned}
P(\text{3rd node off}) &= P(\text{1st node on and 2nd node on}) \times P(\text{off with } N \text{ active nodes})) \\
&+ P(\text{1st node on and 2nd node off}) \times P(\text{off with } N-1 \text{ active nodes}) \\
&+ P(\text{1st node off and 2nd node on}) \times P(\text{off with } N-1 \text{ active nodes}) \\
&+ P(\text{1st node off and 2nd node off}) \times P(\text{off with } N-2 \text{ active nodes}) \\
&= (1-\rho(\lambda_N)) \times (1-\rho(\lambda_N)) \times \rho(\lambda_N) + \rho(\lambda_N) \times (1-\rho(\lambda_{N-1})) \times \rho(\lambda_{N-1}) \\
&+ (1-\rho(\lambda_N)) \times \rho(\lambda_{N-1}) \times \rho(\lambda_{N-1}) + \rho(\lambda_N) \times \rho(\lambda_{N-1}) \times \rho(\lambda_{N-2})
\end{aligned}$$

The probability of (i)-th node eligibility is dependent on the scheduling result of all the predecessors. That makes the expression more and more complicated as i increases. To simplify the problem, in our initial design, we ignore the fact that the on-duty probability may be non-zero and assume that there are $N+1-i$ nodes remaining active, when the i-th node begins to schedule. Based on the assumption, we obtain:

$$P(\text{ind node off}) = \rho(\lambda_{N+1-i}) \tag{5.1.15}$$

Substituting Equation 5.1.12 into Equation 5.1.15, we can approximate the off-duty node percentage corresponding to D as:

$$\frac{1}{N}\sum_{i=1}^{N}\rho(\lambda_{N+1-i}) = \frac{1}{N}\sum_{i=1}^{N}(1-e^{-(N+1-i)\pi D^2/\mathring{A}} = 1 - \frac{e^{\frac{-N\pi D^2}{\mathring{A}}} - 1}{N(1-e^{\pi D^2/\mathring{A}})} \tag{5.1.16}$$

Substituting Equation 5.1.14 into Equation 5.1.15, we can approximate the off-duty node percentage corresponding to K as:

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N}\rho(\lambda_{N+1-i}) &= \frac{1}{N}\sum_{i=1}^{N}[1-\sum_{j=0}^{k-1}\frac{e^{-(N+1-i)\pi R^2/\mathring{A}} \times ((N+1-i)\pi R^2/\mathring{A})^j}{j!}] \\
&= 1 - \frac{1}{N}\sum_{i=1}^{N}[e^{-(N+1-i)\pi R^2/\mathring{A}} \times \sum_{j=0}^{k-1}\frac{((N+1-i)\pi R^2/\mathring{A})^j}{j!}]
\end{aligned} \tag{5.1.17}$$

The experiments presented in section 5.3 show that the off-duty percentages estimated by using Equation 5.1.16 and 5.1.17 are very close to the real results. The above two formulas associate the probability-based scheme with the other two schemes, thereby enable us to a priori estimate the obtained system intensity before running the nearest-neighbor-based or neighbor-number-based node scheduling scheme. By choosing an appropriate D or K value, we may control coverage loss as well as node density.

5.2 Node scheduling Schemes

In the following, we describe three schemes based on the above models respectively. All of them allow users to choose a threshold AUA_{max} to restrict the maximal sensing area loss at an individual node. Then the corresponding D or K value can be obtained by direct calculation or retrieving Figure 5.2 or Figure 5.5 that have been stored in memory in a data-pair manner. Whether a node is eligible for off-duty is directly determined by the value of D or K, and thereby indirectly determined by the threshold AUA_{max}.

Similarly to the location-based node-scheduling scheme in section 4.2, in all of the three new proposed schemes, the system operation is divided into duty cycles. Each duty cycle starts with a node-scheduling phase to assign working status to all nodes. After that, *off-duty* nodes fall asleep and may turn off their sensing unit, even the communication unit, to save energy, while those nodes taking *on-duty* status are responsible for performing sensing tasks and for data collection and data delivery until the coming of the next duty cycle.

The three schemes differ in how a node determines its working status and on whether the mechanism to prevent the simultaneous removal problem needs to be introduced.

5.2.1 Probability-based Node Scheduling Scheme

The simplest and most light-weighted node-scheduling scheme is based on the probability model described in section 5.1.3. In this scheme, each node generates a random number from [0,1) and checks if the number is less than the off-duty probability ρ. If that is the case, the node will take *off-duty* status. Otherwise, it keeps active. Note that after a corresponding D or K value has been found for a chosen threshold AUA_{max}, ρ can be directly calculated by using Equation 5.1.16 or 5.1.17 respectively.

5.2.2 Nearest Neighbor-based Node Scheduling Scheme

This scheme starts with all the nodes broadcasting a short message to let their neighbors know their existence and estimate the distance to them. This phase is not necessary if all the nodes already have a knowledge of neighbor distance by overhearing the data packets transmitted in the previous duty cycles. After collecting the above information, each node can determine its working status by examining if its distance to the nearest neighbor, d_{min}, is not more than the threshold D. If affirmative, the node can take *off-duty* status. However, since this scheme suffers from the simultaneous removal problem addressed in section 4.2 as well, a similar protection mechanism has to be introduced. How a node is scheduled is formally described below:

1. Generate a random back-off time T_d. During T_d, listen to the channel, and delete the sender from the active neighbor list, if a *Status Advertisement Message*

(SAM) is received.

2. After time T_d, find d_{min} from all the active neighbors. If $d_{min} > D$, set status as *on-duty* and enter step 5. Otherwise, set status as *ready-to-off*, broadcast a *Status Advertisement Message* and perform step 3.
3. Wait for a short time T_w. During T_w, listen to the channel. If receiving a *Status Advertisement Message* (SAM), delete the sender from the active neighbor list and re-investigate the off-duty eligibility. If $d_{min} > D$, set status as *on-duty* and enter step 5. Otherwise, continue to perform step 3.
4. After T_w, set the status as *off-duty* and fall asleep.
5. Node scheduling has been done. Wait to enter working phase.

Note that in this scheme, the decision is based on the relative distance of nodes. Thus a distance-estimation technique has to be available at each node. In the case where each node has a knowledge of its own location, the neighbor distance is easy to obtain by each node piggybacking its own location information in its outgoing packets. However, in other cases where the node location is unknown, the signal strength of an incoming message can be used to estimate the distance.

Similarly to our previous location-free node-scheduling scheme described in section 4.2, network topology or energy factors can be involved into the derivation of the back-off delay, other than a simple random delay.

5.2.3 Neighbor Number-based Node Scheduling Scheme

In the previous scheme, neighbor distance is used by nodes to decide their working status. In this scheme, the number of neighbors is used instead.

Similarly, at the beginning of each duty cycle, all the nodes need to broadcast a short message to let their neighbors know their existence. This phase is not necessary if all the nodes have recorded their neighbor *ID* by overhearing data packets transmitted in the previous duty cycles. After getting the above neighbor information, each node can determine its working status by examining if its neighbors' number exceeds a given threshold value K. If affirmative, the node will take *off-duty* status. Similarly, this node-scheduling scheme has to solve the simultaneous removal problem addressed in section 4.2. Formal description of this node-scheduling scheme is presented below:

1. Generate a random back-off time T_d. During T_d, listen to the channel, and delete the sender from the active neighbors list, if a *Status Advertisement Message* (SAM) is received.

2. After time T_d, count the number of active neighbors. If the number $< K$, set status as *on-duty* and enter step 5. Otherwise, take *ready-to-off* status, broadcast a *Status Advertisement Message* and perform step 3.

3. Wait for a short time T_w. During T_w, listen to the channel. If receiving a *Status Advertisement Message* (SAM), delete the sender from the active neighbor list and re-investigate the off-duty eligibility. If the number of active neighbors $< K$, set status as *on-duty* and enter step 5. Otherwise, continue to perform step 3.

4. After T_w, set the status as *off-duty* and fall asleep.

5. Node scheduling has been done. Wait to enter working phase.

Similarly, network topology or energy factor can be considered in the derivation of the back-off delay as well.

5.3 Performance Evaluation

In this section, we present some experimental results to compare the above schemes in the following terms:

- *Off-duty percentage:* the ratio of off-duty nodes' number to the initial deployed nodes' number. This metric implies the energy saving capability owned by a node-scheduling scheme. The more nodes get off-duty status, the less energy is consumed by the system during the working phase.
- *Coverage percentage loss:* the difference between the original system sensing coverage and the sensing area covered by all on-duty nodes after a node-scheduling scheme is executed. This metric indicates the coverage preservation capability provided by a node-scheduling scheme. The less the coverage loss, the better.
- *Average sensing degree:* this metric is defined as the average number of on-duty nodes simultaneously detecting and reporting an event. Its value is the indicator of the observation fidelity of a node-scheduling scheme. The larger the sensing degree, the more credible the system observation. However, more on-duty nodes are needed in order to obtain a larger sensing degree . On the other hand, less on-duty nodes are often expected for more energy saving. To achieve a trade-off, an ideal scheme is to maximize the number of off-duty nodes on the premise of guaranteeing the minimal fidelity.

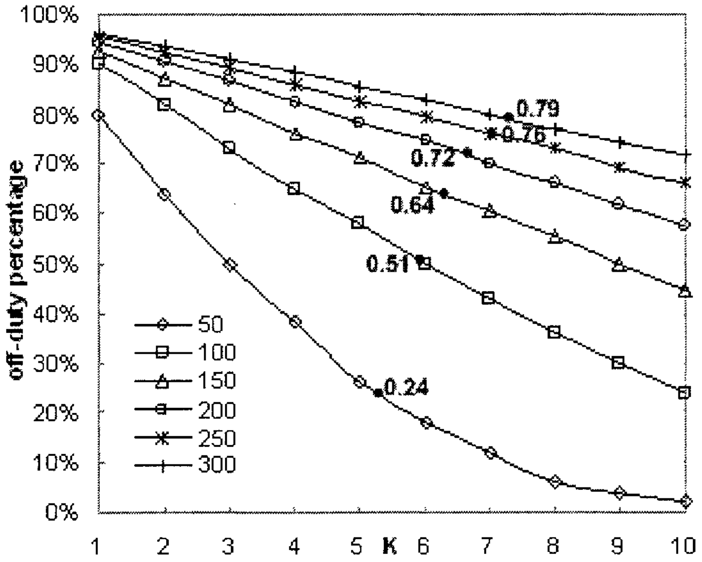


Figure 5.6: Neighbor_number_based node scheduling scheme: off-duty percentage vs.K

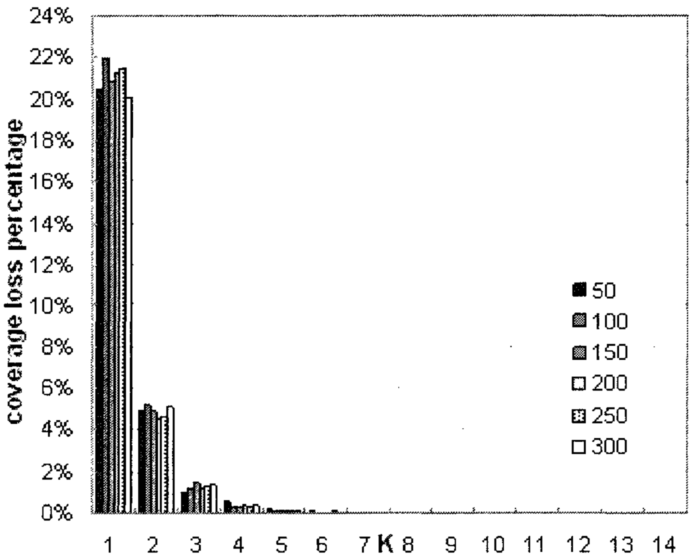


Figure 5.7: Neighbor_number_based node scheduling scheme: coverage loss percentage vs.K

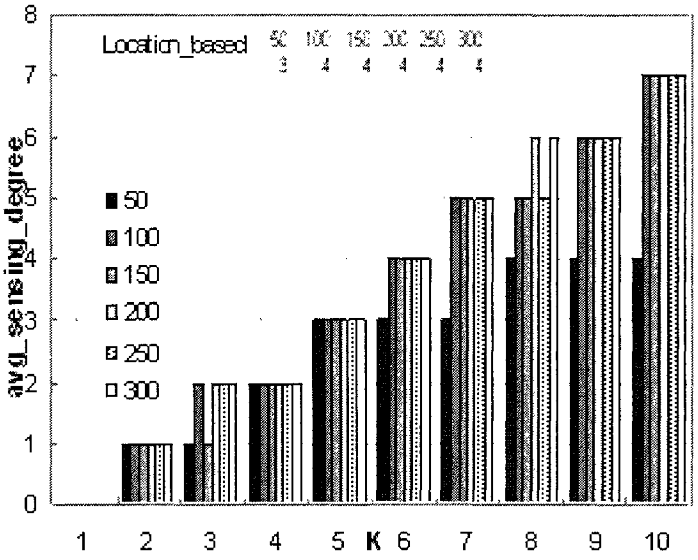


Figure 5.8: Neighbor_number_based node scheduling scheme: average sensing degree vs.K

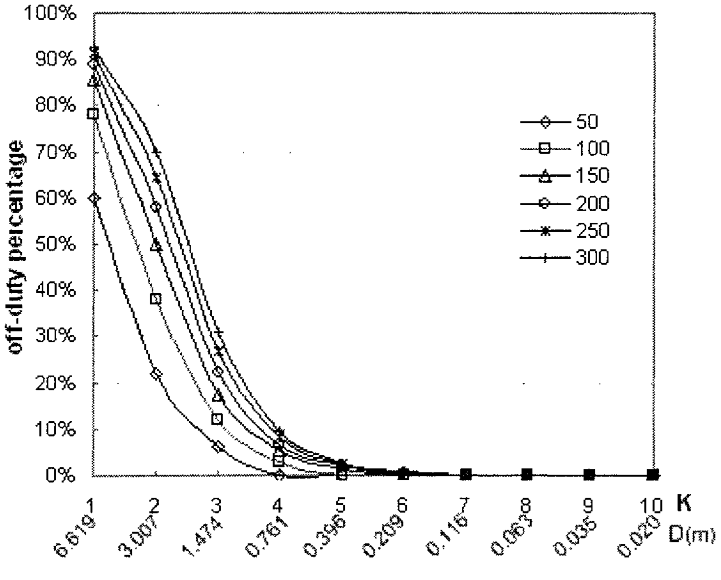


Figure 5.9: Nearest_neighbor_based node scheduling scheme: off-duty percentage vs.D

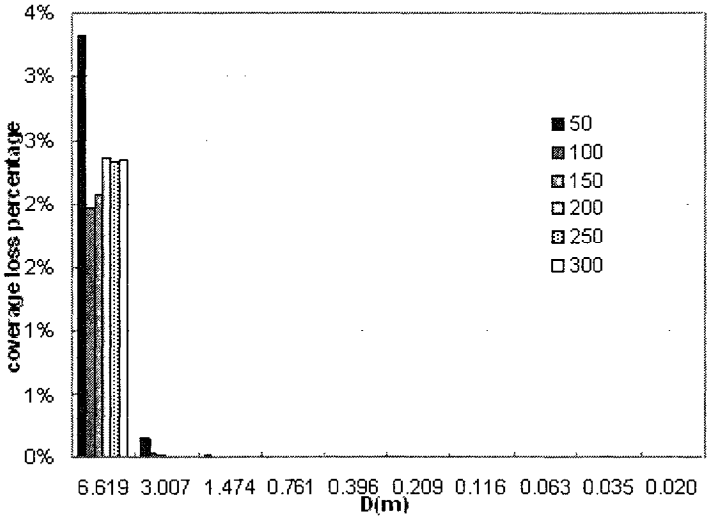


Figure 5.10: Nearest_neighbor_based node scheduling scheme: coverage loss percentage vs.D

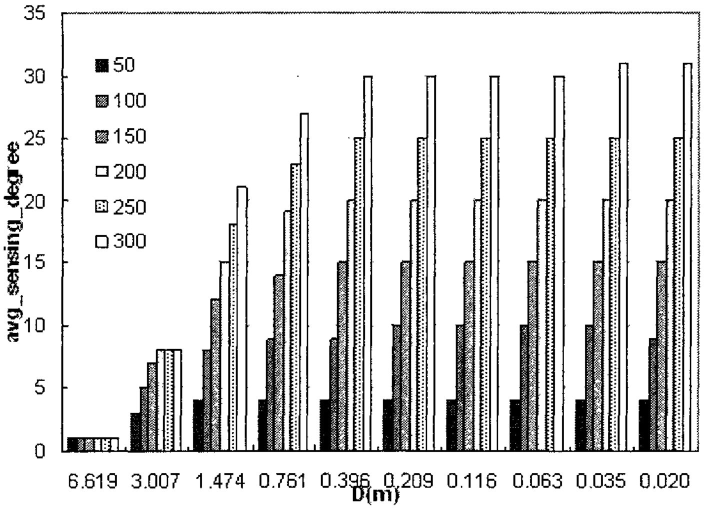


Figure 5.11: Nearest_neighbor_based node scheduling scheme: average sensing degree vs.D

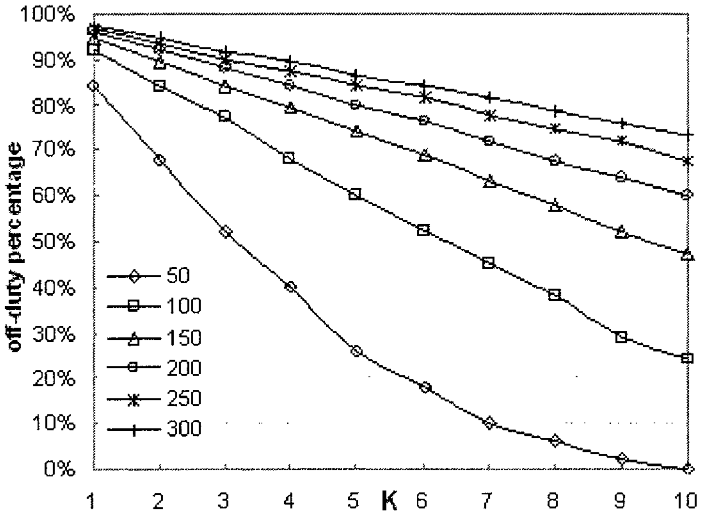


Figure 5.12: Neighbor_number_probability_based node scheduling scheme: off-duty percentage vs.K

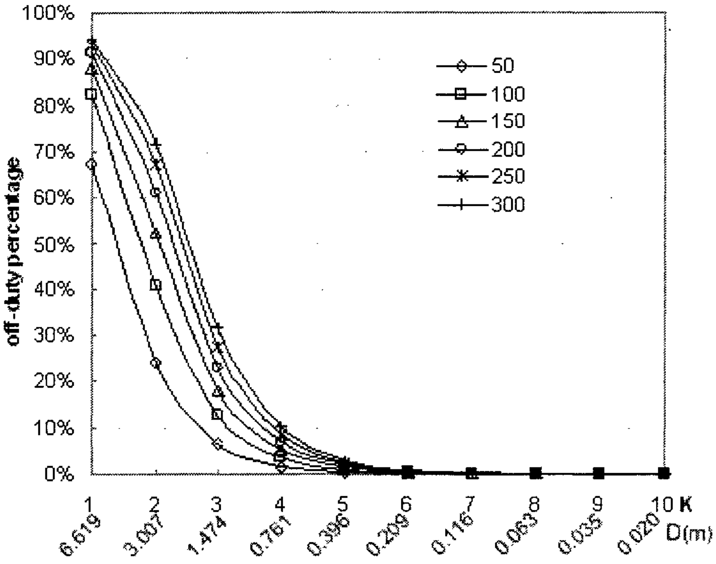


Figure 5.13: Nearest_neighbor_probability_based node scheduling scheme: off-duty percentage vs.D

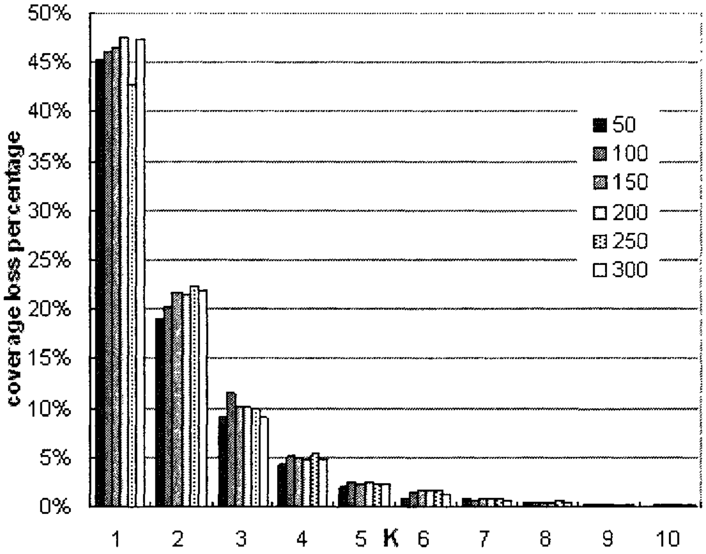


Figure 5.14: Neighbor_number_probability_based node scheduling scheme: coverage loss percentage vs.K

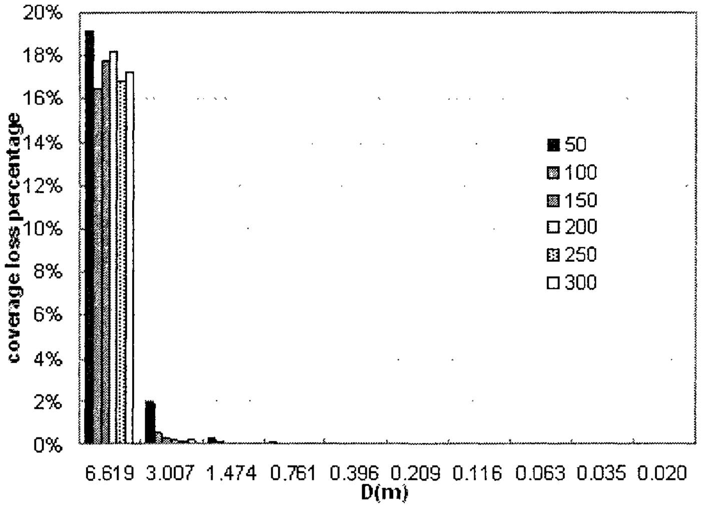


Figure 5.15: Nearest_neighbor_probability_based node scheduling scheme: coverage loss percentage vs.D

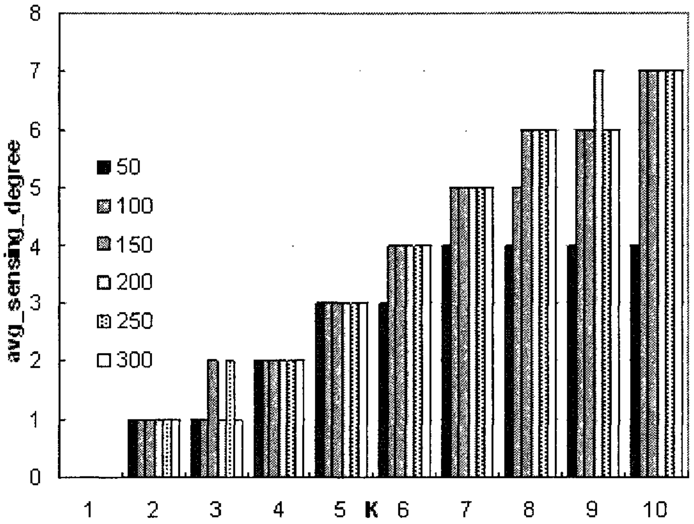


Figure 5.16: Neighbor_number_probability_based node scheduling scheme: average sensing degree vs.K

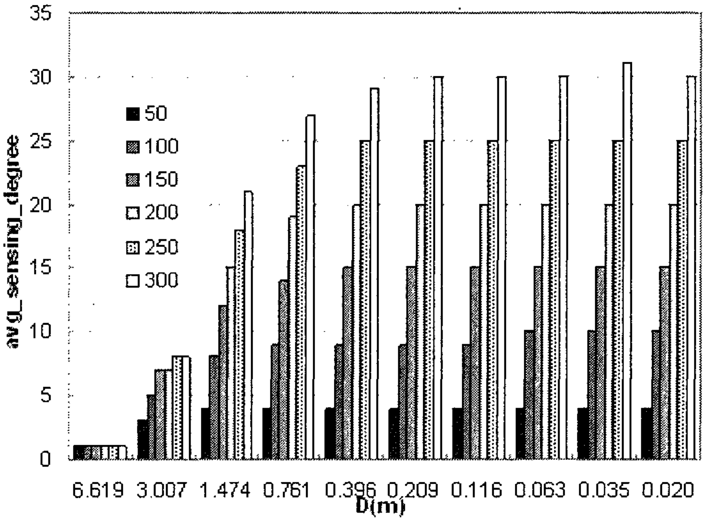


Figure 5.17: Nearest_neighbor_probability_based node scheduling scheme: average sensing degree vs.D

The above three node-scheduling algorithms are evaluated by using the simplified simulator described in section 4.3.1. The same parameters are adopted by them. The only difference is the off-duty eligibility rule they apply. The experimental results are shown in Figures 5.6 - 5.17. Each point in these figures represents the mean value from 100 random network topologies.

Figures 5.6 - 5.8 shows the performance of the neighbor_number_based node-scheduling scheme when κ (the ratio of communication range to sensing range) is set as 1. The same behavior is observed at other κ values as well. To visualize the comparison, we embed some solid points and their labels into Figure 5.6 to indicate the corresponding results obtained by running the location_based node_scheduling scheme described in section 4.2. For the same reason, we attach a small table into Figure 5.7 and Figure 5.8 respectively. As we can see, a larger K value leads to less off-duty nodes, thereby less coverage loss and a larger average sensing degree. The off-duty percentage increases as the node density increases. But the number of the obtained on-duty nodes does not show dramatic change over node density when we fix the value of K and node density rises to a level. In other words, the node density does not have a distinct impact on the coverage percentage loss and the average sensing degree when it is high enough. When we take the sensing degree obtained from running the location-based node-scheduling scheme as the baseline and select 3 as an expected level of detection fidelity, then K should be selected as 5 correspondingly. At this threshold, the coverage percentage loss is less than 0.3%, as shown in Figure 5.10. The off-duty percentage of the neighbor_number_based node scheduling scheme is larger than that of the location_based node scheduling scheme. That implies that the neighbor_number_based node scheduling scheme can lead to more energy saving.

Another observation from the experimental result is the loss of the whole system coverage is always much less than the observable area loss at an individual node. For instance, when K is chosen as 1, the corresponding AUA_{max} is about 41.4% according to Figure 5.5, while the whole system coverage loss is about 20%, instead.

The performance of the nearest_neighbor_based node-scheduling scheme is shown in Figure 5.9 - 5.11. The values of the threshold D are selected purposely to facilitate the comparison between the neighbor_number_based node scheduling scheme and the nearest_neighbor_based node-scheduling scheme. For instance, when K=1, the corresponding is about 41.4% according to Figure 5.5. Thus we set the value of D to match the same threshold. By looking up Figure 5.2 or resolving the root of the Equation 5.1.2, we know that D should be 6.619m. Similarly, the other D values are listed under their corresponding K values along the x-axis in Figures 5.9 - 5.11.

Comparing Figures 5.6 - 5.8 and Figures 5.9 - 5.11, we find that at the same value of AUA_{max}, the nearest_neighbor_based node-scheduling scheme generates fewer off-duty nodes; thereby a less coverage loss and a larger obtained sensing degree. The difference is more distinct as the value of AUA_{max} decreases. The reason is that there is a great chance for a node to have enough neighbors, but none of them close enough to it.

Figures 5.12 - 5.17 illustrates the performance of the probability_based node-scheduling scheme. The values of ρ are also chosen purposely. For instance, when $K = 1$ and $N = 50$, we can calculate the corresponding ρ as 85% by substituting them into the expression 5.1.17. When $D = 6.619$ and $N = 50$, the value of ρ should be set as 67% according to expression 5.1.16. Different results based on a given neighbors' number or a given nearest neighbor distance are presented separately. One of the most important observations is that the off-duty percentages obtained from probability_based scheme are very close to the values from the neighbor_number_based scheme or the nearest_neighbor_based scheme. This implies that the expression 5.1.17 and 5.1.16 can be used to approximately estimate the off-duty node number before node scheduling. Another observation is that the probability_based scheme results in much more coverage loss than its corresponding non_probability_based scheme. That is because in the probability_based scheme, each node "blindly" determines its work status, just based on the probability without considering the nearby node distribution.

5.4 Summary

Table 5.1 summarizes the difference between all of the proposed node-scheduling schemes in this and previous chapter. Compared with the other schemes, the probability-based node-scheduling algorithm is the simplest and the most light-weighted, because it does not rely on any kind of neighbor information to determine node status and because it does not introduce any communication cost to coordinate scheduling among adjacent nodes. But it leads to more coverage loss than the other algorithms. The location-based scheme is designed for complete coverage preservation. However, it requires the availability of a location-estimate technique. Furthermore, it cannot control the off-duty node number. Therefore it is not as flexible as the other schemes. As compared to the nearest-neighbor-based scheme, the neighbor-number-based scheme can generate more off-duty nodes to guarantee the same coverage loss threshold. Therefore, it is better than the former.

Table 5.1: Comparison of different node-scheduling schemes

	Location_based	Nearest_neighbor _based	Neighbor_number _based	Probability_based
Parameter_based	No	Yes,thereby flexible	Yes,thereby flexible	Yes,thereby flexible
Location_aware	Yes	No	No	No
Need to collect neighbor information	Yes,neighbor location	Yes,neighbor distance	Yes,neighbor number	No
Need calculation during scheduling	Yes, with complexity O(neighbor number)	No	No	No
Need to broadcast status change	Yes	Yes	Yes	No
Enable pre-estimation of on-duty node number[1]	Difficult to do, maybe impossible	Possible	Possible	Possible
Preserve original sensing coverage[2]	100% preservation	May have loss	May have loss	Have the greatest loss
Off-duty percentage[3]	Relative large	Relative small	Largest	N/A[4]

1. Thereby enable control of on-duty number
2. Theoretically, in an ideal running environment
3. Upon the same requirement of preserved coverage capability
4. Can be large or small. Depends on how the value of probability is determined

Chapter 6

Connectivity Maintenance and Coverage Preservation

In wireless sensor networks, sensors are performing two operations: sensing and communication. Therefore, there might exist two kinds of redundancy in a network. Most of the previous work (including our work in Chapter 4 and Chapter 5) addressed only one kind of redundancy: sensing or communication alone. [33] [78] first discussed how to combine consideration of coverage and connectivity maintenance in a single activity scheduling. They provided a sufficient condition for safe scheduling integration in those fully covered networks. However, random node deployment makes initial sensing holes inside the deployed area inevitable even in an extremely high-density network. Therefore, in this chapter, we enhance their work for general wireless sensor networks by proving another conclusion: *"the communication range size is twice that of the sensing range" is the sufficient condition and the tight lower bound to ensure that complete coverage preservation implies connectivity among active nodes if the original network topology (induced by all the deployed nodes) is connected.* Also,we extend the conclusion to k-degree network connectivity and k-degree coverage preservation.

The rest of this chapter is organized as follows. In section 6.1, we figure out the relationship between connectivity maintenance and coverage preservation. In section 6.2, we extend the result of section 6.1 to a higher degree. In section 6.3, we present some experimental results. In the last section, we conclude the chapter.

6.1 Network connectivity on coverage preservation

In this chapter, all the terms and notations will follow the definitions in Chapter 3.

As mentioned above, some existing node-scheduling algorithms [75] [77] have the capability of complete coverage preservation, i.e. $C(V_{active}) = C(V)$. Next, we will prove that $R \geq 2r$ is the sufficient condition and the tight lower bound to ensure that any subgraph $G(V_{active})$ induced from a connected network graph $G(V, E)$ is still connected, when the subgraph $G(V_{active})$ satisfied that the condition: $C(V_{active}) = C(V)$.

Our proof is based on the following assumptions:

1. There may be a large number of sensor nodes in the network. However, the number is always finite.

2. There are no two nodes located at the exactly the same position.

First, we prove the sufficient condition.

Lemma 6.1.1. *If $R \geq 2r$ and $C(V_{active}) = C(V)$, let node u and node v be any pair of nodes in $V(G)$ satisfying (i) $u \in V_{inactive}(G)$ (ii) $2r < d(u,v) \leq R$, then there*

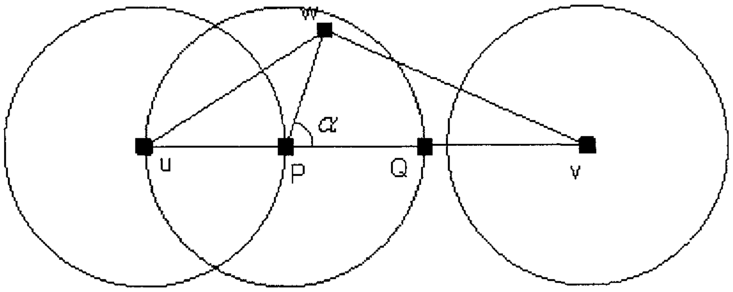


Figure 6.1: Illustration of Lemma 6.1.1

must exist another node w satisfying (i) $w \in V_{active}(G)$ (ii) $w \in N(v)$ (iii) $w \in N(u)$ (iv) $d(u,w) \leq 2r$.

Proof. As illustrated in Figure 6.1, suppose segment $\overline{uv}$ interacts with the boundary of $S(u)$ at the point P. There must exist an active node, supposing w, whose distance to point P is not larger than the sensing range r. Otherwise, the system overall sensing coverage would be reduced after node scheduling (because point P can be covered by the inactive node u, but cannot be reached by any active node). By using the triangle inequality, we have $d(v,w) \leq d(w,P) + d(v,P) \leq r + d(u,v) - r = d(u,v) \leq R$. In the same way, we have $d(u,w) \leq d(u,P) + d(w,P) \leq r + r \leq R$. So w is the element in $N(u)$ and $N(v)$.

□

Lemma 6.1.2. *If $C(V_{active}) = C(V)$, let node u and node v be any pair of nodes in $V(G)$ satisfying (i) $u \in V_{inactive}(G)$ (ii) $d(u,v) \leq 2r$, then there must exist another node w satisfying (i) $w \in V_{active}(G)$ (ii) $Q \in S(w)$, where point Q is the intersection point between segment $\overline{uv}$and the boundary of $S(v)$ as illustrated in Figure 6.2.*

Proof. We split out consideration into two cases.

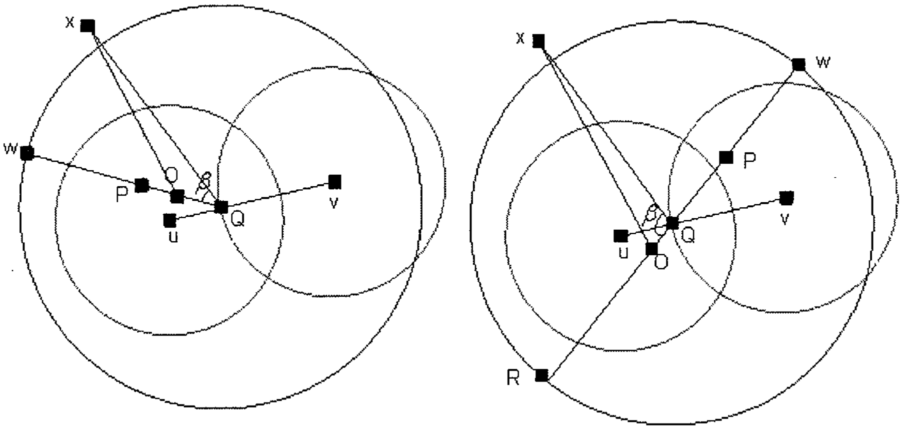


Figure 6.2: Illustration of Lemma 6.1.2

case i: The first case is that node v is an inactive node. Because coverage is completely preserved after node scheduling, there must be an active node that can cover point Q.

case ii: We prove the lemma for the case that node v is an active node. Since in the network, there are no two nodes whose positions are exactly the same, there must be other nodes besides node v in the active node set $V_{active}(G)$. Otherwise, the sensing area of node u cannot be completely preserved after node scheduling. Suppose in $V_{active}(G) - \{v\}$, node w has the minimal distance to point Q. Assume $d(w, Q) = r + \eta, \eta > 0$. As illustrated in Figure 6.2, the segment $\overline{wQ}$ interacts with the boundary of $S(w)$ at point P. The distance between node w and point P is $d(w, P) = r$ and the distance between point P and point Q is $d(P, Q) = d(w, Q) - d(w, P) = \eta$. We consider the following two sub-cases.

- For the sub-case shown on the left of Figure 6.2, i.e. $90^o \leq \angle PQv \leq 180^o$, we can always find a point O, located in segment $\overline{PQ}$ and satisfying (i) $0 <$

$d(O,Q) = \varepsilon < \eta$ (ii) $0 < d(O,P) = \eta - \varepsilon < \eta$ (iii) $O \in S(u)$.

$\because d(w,O) = r + \eta - \varepsilon > r$, $\therefore$ node w cannot cover point O.

$\because d(v,O) = \sqrt{d(v,Q)^2 + d(O,Q)^2 - 2 \times d(v,Q) \times d(O,Q) \times \cos \angle PQv}$ $\geq \sqrt{d(v,Q)^2 + d(O,Q)^2} > d(v,Q) = r$, where $90^o \leq \angle PQv \leq 180^o$, $\therefore$ node v cannot cover point O, either. If $V_{active} - \{v,w\}$ is not empty, let node x be any node in it. The distance from node x to point Q must be $d(x,Q) = r+\eta+\delta, \delta \geq 0$ because w is the node in $V_{active} - \{v\}$ whose distance to point Q is minimal.

By using the triangle inequality, we have $d(x,O) \geq d(x,Q) + d(O,Q) = r + \eta + \delta - \varepsilon \geq r + \eta - \varepsilon = d(w,O)$, where $0 \leq \beta \leq \pi$. This implies that if node w cannot cover point O, other active nodes in $V_{active} - \{v,w\}$ cannot, either. Obviously, it contradicts with the condition $C(V_{active}) = C(V)$.

- For the sub-case shown on the right of Figure 6.2, i.e. $0^o \leq \angle PQv < 90^o$, we always find a point O that is located in line PQ, satisfying (i) $0 < d(O,Q) = \varepsilon' < \eta$ (ii) $0 < d(O,P) = \eta + \varepsilon' > \eta$ (iii) $O \in S(u)$.

  $\because d(w,O) = d(w,Q) + d(Q,O) = r + \eta + \varepsilon' > r$, $\therefore$ node w cannot cover point O.

  $\because d(v,O) = \sqrt{d(v,Q)^2 + d(O,Q)^2 - 2 \times d(v,Q) \times d(O,Q) \times \cos \angle OQv}$ $> \sqrt{d(v,Q)^2 + d(O,Q)^2} > d(v,Q) = r$, where $90^o < \angle OQv = (180^o - \angle PQv) \leq 180^o$, $\therefore$ node v cannot cover point O, either. If $V_{active} - \{v,w\}$ is not empty, let node x be any node in it, whose distance to point Q is $d(x,Q) = r+\eta+\delta', \delta' \geq 0$. $d(x,O) = \sqrt{d(x,Q)^2 + d(O,Q)^2 - 2 \times d(x,Q) \times d(O,Q) \times \cos \beta} \geq d(x,Q) - d(O,Q) = r + \eta + \delta' - \varepsilon' \geq r + \eta - \varepsilon' > r$, where $0 \leq \beta \leq \pi$. This implies that there is no active node that can cover point O. So it contradicts with the

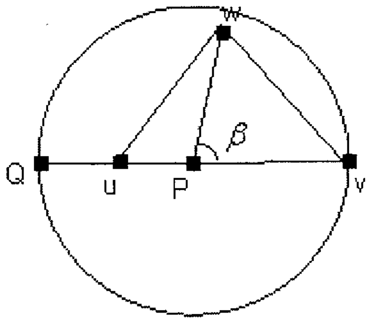


Figure 6.3: Illustration of Lemma 6.1.3

condition $C(V_{active}) = C(V)$.

Therefore, we conclude Lemma 6.1.2 by contradiction. ☐

Lemma 6.1.3. *If $R \geq 2r$ and $C(V_{active}) = C(V)$, let node u and node v be any pair of nodes in $V(G)$ satisfying (i) $u \in V_{inactive}(G)$ (ii) $r < d(u,v) \leq 2r$, then there must exist another node w satisfying (i) $w \in V_{active}(G)$ (ii) $w \in N(v)$ (iii) $w \in N(u)$ (iv) $d(u,w) < d(u,v)$.*

Proof. As illustrated in Figure 6.3, segment $\overline{uv}$ interacts with the boundary of $S(v)$ at point P. According to Lemma 6.1.2, there must exist another node, supposing node w , which is an active one and whose distance to point P doesn't exceed the sensing range, i.e. $0 \leq d(w,P) \leq r$. According to the triangle ineqality, $d(v,w) \leq d(v,P) + d(w,P) \leq r + r \leq R$. Therefore, node w is the element of $N(v)$. Also, $d(u,w) \leq d(u,P)+d(w,P) \leq d(u,v)-d(v,P)+r = d(u,v)-r+r = d(u,v) \leq 2r \leq R$. Therefore, w is the element of $N(u)$. Because w and v may not be located at the same place, $d(u,w) < d(u,v)$. ☐

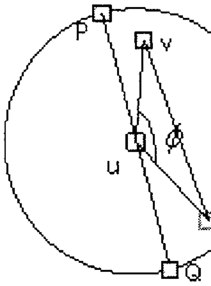


Figure 6.4: Illustration of Lemma 6.1.4

Lemma 6.1.4. *If* $R \geq 2r$*, let node* u*, node* v *and node* w *be any three nodes in* $V(G)$ *satisfying (i)* $0 < d(u,v) \leq r$ *(ii)* $0 < d(u,w) \leq r$*, then node* v *and node* w *are adjacent to each other.*

Proof. $d(v,w) \leq d(u,v) + d(u,w) \leq r + r \leq R$. ☐

Lemma 6.1.5. *If* $R \geq 2r$ *and* $C(V_{active}) = C(V)$*, let node* u*, node* v *and node* w *be any three nodes in* $V(G)$ *satisfying (i)* $u \in V_{inactive}(G)$ *(ii)* $v \in N(u)$ *(iii)* $w \in N(u)$*, then node* v *and node* w *can communicate with each other directly or through an active path in* $G(V,E)$*.*

Proof. We split our consideration into several cases in terms of distances of node v and node w to node u.

Case i: $0 < d(u,v) \leq r$ *and* $0 < d(u,w) \leq r$*.*

According to Lemma 6.1.4, node v and node w can communicate directly with each other.

Case ii: In $\{v,w\}$*, one node has a distance to node* u *not larger than the sensing range. The other has a distance to node* u *larger than the sensing range and but smaller than twice of the sensing range. Without loss of generality, we assume that* $0 < d(u,v) \leq r$ *and* $r < d(u,w) \leq 2r$*.*

If $d(v,w) \leq R$, then node v and node w can communicate with each other directly. Otherwise (i.e. $d(v,w) > R$), according to Lemma 6.1.3, there must exist another active node x in $V_{active}(G)$ satisfying (i) $x \in (N(u) \cap N(w))$ (ii) $d(u,x) < d(u,w)$. It is easy to know $v \neq x$ because $x \in N(w)$ and $v \in \overline{N(w)}$. If we can find such node x further satisfying $0 < d(u,x) \leq r$, then according to the conclusion in *case i*, node v and x are adjacent with each other. Thus, node v and w can communicate with each other via the active path $(v,x)(x,w)$. Otherwise (i.e. $r < d(u,x) < d(u,w) \leq 2r$), according to Lemma 6.1.3, there must exist another active node y with $y \neq x \neq w$ satisfying that $y \in (N(u) \cap N(x))$ and $d(u,y) < d(u,x)$. If the selected node y is node v, then node v and node w can communicate with each other via the active path $(v = y, x)(x,w)$. Otherwise (i.e. $v \neq y \neq x \neq w$), if we can find such an active node y further satisfying $0 < d(u,y) \leq r$, according to the conclusion in *case i*, node v and node y are adjacent with each other. Thus, node v and node w can communicate with each other via the active path $(v,y)(y,x)(x,w)$. However, if $r < d(u,y) < d(u,x) < d(u,w) \leq 2r$, there must be another active node z satisfying that $x \in (N(u) \cap N(y))$ and $d(u,z) < d(u,y)$. Similarly, it can be concluded that there is an active path $(v = z, y)(y,x)(x,w)$ or $(v,z)(z,y)(y,x)(x,w)$ unless $r < d(u,z) \leq 2r$. The proof is going on if the new identified active node has a distance to node u larger than the sensing range. Since, the number of nodes in the graph $G(V,E)$ is finite, the process can always end at an active node that equals to node v or whose distance to node u is equal to or smaller than the sensing range. In any case, there exists a network path between node v and node w whose intermediate nodes are all active ones. So we can conclude the Lemma 6.1.5 for *case ii.*

Case iii: In $\{v,w\}$, one node has a distance to node u equal to or smaller than *the sensing range. The other has a distance to node u exceeding twice of the sensing range. Without loss of generality, we assume* $0 < d(u,v) \leq 2r$ *and* $2r < d(u,w)$.

If $d(v,w) \leq R$, then node v and node w can communicate with each other directly. Otherwise (i.e. $d(v,w) > R$), according to Lemma 6.1.1, there must exist another active node x in $V_{active}(G)$ satisfying that $x \in (N(u) \cap N(w))$ and $d(u,x) \leq 2r$. According to the conclusions for *case i* and *case ii*, node v and node x are adjacent to each other or can communicate through an active path $v \longleftrightarrow_{active} x$. Consequently, the active path between node v and node w is $(v,x)(x,w)$ or $v \longleftrightarrow_{active} x + (x,w)$. Therefore, we conclude the Lemma 6.1.5 for *case iii*.

Case iv: Both nodes v and w have a distance to node u larger than the sensing range and equal to or smaller than twice of the sensing range, i.e. $r < d(u,v) \leq 2r$ *and* $r < d(u,w) \leq 2r$.

If $d(v,w) \leq R$, then node v and node w can communicate with each other directly. Otherwise (i.e. $d(v,w) > R$), similar to the proof process for *case ii*, we can always get a sequence of active nodes, denoted as $x_1, x_2, ..., x_i, i \geq 1$, satisfying (i) $d(u,x_i) < d(u,x_{i-1}) < ... < d(u,x_2) < d(u,x_1) < d(u,w)$ (ii) $x_j \in N(u), 1 \leq j \leq i$ (iii) $x_1 \in N(w)$ (iv) $x_j \in N(x_{j-1}), 2 \leq j \leq i$ (v) $\{r < d(u,x_j) \geq 2r$ and $x_j \neq v, 1 \leq j < i\}$ (vi) $\{0 < d(u,x_i) \leq r$ or $x_i = v\}$. If $x_i = v$, we get an active path between node v and node w as $(w,x_1)(x_1,x_2)..(x_{i-1},x_i = v)$. Otherwise (i.e. $x_i \neq v$ and $0 < d(u,x_i) \leq r$), since $0 < d(u,x_i) \leq r$ and $r < d(u,v) \leq 2r$, according to the conclusion in *case ii*, we know that node v and node x_i must be adjacent to each other or connected through an active path $v \longleftrightarrow_{active} x_i$. Because we have got an active path between node w and node x_i as $(w,x_1)(x_1,x_2)..(x_{i-1},x_i)$, after concatenating the above two paths, we get an active path between node v and node w as $(w,x_1)(x_1,x_2)..(x_{i-1},x_i)(x_i,v)$ or

$(w,x_1)(x_1,x_2)..(x_{i-1},x_i)(x_i,v) + v \longleftrightarrow_{active} x_i$. Therefore, we conclude the Lemma 6.1.5 for *case iv*.

Case v: In $\{v,w\}$, one node has the distance to node u larger than the sensing range and equal to or smaller than twice of the sensing range. The other has the distance exceeding twice of the sensing range. Without loss of generality, we assume $r < d(u,v) \leq 2r$ and $2r < d(u,w)$.

If $d(v,w) \leq R$, then node v and node w can communicate with each other directly. Otherwise (i.e. $d(v,w) > R$), according to Lemma 6.1.1, there must exist another active node x in $V_{active}(G)$ satisfying that $x \in (N(u) \cap N(w))$ and $d(u,x) \leq 2r$. According to the conclusion for *case iv*, node v and node x are adjacent to each other or can communicate through an active path $v \longleftrightarrow_{active} x$. So, we can get an active path between node v and node w as $(v,x)(x,w)$ or $v \longleftrightarrow_{active} x + (x,w)$. Therefore, we conclude the Lemma 6.1.5 for *case v*.

Case vi: Both nodes v and w have the distance to node u larger than twice of the sensing range, i.e. $2r < d(u,v)$ and $2r < d(u,w)$.

If $d(v,w) \leq R$, then node v and node w can communicate with each other directly. Otherwise (i.e. $d(v,w) > R$), according to Lemma 6.1.1, there must exist another active node x satisfying that $x \in (N(u) \cap N(w))$ and $d(u,x) \leq 2r$. According to the conclusions for *case v* and *case iii*, node v and node x are adjacent to each other or can communicate through an active path $v \longleftrightarrow_{active} x$. Thus, we can get an active path between node v and node w as $(v,x)(x,w)$ or $v \longleftrightarrow_{active} x + (x,w)$. Therefore, we conclude the Lemma 6.1.5 for *case vi*. □

Lemma 6.1.6. *If $R \geq 2r$ and $C(V_{active}) = C(V)$, for any inactive path in $G(V,E)$, there is a corresponding active path in $G(V,E)$, which ends at the same pair of nodes.*

Proof. Let P_0 be an inactive path ending at node u and node v: $P_0 = (u = w_1, w_2)(w_2, w_3)...(w_{i-2}, w_{i-1})(w_{i-1}, w_i = v)$. In $\{w_j, 2 \leq j \leq i-1\}$, the number of inactive nodes is l, $l \geq 1$. Suppose w_k is the first inactive intermediate node in path P_0. We know that all $w_j, 2 \leq j \leq k-1$ are active ones and the number of inactive nodes in $\{w_j, k+1 \leq j \leq i-1\}$ is $l-1$. According to Lemma 6.1.5, node w_{k-1} and node w_{k+1} are adjacent to each other or can communicate through an active path $w_{k-1} \longleftrightarrow_{active} w_{k+1}$. Thus, we get a new path $P_1 = (u = w_1, w_2)(w_2, w_3)...(w_{k-2}, w_{k-1})(w_{k-1}, w_{k+1})(w_{k+1}, w_{k+2})...(w_{i-1}, w_i = v)$ or $P_1 = (u = w_1, w_2)(w_2, w_3)...(w_{k-2}, w_{k-1}) + w_{k-1} \longleftrightarrow_{active} w_{k+1} + (w_{k+1}, w_{k+2})...(w_{i-1}, w_i = v)$. In either case, the number of inactive intermediate nodes in the new path P_1 is one less than that in path P_0. In the same way, we can find another new path P_2 whose inactive intermediate node number is $l-2$. The process is going on. In each step, the number of inactive intermediate nodes is decremented by one. Since the node number in the graph $G(V, E)$ is finite, the number of node in the old path P_0 is finite as well. This implies that we can finally get a path between node u and node v in $G(V, E)$, where all intermediate nodes are active ones. □

Theorem 6.1.7. *When $R \geq 2r$ and the system sensing coverage is completely preserved after node scheduling, if a network graph $G(V, E)$ is originally connected, then the induced subgraph $G(V_{active})$ must be connected.*

Proof. Let node u and node v be any pair of nodes in $V_{active}(G)$. $\because V(G) \supseteq V_{active}(G)$ and the network graph $G(V, E)$ is connected, $\therefore$ there must exist a path between these two nodes in $G(V, E)$. Let path P be such a path. If P is active, according to the definition of $G(V_{active})$, path P is in $G(V_{active})$. If path P is inactive, according to Lemma 6.1.6, we can always find a corresponding path Q between node u and node

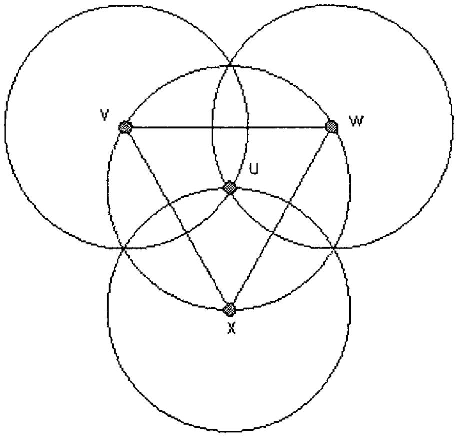


Figure 6.5: Connectivity maintenance for $R < 2r$

v in $G(V, E)$, whose all intermediate nodes as active ones, i.e. path Q is in $G(V_{active})$. That is, for any pair of nodes in $V_{active}(G)$, we can always find a path in $G(V_{active})$ when the original network graph $G(V, E)$ is connected. The proof ends. □

Theorem 6.1.7 just tells us that $R \geq 2r$ guarantees connectivity maintenance after complete coverage preservation. It does not imply disconnection among active nodes in the networks with $R < 2r$. Consider the simple scenario illustrated in Figure 6.5 with $R = 1.8r$. Node u is the only node assigned an inactive status after node scheduling. The distance between any pair of node v, node w and node x is $\sqrt{3}r < R$. Therefore, the subgraph consisting of the three inactive nodes is still connected. So $R \geq 2r$ is not the necessary condition to ensure that coverage preservation implies connectivity maintenance. However, we can prove that $R = 2r$ is the tight lower bound for this conclusion.

Theorem 6.1.8. *When $R \geq 2r$ and the system sensing coverage is completely preserved after node scheduling, the induced network subgraph $G(V_{active})$ may be disconnected even when the original network graph $G(V, E)$ is connected.*

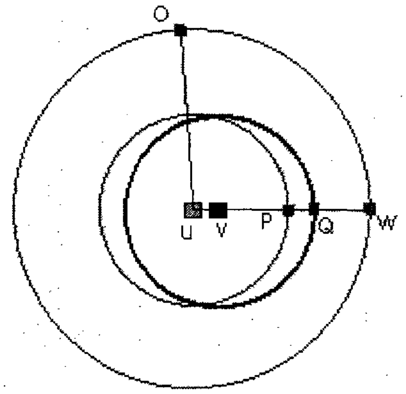


Figure 6.6: Illustration for Theorem 6.1.8

Proof. Assume $R = 2r - \xi, 0 < \xi < 2r$. Let us consider the scenario shown in Figure 6.6, node u and node v are adjacent to each other and has a distance between them as $0 < \mu = d(u, v) < min(r, \xi, R)$. Besides node u and node v, there are a sufficient number of nodes located on the circle centered at node u with a radius as $R + \varepsilon$, as illustrated in Figure 6.6. ε satisfies that $0 < \varepsilon < \mu$. The union of the sensing areas of these nodes forms a "ring", whose inner radius is $R + \varepsilon - r$ and whose outer radius is $R + \varepsilon + r$. From the figure, we can see that node u can cover part of node v's sensing area. The remaining part is a "crescent". The "crescent" is contained in the "ring". That is because (i) The closed distance from node u to any point in the "crescent" is r that satisfies $r > R + \varepsilon - r (\because r = 2r - r = R + \xi - r > R + \mu - r > R + \varepsilon - r)$, and (ii) The farthest distance from node u to any point in the crescent is $d(u, Q) = r + \mu$ that satisfies $r + \mu < r + R + \varepsilon (\because r + \mu = r + R + \mu - R < r + R + 0 < r + R + \varepsilon)$. This implies that node v can take an inactive status during node scheduling. Before

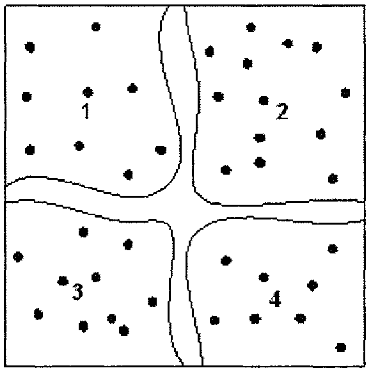


Figure 6.7: Illustration the conclusion in [33] [78] and ours

node scheduling, the network is connected because $d(u, v) = \mu < R$ and $d(v, w) = R + \varepsilon - \mu < R$. However, after removing node v, node u becomes an "isolated" node because its distance to any other active node is $R + \varepsilon > R$. Therefore, the proof ends. ☐

Combining the proofs for Theorem 6.1.7 and Theorem 6.1.8, we can conclude that:

Theorem 6.1.9. *$R \geq 2r$ is the sufficient condition and tight lower bound to ensure that complete sensing coverage preservation implies network connectivity among active nodes in an original connected network graph.*

To the best of our knowledge, [33] [78] are the earliest papers discussing how to integrate activity scheduling in both domains: communication and sensing. Both of them independently proved the same conclusion: *if a convex region is completely covered by a set of nodes, the communication graph consisting of these nodes is connected when $R \geq 2r$.* In this chapter, we enhance their work by providing the proof: *if an original network is connected and the identified active nodes can cover the same region as all the original nodes, then the active nodes are connected when $R \geq 2r$.* The difference between their conclusion and ours can be better understood through a simple scenario as illustrated in Figure 6.7. In this scenario, the sensing field can be divided into four subfields 1-4, with each subfield completely covered by a subset of nodes. We call the node subset as subset 1-4 correspondingly. There are two narrow gaps in the middle of the sensing field that cannot be monitored by any node. Furthermore, the communication graph consisting of all the nodes are connected. After node scheduling, redundant nodes is removed from each subset without changing the coverage graph. According to the work in [33] [78], we can only know that the active nodes within each subset are connected with each other, but cannot determine if the active nodes among different subsets are connected. However, the latter can be confirmed to be affirmative, according to the proofs above.

The conclusion in this section is significant in that it establishes the relationship between connectivity maintenance and coverage preservation, enables us to focus on sensing domain when connectivity and coverage are both required, and therefore greatly simplifies the safe integration of activity scheduling in both domains.

6.2 k-degree connectivity on k-degree coverage preservation

The result in the previous section can be extended to the relationship between k-degree connectivity and k-degree coverage preservation with $k > 1$. That is, $R \geq 2r$ is the sufficient condition to ensure that k-degree coverage preservation implies k-degree connectivity maintenance.

Before proving it, we first formulate definitions of k-degree connectivity and k-degree coverage preservation. We call a graph $G(V, E)$ is k-connected if for every

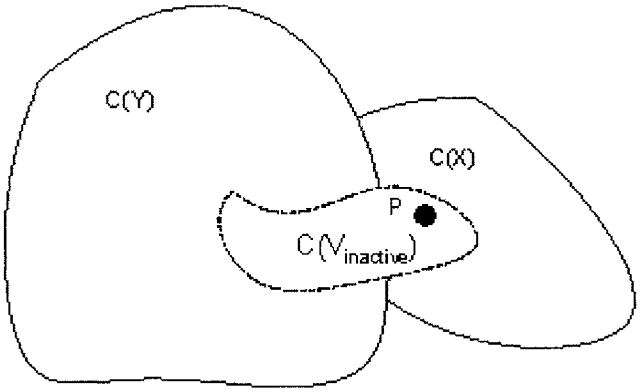


Figure 6.8: Illustration of Theorem 6.2.1

subset X of $V(G)$ with fewer than k elements, $V(G) - X$ is connected. We say that a node-scheduling algorithm can preserve k-degree coverage, if for every inactive node identified by that algorithm, any point in its sensing area can be covered by at least k active nodes. Similar to 1-degree coverage preservation explained in the previous section, this definition accepts the situation that part of the expected monitoring area $\mathring{A}$ is weakly covered initially due to random node deployment, i.e. the following discussion is not restricted to those networks where every point in the expected monitoring area $\mathring{A}$ must be covered by at least k nodes before node scheduling. It just requires that node-scheduling algorithms can guarantee that the initially weak coverage would not be exacerbated.

Theorem 6.2.1. *When $R \geq 2r$ and the system sensing coverage is completely k-degree preserved after node scheduling, if a network graph $G(V, E)$ is originally k-connected, then the induced subgraph $G(V_{active})$ must be k-connected.*

Proof. Let node set X be any subset of V_{active} fewer than k elements. We denote the node subset $V_{active} - X$ as Y, consisting of all the other elements in V_{active}. Based on coverage definition, $C(X) \cup C(Y) = C(V_{active})$. Since the original system coverage is preserved after node scheduling, we have $C(V) = C(V_{active})$. $\because C(V) = C(V_{active} + V_{inactive}) = C(V_{active}) \cup C(V_{inactive}) = C(V_{active})$, $\therefore C(V_{active}) \supseteq C(V_{inactive})$. We can prove $C(V_{inactive}) \subseteq C(Y)$ by contradiction. Suppose there is a point P in $C(V_{inactive})$ satisfying that $P \in \overline{C(Y)}$, as shown in Figure 6.8. Point P must be in $C(X)$ because $C(X) \cup C(Y) = C(V_{active}) \supseteq C(V_{inactive})$. According to the definition of k-degree coverage preservation, point P must be covered by at least k active nodes. Since node set X has fewer than k elements, there must be a node in Y, covering point P. It is contradictory with the assumption $P \in \overline{C(Y)}$. Therefore, we have $C(V_{inactive}) \subseteq C(Y)$. We can further get $C(Y + V_{inactive}) = C(Y)$. Based on the definition of k connectivity, the subgraph $G(Y+V_{inactive})$ is connected. Also, after removing the node subset $V_{inactive}$ from the network graph $G(Y + V_{inactive})$, the system coverage is not reduced. Therefore, according to Theorem 1, the subgraph $G(Y)$ is also connected. Removing any subset X with fewer than k elements from $G(V_{active})$, the subgraph $G(V_{active} - X)$ is still connected. The means $G(V_{active})$ is k-connected. Therefore, the proof ends. □

6.3 Experimental Results

The experiments are performed based on the light-weighed sensing simulator implemented for evaluating the off-duty eligibility model in section 4.3.1. However, the model in section 4.3.1 is not optimal in that it conservatively underestimates the intersection areas of neighboring nodes for easy calculation purposes. That implies the model generates more active nodes than an optimal one. In order to eliminate the

Figure 6.9: connectivity maintenance and coverage preservation: overall sensing coverage of a network with 350 nodes in a 100*100 m2 region

effect of this problem, we made a small modification of the off-duty eligibility rule to reduce the size of the active node set. We divide each node's sensing area into small grids and determine each node's status by checking if each grid inside its sensing area is covered by a node succeeding the concerned node in the scheduling sequence. It is justified in this context because we focus on the connectivity maintenance among minimal active nodes in original connected network graphs instead of simplicity and efficiency of the scheduling algorithm.

Figure 6.9-6.14 illustrates a network topology with 350 nodes uniform randomly placed in a 100-by-100-meter region, when the sensing range is 6 meters and the original system sensing coverage is shown as Figure 6.9. The modified coverage-preserving node scheduling algorithm in Chapter 4 identifies some inactive nodes as marked in Figure 6.10 while maintaining the system sensing coverage as unchanged. Figure 6.12 shows that the connectivity among active nodes is destroyed with the communication

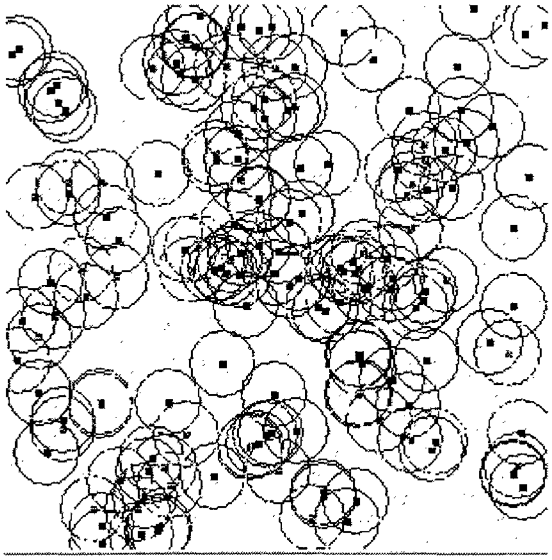


Figure 6.10: connectivity maintenance and coverage preservation: scheduling result with r as 6 meters(black circles representing inactive nodes)

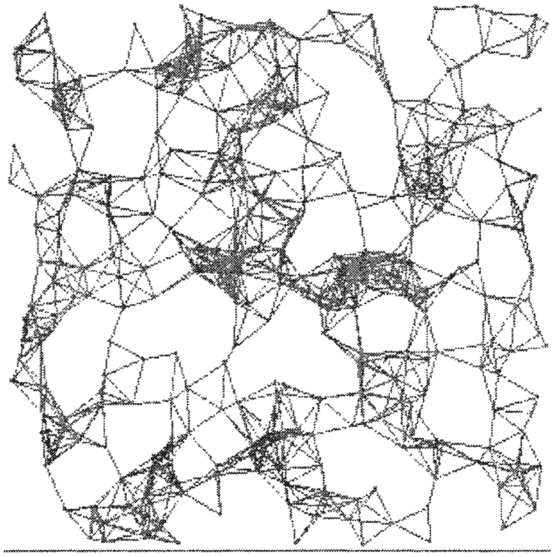


Figure 6.11: connectivity maintenance and coverage preservation: network is connected before node scheduling with $R = 9m$

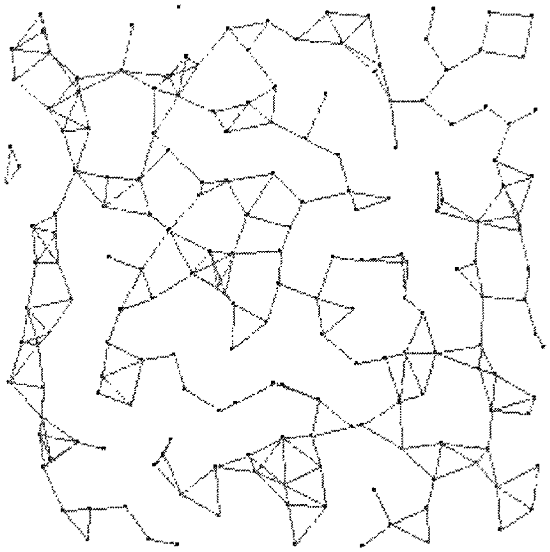


Figure 6.12: connectivity maintenance and coverage preservation: active nodes are disconnected with $R = 9m$

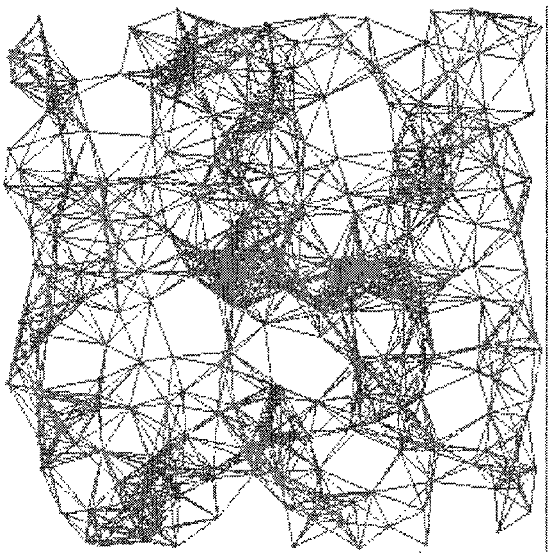


Figure 6.13: connectivity maintenance and coverage preservation: network is connected before node scheduling with $R = 12m$

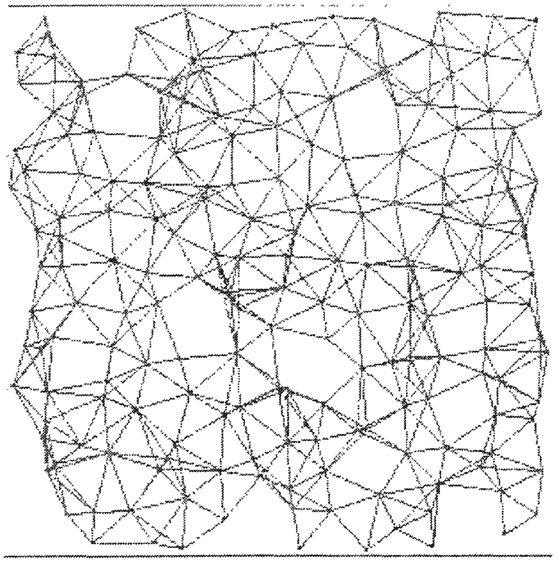


Figure 6.14: connectivity maintenance and coverage preservation: active nodes are disconnected with $R = 12m$

range is 1.5 times the sensing range. While in the same sensing configuration, if the communication range is twice the size of the sensing range, the connectivity is still maintained, as illustrated in Figure 6.14. Energy saving can be achieved by communicating upon the simplified network graph with a lower node density and lower link density, as shown in Figure 6.14, and by monitoring the environments with less active nodes, as shown in Figure 6.12, compared with the original network graph.

We also investigate the impact of the ratio of communication range to sensing range on the relationship between connectivity maintenance and coverage preservation. The networks we study have a sensing density from an average 1.6 nodes per sensing area to 17.3 nodes per sensing area, which are obtained by varying the deployed node number from 200 to 550 and varying the size of the sensing range based on a certain ratio to a fixed communication range that is 10 meters. For each configuration, we generate 1000 random network topologies that are original connected.

Table 6.1: The number of network topologies with connectivity maintenance after coverage preserving sensing-scheduling (N nodes in a 100*100 m2 region, communication range $R = 10m$)

	R/r												
N	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2
200	670	813	879	955	982	997	1000	999	1000	1000	1000	1000	1000
250	769	794	901	971	984	994	995	999	1000	1000	1000	1000	1000
300	648	794	896	957	980	996	996	1000	1000	1000	1000	1000	1000
350	699	812	885	954	985	997	1000	1000	1000	1000	1000	1000	1000
400	634	784	890	952	989	997	998	1000	1000	1000	1000	1000	1000
450	656	803	887	960	982	995	999	1000	1000	1000	1000	1000	1000
500	666	785	888	953	978	991	999	1000	1000	1000	1000	1000	1000
550	674	801	885	955	987	998	998	1000	1000	1000	1000	1000	1000

Among these 1000 network topologies, those maintaining connectivity after coverage preserving sensing-scheduling are counted. The results are listed in Table 6.1. Consistent with our expectation, the smaller the ratio of the communication range to the sensing range is, the lower the connectivity ratio is. However, as shown by the data, the lower bound of the range ratio for connectivity maintenance seems to be 1.8 instead of 2.0. That is because we run just 1000 network topologies for each configuration and the scenario illustrated in Figure 6.6 is a rare and extreme situation with a very high node density. Furthermore, from the table, we observe that node density seems to have no significant impact on connectivity possibility when the ratio of communication range to sensing range is fixed. Approximating the connectivity possibility based on a given ratio of communication range to sensing range can be one of our future works.

6.4 Summary

In this chapter, we provided a complete proof: *"the communication range size is twice that of the sensing range" is the sufficient condition and the tight lower bound to ensure that complete coverage preservation implies connectivity among active nodes if the original network topology (consisting of all the deployed nodes) is connected.* The significance of the above conclusion is that it enables us to focus on the sensing domain when connectivity and coverage are both required. It therefore greatly simplifies the safe integration of activity scheduling in both domains. We also extended the result for k-degrees by proving that: *"the communication range size is twice that of the sensing range" is the sufficient condition to ensure that k-degree coverage preservation implies k-degree connectivity among active nodes if the original network topology (consisting of all the deployed nodes) is k-connected.*

Chapter 7

Conclusions and Future Research Direction

7.1 Conclusion

A wireless sensor network consists of tiny sensing devices, deployed in a region of interest. Although each device has restricted sensing, processing and wireless communication capabilities, coordination of them enables the system to carry out a big sensing task accurately and effectively. In wireless sensor networks, the energy source provided for sensors is usually constrained battery power. The environments where sensor networks are operating often prohibit users to recharge or replace battery power for sensors. However, long system lifetime is always expected by monitoring applications. Therefore, conserving energy resource, and thus prolonging system lifetime, is an important issue in the design of large-scale wireless sensor networks. Fortunately, due to low cost of sensors, it is well accepted that the nodes in wireless sensor networks can be initially deployed with a very high density. Then redundancy can be exploited to prolong system life.

In this thesis, we first proposed a novel node-scheduling scheme for dynamically configuring nodes' sensing activity to minimize the number of active nodes while maintaining the original system coverage. In this scheme, each node autonomously and periodically determines its work status based on its own and neighbors' positional information. To preserve sensing coverage, a node is eligible for off-duty only when its neighbors can help it to monitor its whole sensing area. To avoid simultaneous removing when two neighboring nodes expect each other's helping, a back-off delay is introduced to let each node postpone its evaluation with a random period of time.

This scheme relies on location information to calculate the union of overlapping areas among multiple nodes. It aims at completely preserving the system original sensing coverage. To alleviate the dependence on location, an intuitive thought is to relax the requirement of coverage preservation. In the second part of the thesis, we proposed three such alternative location-free scheduling schemes, which are based on neighbor number, nearest neighbor distance and probability of off duty node number, respectively. The common idea is that users are able to select the level of coverage percentage. Based on this level, the minimal neighbor number K, the nearest neighbor distance D or the off-duty probability ρ are determined by using a given expression or prior collected data-pairs. During the scheduling, each node determines its work status according to the corresponding value.

Monitoring is just one duty undertaken by sensors. Due to limited communication capability, each sensor has to serve as router to help others to forward data packets to remote base stations. Similarly, redundancy also exists in the communication domain in highly dense networks. Energy would be wasted when all the nodes continuously are listening to the media because adaptively turning off radio at some nodes may have no influence on network connectivity. In highly dense wireless sensor networks, energy waste always exists if we only remove one kind of redundancy. Minimizing active nodes in both domains should consider network connectivity and sensing coverage at the same time. Without sufficient coverage, the network cannot guarantee the quality of monitoring service. Without network connectivity, active nodes may not be able to send data back to base stations. Therefore, in the third part of this thesis, we enhance their work to support randomly-deployed wireless sensor networks by providing the proof of another conclusion: *"the communication range is twice the sensing range" is the sufficient condition and the tight lower bound to ensure that complete coverage preservation implies connectivity among active nodes, if the original network topology (induced by all the deployed nodes) is connected.* It is worth to notice that node-scheduling algorithms with the capability of complete coverage preservation have existed in the literature. Furthermore, we extended the conclusion from 1-degree to K-degrees for applications that may require different tolerance against node failures.

7.2 Future research directions

We identify the following research directions to continue our research work.

7.2.1 Integrated activity scheduling schemes in wireless sensor networks

In the completed work, we have proved that *"the communication range size is twice that of the sensing range" is the sufficient condition and the tight lower bound to ensure that complete coverage preservation implies connectivity among active nodes if the original network topology (consisting of all the deployed nodes) is connected.* This conclusion provides the theoretical basis for integrating activity scheduling in a single evaluation step. In other words, all the redundant nodes in sensing domain identified by a coverage-preserving node scheduling scheme are also redundant nodes in communication domain, if $R \geq 2r$. However, when the communication range R is much larger than the sensing range r, there is a lot of communication redundancy in the remaining active node after a sensing scheduling. Therefore, it is desirable to perform scheduling again in the communication domain. How to combine coverage-preserving scheduling algorithms, such as [77] [78] [33] , with dominating set algorithms, such as [44] [11] [80], to maximize the number of *both_inactive* (inactive in both sensing and communication domain), *sensing_inactive* (inactive only in sensing domain), *comm_inactive* (inactive only in communication domain) nodes, needs further investigation. In wireless sensor networks, all-to-one data collection can be viewed as a reverse broadcasting problem. Energy efficient broadcasting algorithms have been studied in the literature by constructing subgraphs such as LMST, RNG, etc [64] [35]. Constructing LMST/RNG before connected dominating set forms a three-phase scheduling scheme. What is the performance of this three-phase scheduling scheme compared with the two-phase one is our future work.

7.2.2 Impact of node density and network topology on coverage capability

Our previous works focus on algorithm design as to how to pick up more redundant nodes, while guaranteeing full or sufficient system coverage. Most of related works are contributed to algorithm design as well. Although in [15], Liu et al present preliminary analysis about the weakest-level coverage capability, defined as the percentage of monitored area that can be covered by at least one node in a square-distributed and a random-distributed network topology, their studies are far from being complete.

In the future, we will qualify any k-degree coverage capabilities ($k > 0$), defined as the percentage of monitored area that can be observed by at least k nodes in more kinds of node distributions (square-based, triangle-based, pentagon-based, hexagon-based, uniformly-distributed and Possion-distributed). Our purpose is to determine which kind of node distribution can create maximal coverage with minimal active nodes. The conclusion can be used to guide node deployment and network planning. Another goal is to bound the coverage capability in an unpredictable network topology with a given node number. The result can help us to determine the number of initially deployed nodes in order to achieve the desired coverage capability in a kind of network topology. In addition, the result is useful for evaluating node-scheduling algorithms based on the topology change caused by them.

7.2.3 Node deployment strategy to enhance coverage capability

In most of wireless sensor network scenarios, sensors are supposed to work in a remote or inhospitable environment. In such an environment, it is impossible or inconvenient or expensive to manually deploy all nodes according to the prior plan. Sensors may be thrown from a plane in such a way that the node distribution is random and unpredictable. Consequently, sensor nodes may not cover the entire monitored region, leaving some holes blinded. In addition, even if the initial deployment can be effectively controlled, more and more sensing holes or blind points will show up due to energy depletion or node destruction. To guarantee quality of monitoring, it is necessary to discover the sensing holes and alarm users to add new nodes in them. Thus, the following problems arise: how to find sensing holes efficiently and quickly; where to put new nodes to enhance the sensing coverage to a desired level while minimizing the number of added nodes. Besides to eliminate sensing holes or blind points, node deployment schemes have to consider those areas weekly covered (i.e. covered by those nodes whose energy resource may be used up before the next deployment).

7.2.4 Activity scheduling schemes based on 3-D general sensing model

Almost all the existing scheduling algorithms are based on simple 0/1 sensing model (whether an event can be detected is determined by its distance to sensors). More general sensing model defines that the sensing capability of a single sensor is exponentially proportional to its distance to the event, and final sensing result is the integral function of k sensors. Although the 0/1 model is a special case of the general model when $k = 1$, the latter is more realistic and sophisticated than the former. Therefore, the activity scheduling based on the general model needs further investigation. Also, all the existing discussions on coverage, detectability and scheduling are designed in 2-dimensional space. It will be quite difficult to transform the existing algorithms to 3-dimensional space. So new algorithms or schemes need to be explored.

Appendix-1: Extended LEACH (ns-2 simulation)

To evaluate the performance of the location_free node scheduling scheme, we extended LEACH ns-2 simulation code.

Ns [1] is an object-oriented, discrete event-driven network simulator developed at University of California at Berkeley. It is written in C++ with OTcl interpreter as a frontend. Ns was initially developed for simulating local and wide area networks, but Carnegie Mellon University extended it to support wireless networks. In ns simulation for wireless networks, the basic element is mobile nodes, as shown in Figure 7.1. An application class is used to perform different application-specific functions. It generates and sends data to the Agent class. The Agent implements the transport and network layer protocols. It consumes data and then forwards data to CMU-Trace, which is responsible for writing statistics information about the data packets to trace files. Then the data packets are passed to a Connector which sends them to the Link-Layer for processing. After a small delay, data packets are pushed into the Queue. They will wait there until all other packets in front have been transmitted. Once a packet is pulled out of the Queue, the MAC layer receives it and runs the media access protocols. The last processing unit of a mobile node class is the Network Interface, which adds transmit power to the packet and emits it in

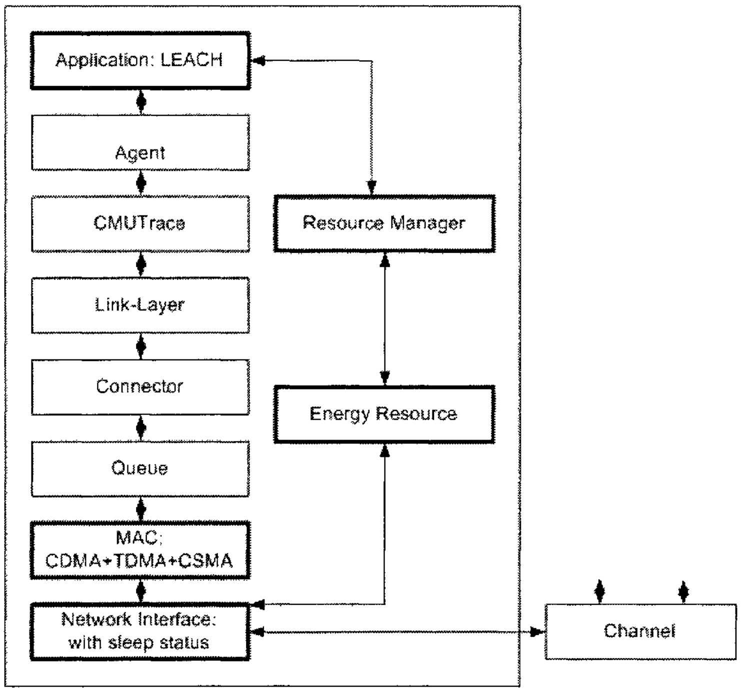


Figure 7.1: Architecture of Sensor Node in LEACH

the Channel. Thus, all the nodes connected with the Channel will receive a copy of the packet. The packet received will pass through the Network Interface, MAC, Link-Layer, Connector, CMUTrace, Agent, finally to the Application Layer.

The main functions of LEACH is implemented at the Application Layer in Tcl language. Its MAC layer is a combination of carrier-sense multiple access(CSMA), time-division multiple access(TDMA), and direct-sequence spread spectrum(DSSS). LEACH simulation code has a MAC class, called MacSensor, which performs CSMA functions. Since LEACH has a cross-layer architecture, it is the Application that determines at which timeslot with what direct-sequence a packet is sent when TDMA and DSSS are supposed to use. LEACH adds two classes for resource management. The Energy Resource records the current level of energy resource. The Resource

Manager provides a common interface between the application and energy resource. It allows the application to add or remove or query the supply of energy. In LEACH, the Network Interface can directly access the Energy Resource and remove energy used for transmission or reception. Furthermore, LEACH allows the Application to configure nodes' work status. If a node is in sleep state, the Network Interface discards any packet received from the Channel and does not consume any power for it.

The location_based node scheduling scheme is used to configure nodes' sensing status. In other words, it determines whether sensor nodes would generate and send data packets. Therefore, it should be implemented at the Application layer. We modified LEACH simulation code by inserting a scheduling phase before cluster forming phase as shown below:

```
################################################################################
# Application/LEACH instproc schedulePhase
################################################################################
Application/LEACH instproc schedulePhase {} {
  global ns_ opt
  global tracefd_1
  $self instvar code_ id_ neighborOff_ neighborID_ neighborX_ neighborY_
  $self instvar round_begin_time_ step_ next_change_time_ alive_

  set step_ 1
  self setTurnOn

  #initiate neighbor information
  $self setCode 0
  set neighborOff_ ""
  set neighborID_ ""
  set neighborX_ ""
  set neighborY_ ""

  #Check the alive status of the node.  If the node has run out of
  #energy, it will no longer perform functions in the network.
  set ISalive [[[$self node] set netif_(0)] set alive_]
  if {$alive_ == 1 && $ISalive == 0} {
    $chan removeif [[$self node] set netif_(0)]
    set alive_ 0
    set opt(nn_) [expr $opt(nn_) - 1]
  }
  if {$alive_ == 0} {return}

  #set the timeout for next scheduling
  set now_ [$ns_ now]
  set next_change_time_ [expr $now_ + $opt(ch_change)]
  $ns_ at $next_change_time_ "$self schedulePhase"

  #advertise position
  set random_access [$self getRandomNumber 0 [expr $opt(adv_wait) - $opt(adv_xtram)]]
  set time [expr $now_ + $random_access]
  $ns_ at $time "$self advertisePosition"

  #set decide turnoff time
  set random_access [$self getRandomNumber 0 [expr $opt(backoff_time)- $opt(adv_xtram) -
$opt(compute_time)]]
  set time [expr $now_ + $opt(adv_wait) + $random_access]
  $ns_ at  $time "$self decideTurnOff"
}
################################################################################
# Application/LEACH instproc clusterFormingPhase
################################################################################
Application/LEACH instproc clusterFormingPhase {} {
  global ns_ tracefd_1
  $self instvar id_ step_

  if {[$self isTurnOff?] != 1 } {
     set now_ [$ns_ now]
     set step_ 2
     $ns_ at $now_ "$self decideClusterHead"
  }
}
################################################################################
# Application/LEACH instproc decideTurnoff
################################################################################
Application/LEACH instproc decideTurnOff {} {
  global opt ns_ PI TURNOFF
  global tracefd_1
  $self instvar rng_ id_ round_begin_time_

  set now_ [$ns_ now]

  #calculate the percentage of sensing area which can be covered by neighbors
  $self calculateCoverage

  if {[$self isFull?]} {
      #if the node is eligible for off-duty, broadcast its eligibility and wait a grace
period for turning off
      $self setWaitTurnOff
      set random_delay [expr [$ns_ now] + [$rng_ uniform 0 0.001]]
      $ns_ at  $random_delay "$self advertiseState $TURNOFF"
      $ns_ at [expr $now_ + $opt(turnoff_wait)] "$self setTurnOff"
  } else {
      #otherwise, falling sleep to save energy until clusterForming phase is scheduled to
start
       $self GoToSleep
       $ns_ at [expr $round_begin_time_ + $opt(scheduling_time)] "$self beginWork"
 }
}
```

```
##############################################################################
# Application/LEACH instproc beginWork
##############################################################################
Application/LEACH instproc beginWork {} {
  $self WakeUp
  $self clusterFormingPhase
}
##############################################################################
# Application/LEACH instproc calculateCoverage
##############################################################################
Application/LEACH instproc calculateCoverage {} {
  global opt ns_ PI tracefd_1

  $self instvar neighborID_ neighborX_ neighborY_ id_ neighborOff_
  $self instvar full_ fill_
  $self instvar angle_l_0_ angle_l_1_ angle_h_0_ angle_h_1_ cover_ids_
  $self instvar this_full_ this_fill_ this_angle_l_ this_angle_h_

  #initialize
  for {set i 0} {$i < 4} {incr i 1} {
     set full_($i) 0
     set fill_($i) 0
     set angle_l_0_($i) 0
     set angle_h_0_($i) 0
     set angle_l_1_($i) 0
     set angle_h_1_($i) 0
  }
  set cover_ids_ ""

  #begin to compute
  set neighbor_num [llength $neighborID_]
  set i 0
  while { $i<$neighbor_num && ![$self isFull?] } {
      set neighbor [lindex $neighborID_ $i]
      if {[lsearch $neighborOff_ $neighbor] < 0} {
          #get neighbor(i) position
          set n_x [lindex $neighborX_ $i]
          set n_y [lindex $neighborY_ $i]

          #get my position
          set my_x [$self getX]
          set my_y [$self getY]

          #compute relative distance
          set r_x [expr $n_x - $my_x]
          set r_y [expr $n_y - $my_y]
          set distance [expr sqrt([expr [expr $r_x * $r_x] + [expr $r_y * $r_y]])]

          #compute magnitude of AOA
          set range [expr asin([expr $distance / [expr 2 * $opt(sensing_range)]])]
          set range [expr [expr $PI /2] - $range]
          set range [toDegree $range]

          #compute phase of AOA
          set direction [expr atan([expr $r_y / $r_x])]
          set direction [toDegree $direction]
          if { $r_x <0 && $r_y >0 } {
                set direction [expr $direction + 180]
          } elseif { $r_x >0 && $r_y <0 } {
                set direction [expr $direction + 360]
          } elseif { $r_x <0 && $r_y <0 } {
                set direction [expr $direction + 180]
          } elseif { $r_x ==0 && $r_y >0 } {
                set direction 90
          } elseif { $r_x ==0 && $r_y <0 } {
                set direction 270
          } elseif { $r_x >0 && $r_y ==0 } {
                set direction 0
          } elseif { $r_x <0 && $r_y ==0 } {
                set direction 180
          }

         #split central angle into 4 parts
         $self split $direction $range

         #merge central angle
         $self merge

         set cover_ids_ [lappend cover_ids_ $neighbor]
   }
         incr i
  }
}
```

```
################################################################################
# Application/LEACH instproc split
################################################################################
Application/LEACH instproc split {d r} {
  global tracefd_1
  $self instvar this_full_ this_fill_ this_angle_l_ this_angle_h_
  $self instvar id_

  for {set j 0} {$j < 4} {incr j 1} {
        set this_full_($j) 0
        set this_fill_($j) 0
        set this_angle_l_($j) 0
        set this_angle_h_($j) 0
  }
  if { $d >=0 && $d <90 } {
       if { $d >= $r } {
             set this_angle_l_(0) [expr $d - $r]
             set this_fill_(0) 1
        } else {
             set this_angle_l_(0) 0
             set this_fill_(0) 1
              set this_angle_l_(3) [expr 360 + $d - $r]
             set this_angle_h_(3) 360
             set this_fill_(3) 1
        }
       if { 90 >= [expr $d + $r] } {
             set this_angle_h_(0) [expr $d + $r]
             set this_fill_(0) 1
       } else {
             set this_angle_h_(0) 90
             set this_fill_(0) 1
              set this_angle_h_(1) [expr $d + $r]
             set this_angle_l_(1) 90
             set this_fill_(1) 1
        }
  } elseif { $d >=90 && $d <180 } {
       if { [expr $d - $r] >= 90 } {
             set this_angle_l_(1) [expr $d - $r]
             set this_fill_(1) 1
        } else {
             set this_angle_l_(0) [expr $d - $r]
             set this_angle_h_(0) 90
             set this_fill_(0) 1
              set this_angle_l_(1) 90
             set this_fill_(1) 1
        }
       if { 180 >= [expr $d + $r] } {
             set this_angle_h_(1) [expr $d + $r]
             set this_fill_(1) 1
        } else {
             set this_angle_h_(1) 180
             set this_fill_(1) 1
              set this_angle_h_(2) [expr $d + $r]
             set this_angle_l_(2) 180
             set this_fill_(2) 1
        }
  } elseif { $d >=180 && $d <270 } {
       if { [expr $d - $r] >= 180 } {
             set this_angle_l_(2) [expr $d - $r]
             set this_fill_(2) 1
        } else {
             set this_angle_l_(1) [expr $d - $r]
             set this_angle_h_(1) 180
             set this_fill_(1) 1
              set this_angle_l_(2) 180
             set this_fill_(2) 1
        }
       if { 270 >= [expr $d + $r] } {
             set this_angle_h_(2) [expr $d + $r]
             set this_fill_(2) 1
        } else {
             set this_angle_h_(2) 270
             set this_fill_(2) 1
              set this_angle_h_(3) [expr $d + $r]
             set this_angle_l_(3) 270
             set this_fill_(3) 1
        }
  } elseif { $d >=270 && $d <360 } {
       if { [expr $d - $r] >= 270 } {
             set this_angle_l_(3) [expr $d - $r]
             set this_fill_(3) 1
        } else {
             set this_angle_l_(2) [expr $d - $r]
             set this_angle_h_(2) 270
             set this_fill_(2) 1
              set this_angle_l_(3) 270
             set this_fill_(3) 1
        }
       if { 360 >= [expr $d + $r] } {
             set this_angle_h_(3) [expr $d + $r]
             set this_fill_(3) 1
        } else {
             set this_angle_h_(3) 360
             set this_fill_(3) 1
              set this_angle_h_(0) [expr [expr $d + $r] - 360]
             set this_angle_l_(0) 0
             set this_fill_(0) 1
        }
  }
  if { $this_fill_(0) == 1 && $this_angle_l_(0) == 0 && $this_angle_h_(0) == 90} {
       set this_full_(0) 1 }

  if { $this_fill_(1) == 1 && $this_angle_l_(1) == 90 && $this_angle_h_(1) == 180} {
       set this_full_(1) 1  }

  if { $this_fill_(2) == 1 && $this_angle_l_(2) == 180 && $this_angle_h_(2) == 270} {
       set this_full_(2) 1  }

  if { $this_fill_(3) == 1 && $this_angle_l_(3) == 270 && $this_angle_h_(3) == 360} {
       set this_full_(3) 1  }
}
```

```
############################################################################
# Application/LEACH instproc merge
############################################################################
Application/LEACH instproc merge {} {
  global tracefd_1
  $self instvar this_full_ this_fill_ this_angle_l_ this_angle_h_
  $self instvar full_ fill_ angle_l_0_ angle_l_1_ angle_h_0_ angle_h_1_
  $self instvar id_

  for {set i 0} {$i < 4} {incr i 1} {
        set max [expr 90 + [expr 90 * $i]]
        set min [expr 90 * $i]
      if {$full_($i) == 1} {
              #do nothing
        } elseif {$this_fill_($i) == 0} {
            #do nothing
        } elseif {$this_full_($i) == 1} {
            set full_($i) 1
              set angle_l_0_($i) $min
                set angle_h_0_($i) $max
                set fill_($i) 1
        } elseif {$this_fill_($i) == 1 && $fill_($i) == 0} {
              set angle_l_0_($i) $this_angle_l_($i)
                set angle_h_0_($i) $this_angle_h_($i)
                set fill_($i) 1
      } elseif {$fill_($i) == 1} {
              if {$angle_l_0_($i) == $min} {
                  if {$this_angle_l_($i) == $min } {
                        if {$this_angle_h_($i) > $angle_h_0_($i) } {
                              set angle_h_0_($i) $this_angle_h_($i)
                        }
                    } elseif {$angle_h_0_($i) >= $this_angle_l_($i)} {
                            set full_($i) 1
                            set angle_l_0_($i) $min
                            set angle_h_0_($i) $max
                            set fill_($i) 1
                  } else {
                              set fill_($i) [expr $fill_($i) + 1]
                              set angle_l_1_($i) $this_angle_l_($i)
                              set angle_h_1_($i) $this_angle_h_($i)
                  }
            } else {
                  if {$this_angle_h_($i) == $max } {
                        if {$this_angle_l_($i) < $angle_l_0_($i) } {
                              set angle_l_0_($i) $this_angle_l_($i)
                        }
                    } elseif {$angle_l_0_($i) <= $this_angle_h_($i)} {
                            set full_($i) 1
                            set angle_l_0_($i) $min
                            set angle_h_0_($i) $max
                            set fill_($i) 1
                  } else {
                              set fill_($i) [expr $fill_($i) + 1]
                              set angle_l_1_($i) $angle_l_0_($i)
                              set angle_h_1_($i) $angle_h_0_($i)
                              set angle_l_0_($i) $this_angle_l_($i)
                              set angle_h_0_($i) $this_angle_h_($i)
                  }
            }
      } else {
                  if {$this_angle_l_($i) == $min} {
                        if {$this_angle_h_($i) >= $angle_l_1_($i)} {
                            set full_($i) 1
                            set angle_l_0_($i) $min
                            set angle_h_0_($i) $max
                            set fill_($i) 1
                        } elseif {$this_angle_h_($i) > $angle_h_0_($i)} {
                              set angle_h_0_($i) $this_angle_h_($i)
                        }
                  } else {
                        if {$this_angle_l_($i) <= $angle_h_0_($i)} {
                              set full_($i) 1
                            set angle_l_0_($i) $min
                            set angle_h_0_($i) $max
                            set fill_($i) 1
                        } elseif {$this_angle_l_($i) < $angle_l_1_($i)} {
                              set angle_l_1_($i) $this_angle_l_($i)
                        }
                  }
      }
  }
}
############################################################################
# Application/LEACH instproc isFull?
############################################################################
Application/LEACH instproc isFull? {} {
  $self instvar full_

  if {$full_(0) == 1 && $full_(1) == 1 && $full_(2) == 1 && $full_(3) == 1}
{
      return 1
  }
  return 0
}
```

Appendix-2: Confidence level of the experimental results

Table 7.1: Neighbor_number_based node scheduling scheme: confidence range of off-duty percentage vs.K at 95% confidence level

N	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$
300	95-97	92-95	89-92	87-90	83-86	80-84	77-82	74-79	72-76	68-74
250	94-96	91-94	87-91	84-88	79-84	77-81	72-77	69-74	66-72	63-69
200	93-96	89-92	84-88	80-84	75-81	71-78	67-72	63-68	58-64	53-60
150	91-94	84-90	78-84	73-79	67-74	61-68	56-64	51-60	46-54	38-48
100	86-92	76-85	69-77	60-70	52-61	45-56	35-48	28-41	24-35	17-30
50	70-84	52-72	42-68	28-44	18-36	10-26	2-22	0-16	0-12	0-8

Table 7.2: Neighbor_number_based node scheduling scheme: confidence range of coverage loss percentage vs.K at 95% confidence level

N	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$
300	8-33	5-10	0-3	0-2	0-1	0-0	0-0	0-0	0-0	0-0
250	8-33	4-10	0-3	0-1	0-1	0-0	0-0	0-0	0-0	0-0
200	9-34	4-11	0-3	0-1	0-0	0-0	0-0	0-0	0-0	0-0
150	8-33	6-10	0-4	0-1	0-1	0-0	0-0	0-0	0-0	0-0
100	8-33	6-11	0-4	0-1	0-0	0-0	0-0	0-0	0-0	0-0
50	7-35	5-10	0-3	0-1	0-0	0-0	0-0	0-0	0-0	0-0

Table 7.3: Neighbor_number_based node scheduling scheme: confidence range of average sensing degree vs.K at 95% confidence level

N	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$
300	0-1	1-2	2-3	2-3	3-4	4-5	5-6	6-7	6-7	7-8
250	0-1	1-2	2-3	2-3	3-4	4-5	5-6	6-7	6-7	7-8
200	0-1	1-2	2-3	2-3	3-4	4-5	5-6	6-7	6-7	7-8
150	0-1	1-2	2-2	2-3	3-4	4-5	5-6	5-6	6-7	7-8
100	0-1	1-2	2-2	2-3	3-4	4-5	5-6	5-6	6-7	7-8
50	0-1	1-2	2-2	2-3	3-4	3-4	4-5	4-5	4-5	4-5

Table 7.4: Nearest_neighbor_based node scheduling scheme: confidence range of off-duty percentage vs.D at 95% confidence level

N	$D = 0.020$	0.035	0.063	0.116	0.209	0.396	0.761	1.474	3.007	6.619
300	0-0	0-0	0-0	0-1	0-2	2-5	6-13	26-34	65-71	91-94
250	0-0	0-0	0-0	0-1	0-3	1-5	5-11	21-30	61-68	88-93
200	0-0	0-0	0-0	0-1	0-2	0-4	3-10	15-27	53-62	86-91
150	0-0	0-0	0-0	0-1	0-1	0-3	2-7	11-22	45-56	82-88
100	0-0	0-0	0-0	0-1	0-1	0-3	0-6	6-18	30-45	74-83
50	0-0	0-0	0-0	0-0	0-2	0-4	0-6	2-14	12-30	52-68

Table 7.5: Nearest_neighbor_based node scheduling scheme: confidence range of coverage loss percentage vs.D at 95% confidence level

N	$D = 0.020$	0.035	0.063	0.116	0.209	0.396	0.761	1.474	3.007	6.619
300	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-4
250	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-5
200	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-6
150	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-5
100	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-5
50	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-1	0-8

Table 7.6: Nearest_neighbor_based node scheduling scheme: confidence range of average sensing degree vs.D at 95% confidence level

N	$D = 0.020$	0.035	0.063	0.116	0.209	0.396	0.761	1.474	3.007	6.619
300	30-32	30-31	30-31	30-31	30-31	29-31	27-29	20-22	8-10	1-2
250	25-26	25-26	25-26	25-26	25-25	24-26	23-24	18-20	8-9	1-2
200	20-21	20-21	20-21	20-21	20-21	20-21	18-20	15-17	7-9	1-2
150	15-16	15-16	15-16	15-16	15-16	14-16	14-15	11-13	6-8	1-2
100	10-10	10-10	10-10	10-10	10-10	9-10	9-10	8-9	5-7	1-2
50	4-5	4-5	4-5	4-5	4-5	4-5	4-5	4-4	3-4	1-2

Table 7.7: Neighbor_number_probability_based node scheduling scheme: confidence range of off-duty percentage vs.K at 95% confidence level

N	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$
300	95-99	92-96	88-94	84-92	81-89	79-88	75-85	73-83	69-81	67-77
250	94-98	90-96	85-93	82-91	80-87	73-85	72-81	69-79	66-75	61-72
200	93-98	87-95	81-91	78-88	74-85	70-81	62-78	58-72	56-69	52-65
150	90-97	82-93	76-89	68-84	63-79	60-75	54-69	44-64	44-58	35-54
100	87-97	77-89	62-82	56-77	49-69	39-62	32-52	26-47	21-38	13-33
50	76-92	54-80	32-66	24-52	12-38	8-32	0-20	0-12	0-8	0-6

Table 7.8: Neighbor_number_probability_based node scheduling scheme: confidence range of coverage loss percentage vs.K at 95% confidence level

N	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$
300	19-74	6-38	1-17	0-12	0-6	0-5	0-3	0-2	0-1	0-1
250	20-70	4-39	1-20	0-12	0-7	0-4	0-3	0-2	0-1	0-1
200	18-75	5-40	1-20	0-14	0-7	0-5	0-2	0-2	0-1	0-1
150	17-68	4-38	1-23	0-12	0-6	0-5	0-3	0-2	0-1	0-1
100	16-67	5-38	1-17	0-14	0-7	0-3	0-2	0-3	0-3	0-1
50	14-67	5-38	1-19	0-12	0-6	0-2	0-2	0-1	0-1	0-0

Table 7.9: Neighbor_number_probability_based node scheduling scheme: confidence range of average sensing degree vs.K at 95% confidence level

N	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$	$K=6$	$K=7$	$K=8$	$K=9$	$K=10$
300	0-1	1-2	1-3	2-4	3-5	3-5	4-7	5-7	6-9	7-9
250	0-1	0-2	1-3	2-4	3-5	3-6	4-7	6-8	6-9	7-8
200	0-1	0-2	1-3	2-4	3-5	3-6	4-9	5-8	6-8	7-9
150	0-1	1-2	1-3	2-4	3-5	3-5	4-6	5-7	6-8	6-9
100	0-1	1-2	1-3	2-4	3-5	3-5	4-6	5-7	6-8	7-8
50	0-1	1-2	1-3	2-4	3-4	3-4	4-5	4-5	4-5	4-5

Table 7.10: Nearest_neighbor_probability_based node scheduling scheme: confidence range of off-duty percentage vs.D at 95% confidence level

N	$D = 0.020$	0.035	0.063	0.116	0.209	0.396	0.761	1.474	3.007	6.619
300	0-0	0-0	0-0	0-1	0-2	1-5	6-12	26-36	65-76	91-96
250	0-0	0-0	0-0	0-1	0-1	0-4	5-11	21-33	60-73	88-95
200	0-0	0-0	0-0	0-1	0-2	0-3	4-10	17-28	49-67	87-94
150	0-0	0-0	0-0	0-1	0-1	0-3	0-9	9-23	43-58	80-93
100	0-0	0-0	0-1	0-1	0-2	0-3	0-4	5-19	28-49	74-88
50	0-0	0-0	0-2	0-2	0-2	0-2	0-4	2-12	14-34	52-78

Table 7.11: Nearest_neighbor_probability_based node scheduling scheme: confidence range of coverage loss percentage vs.D at 95% confidence level

N	$D = 0.020$	0.035	0.063	0.116	0.209	0.396	0.761	1.474	3.007	6.619
300	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-1	1-30
250	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-1	2-30
200	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-1	4-30
150	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-1	3-38
100	0-0	0-0	0-0	0-0	0-0	0-0	0-0	0-1	0-2	3-38
50	0-0	0-0	0-0	0-0	0-0	0-0	0-1	0-2	0-4	6-35

Table 7.12: Nearest_neighbor_probability_based node scheduling scheme: confidence range of average sensing degree vs.D at 95% confidence level

N	$D = 0.020$	0.035	0.063	0.116	0.209	0.396	0.761	1.474	3.007	6.619
50	4-5	4-5	4-5	4-5	4-5	4-5	4-5	4-5	3-4	1-2
100	10-10	10	9-10	9-10	9-10	9-10	9-10	8-9	5-7	1-2
150	15-16	15-16	15-16	15-16	15-16	14-15	14-15	11-13	6-8	1-2
200	20-21	20-21	20-21	20-21	20-21	20-21	18-20	14-17	6-9	1-2
250	25-26	25-26	25-26	25-26	25-26	24-26	22-24	17-20	7-10	1-2
300	30-32	30-32	30-32	30-32	29-30	29-30	27-29	19-22	7-10	1-2

Bibliography

[1] *network simulator-2*, http://www.isi.edu/nsnam/ns/.

[2] A.Beaufour, M.Leopoid, and P.Bonnet, *Smart-tag based data dissemination*, In proceeding of the first ACM workshop on Wireless Sensor Networks and Application (2002), 68–76.

[3] A.Cerpa and D.Estrin, *Ascent: Adaptive Self-Configuring Sensor Networks Topologies*, In Proceedings of the Twenty First International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002) (2002), 23–27.

[4] I. F. Akyildiz, W. Su, Y.Sankarasubramaniam, and E.cayirci, *Wireless Sensor Networks: A Survey*, Computer Networks (March 2002), 393–422.

[5] A.Manjeshwar and D.P.Agrawal, *Teen:a routing protocol for enhanced efficiency in wireless sensor networks*, In Proceeding of 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing (2001).

[6] A.Manjeshwar and D.P.Agrawal, *Aptten: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks*, In Proceeding of International Parallel and Distributed Processing Symposium (2002).

[7] A.Manjeshwar, Q.Zeng, and D.P.Agrawal, *An analytical model for information retrieval in wireless sensor networks using enhanced apteen protocol*, IEEE Transactions on parallel and distributed systems (December 2002), no. 12.

[8] A.Nasipuri and K.Li, *A directionality based location discovery scheme for wireless sensor networks*, n Proceeding of First ACM Wireless Sensor Network and Application Workshop (2002), 105–111.

[9] A.Savvides, C.Han, and M.B.Strivastava, *Dynamic fine-grained localization in ad-hoc networks of sensor*, In Proceeding of 7th ACM International Conference on Mobile Computing and Networking (2001).

[10] A.Srivastava, J.Sobaje, M.Potkonjak, and M.Sarrafzadeh, *Optimal Node Scheduling for Effective Energy Usage in Sensor Networks*, IEEE Workshop on Integrated Management of Power Aware Communications, Computing and Networking (2002).

[11] B.Chen, K. Jamieson, H. Balakrishnana, and R. Morris, *Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks*, In Proceeding of ACM International Conference on Mobile Computing and Networking(MOBICOM'01) (2001), 85–96.

[12] B.Ko and D.Rubenstein, *A greedy approach to replicated content placement using graph coloring*, In Proceeding of SPIE ITCom Conference on Scalability and Traffic Control in IP Networks II (July 2002).

[13] B.Ko, K.Ross, and D.Rubenstein, *Conserving Energy in Dense Sensor Networks via Distributed Scheduling*, http://www-sop.inria.fr/mistral/personnel/K.Avrachenkov/WiOpt/PDFfiles/ko44.pdf.

[14] B.Krishnamachari, D.Estrin, and S.Wicker, *Modelling data-centric routing in wireless sensor networks*, In Proceeding of IEEE Infocom (2002).

[15] B.Liu and D.Towsley, *On the Coverage and Detectability of Large-scale Wireless Sensor Networks*, In Proceeding of WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (2003).

[16] B.Liu and D.Towsley, *A study of the coverage of large-scale sensor networks*, Submitted to the first IEEE mobile ad hoc and sensor conference (2004).

[17] B.M.Blum, T.He, S.Son, and J.A.Stanknovic, *Igf. a state-free robust communication protocol for wireless sensor networks*, http://www.cs.virginia.edu/ th7c/paper/IGF.pdf.

[18] J. Cartigny, D. Simplot, and I. Stojmenovic, *Localized energy efficient broadcast for wireless networks with directional antennas*, Proc. IFIP Mediterranean Workshop on Ad Hoc Networks Med-Hoc (2002).

[19] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J.Zhao, *Habitat Monitoring: Application Driver for Wireless Communications Technology*, In Proceeding of ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean Costa Rica, 3–5.

[20] C.Florens and R.McEliece, *Scheduling Algorithms for Wireless Ad-Hoc Sensor Networks*, In Proceeding of Lee Center Workshop (October 2002).

[21] C.Intanagonwiwat, D.Estrin, and R.Govindan, *Impact of network density on data aggregation in wireless sensor networks*, In Proceedings of the 22nd International Conference on Distributed Computing Systems (2002).

[22] D.Braginsky and D.Estrin, *Rumor routing algorithm for sensor networks*, In proceeding of the first ACM workshop on Wireless Sensor Networks and Application (2002).

[23] B. Deb, S. Bhatnagar, and B. Nath, *A topology discovery algorithm for sensor networks with applications to network management*, Technical Report. DCS-TR-441, Dept of Computer Science, Rutgers University (May 2001).

[24] D.Ganesan, R.Govindan, S.Shenker, and D.Estrin, *Highly-resilient, energy-efficient multipath routing in wireless sensor networks*, Mobile Computing and Communications Review (2002), no. 2.

[25] L. Doherty, L. El Ghaoui, and K. S. J. Pister, *Convex position estimation in wireless sensor networks*, In Proceeding of Infocom (2001).

[26] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, *Next Century Challenges: Scalable Coordination in Sensor Networks*, In Proceeding of ACM International Conference on Mobile Computing and Networking (MOBICOM'99) (1999), 263–270.

[27] F.Ye, A.Chen, S.Liu, and L.Zhang, *A scalable solution to minimum cost forwarding in large sensor networks*, In Proceeding of Tenth International Conference on Computer Communications and Networks (2001), 304–309.

[28] F.Ye, S. Liu, , and L.Zhang, *Gradient broadcast: A robust, long-lived sensor network*, Http://irl.cs.ucla.edu/papers/grab-tech-reports.ps (2001).

[29] G.Gupta and M.Younis, *Fault-tolerant clustering of wireless sensor networks*, In Proceeding of IEEE Wireless Communications and Networks Conference (2003).

[30] W. R. Heizelman, A. Chandrakasan, , and H. Balakrishnan, *Energy-efficient communication protocol for wireless micro sensor networks*, In IEEE Proceedings of the Hawaii International Conference on System Sciences (2000), 1–10.

[31] W. R. Heizelman, A. Chandrakasan, and H. Balakrishnan, *Energy-Efficient Communication Protocol for wireless Micro sensor networks*, In IEEE Proceedings of the Hawaii International Conference on System Sciences (2000), 1–10.

[32] H.O.Sanli and X.Cheng, *Eqos: an energy efficient Qos preserving protocol for heterogeneous wireless sensor networks*, submitted (2003).

[33] H.Zhang and J.C.Hou, *Maintaining Sensing Coverage and Connectivity in Large Sensor Networks*, Technical report UIUCDCS-R-2003-2351 (June 2003).

[34] F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic, *A dominating sets and target radius based activity scheduling and minimum energy broadcast protocol for ad hoc and sensor networks*, The Third Annual Mediterranean Ad Hoc Networking Workshop Med-Hoc-Net (2004), 351–359.

[35] F. Ingelrest, D. Simplot-Ryl, and I. Stojmenovic, *Target transmission radius over lmst for energy-efficient broadcast protocol in ad hoc networks*, IEEE International Conference on Communications ICC (2004).

[36] C. Intanagonwiwat, R. Govindan, and D. Estrin, *Directed diffusion: A scalable and robust communication paradigm for sensor networks*, In Proceeding of ACM International Conference on Mobile Computing and Networking(MOBICOM'00) (2000), 56–57.

[37] I.Stojmenovic and X.Lin, *Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks*, IEEE Transactions on parallel and distributed systems (2001), no. 10.

[38] J.Albowicz, A.Chen, and L.Zhang, *Recursive position estimation in sensor networks*, In Proceeding of IEEE International Conference on Network Protocols (ICNP'01) (2001).

[39] J.Elson and D.Estrin, *Time synchronization for wireless sensor networks*, In Proceedings of the 2001 International Parallel and Distributed Processing Symposium,workshop on parallel and distributed computing issues in wireless networks and mobile computing (April 2001).

[40] J.Kulik, W.Rabiner, and H.Balakrishnana, *Adaptive protocols for information dissemination in wireless sensor networks*, In Proceeding of ACM International Conference on Mobile Computing and Networking(MOBICOM) (1999).

[41] J.Liu, P.Cheung, L.Guibas, and Zhao F, *A Dual-Space Approach to Tracking and Sensor Management in Wireless Sensor Networks*, In Proceeding of First ACM Wireless Sensor Network and Application Workshop (October, 2002), 131–139.

[42] J.Wu, B.Wu, and I.Stojmenovic, *Power-aware Broadcasting and Activity Scheduling in Ad Hoc Wireless Networks Using Connected Dominating Sets*, In Proceeding of IASTED International Conference on Wireless and Optical Communication (2002).

[43] J.Wu and H.Li, *On Calculating Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks*, In Proceeding Of the 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (1999), 7–14.

[44] J.Wu, M.Gao, and I.Stojmenovic, *On Calculating Power-aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks*, In Proceeding Of International Conference on Parallel Processing (2001), 346–356.

[45] J. Kahn, R. Katz, and K. Pister, *Next Century Challenges: Mobile Networking for Smart Dust*, In Proceeding of ACM International Conference on Mobile Computing and Networking (MOBICOM'99) (1999), 271–278.

[46] K.Akkaya and M.Younis, *A survey on routing protocols for wireless sensor networks*, Ad hoc networks journal (2004).

[47] K.Chakrabarty, S.S.Iyengar, H.Qi, and E.Cho, *Grid coverage for surveillance and target location in distributed sensor networks*, IEEE Transactions on computers (December 2002), 1448–1453.

[48] L.B.Ruiz A.A Loureiro L.F.Vieira, M.a.Vieira and A.O.Fernandes, *Applying Voronoi Diagram to Schedule Nodes in Wireless Sensor Networks*, Submitted to the Special Issue of MONET on Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks. (2003).

[49] L. Li and J.Y.Halpern, *Minimum-energy mobile wireless networks revisited*, In proceeding of IEEE International Conference on Communications ICC'01 (June 2001).

[50] X. Lin and I. Stojmenovic, *Location-based localized alternate, disjoint and multi-path routing algorithms for wireless networks*, Journal of Parallel and Distributed Computing (2003), no. 1, 22–32.

[51] J. Mirkovic, G. P. Venkataramani, S. Lu, and L. Zhang, *A self-organizing approach to data forwarding in large-scale sensor networks*, In Proceedings of ICC (2001).

[52] N.Bulusu, *Self-configuring localization systems*, Ph.D Thesis,University of California, Los Angeles (2002).

[53] N.Li, J.C.Hou, and L.Sha, *Design and analysis of an mst-based topology control algorithms*, In Proceeding of Twenty Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) (2003).

[54] F.J. Ovalle-Martinez, I. Stojmenovic, F. Garcia-Nocetti, and J. Solano-Gonzalez, *Finding minimum transmission radii and constructing minimal spanning trees in ad hoc and sensor networks*, In Proceeding of the third Workshop on Efficient and Experimental Algorithms (2004).

[55] R.C.Shah and H.M.Rabaey, *Energy aware routing for low energy ad hoc sensor networks*, In Proceeding of IEEE Wireless Communications and Networking Conference (2002).

[56] R.D.Poor, *Gradient routing in ad hoc networks*, http://www.media.mit.edu/pia/Research/ESP/texts/poorieeepaper.pdf (2001).

[57] V. Rodoplu and T.H.Meng, *Minimum energy mobile wireless networks*, IEEE J.Selected Areas in Communications (August 1999), no. 8.

[58] Narayanan Sadagopan, Bhaskar Krishnamachari, and Ahmed Helmy, *Active query forwarding in sensor networks*, Ad hoc networks journal (2003).

[59] S.Bandyopadhyay and E.J.Coyle, *Minimizing Communication Costs in Hierarchically Clustered Network of Wireless Sensors*, In Proceeding of IEEE Wireless Communication and Network Conference (2003).

[60] S.Bhatnagar, B.Deb, and B.Nah, *Service differentiation in sensor networks*, In Proceeding of the Fourth International Symposium on Wireless Personal Multimedia Communications (2001).

[61] S.Bhattacharya, H.Kim, S.Prabh, and T.Abdelzaher, *Energy-conserving data placement and asynchronous multicast in wireless sensor networks*, In Proceeding of the First International Conference on Mobile Systems, Applications and Services (MobiSys) (2003).

[62] S.De, C.Qiao, and H.Wu, *Meshed multipath routing: An efficient strategy in sensor networks*, In Proceeding of IEEE Wireless Communications and Networks Conference (2003).

[63] S.Dulman, T. Nieberg, J.Wu, and P.Havinga, *Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks*, In Proceeding of IEEE Wireless Communication and Network Conference (2003).

[64] M. Seddigh, J. Solano, and I. Stojmenovic, *Rng and internal node based broadcasting in one-to-one wireless networks*, ACM Mobile Computing and Communications Review (2001), no. 2, 37–44.

[65] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A.Chandrakasan, *Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks*, In Proceeding of ACM Special Interest group on Mobility of systems users, data, and computing(SIGMOBILE'01) (2001), 272–286.

[66] S.Lindsey and C.S.Raghavendra, *Pegasis: Power-efficient gathering in sensor information systems*, In Proceeding of International Conference on Communications (2001).

[67] S.Meguerdichian, F.Koushanfar, M.Potkonjak, and M.Srivastava, *Coverage Problems in Wireless Add-hoc Sensor Networks*, In Proceeding of IEEE Infocom (April 2001), 1380–1387.

[68] S.Meguerdichian, F.Loushanfar, G.Qu, and M.Potkonjak, *Exposure in wireless ad-hoc sensor networks*, In Proceeding of ACM International Conference on Mobile Computing and Networking(MOBICOM'01) (July 2001), 139–150.

[69] S.Meguerdichian and M.Potkonjak, *Low power 0/1 coverage and scheduling techniques in sensor networks*, UCLA Technical Reports 030001 (January 2003).

[70] S.Ni, Y.Tseng, Y.Chen, and J.Sheu, *The broadcast storm problem in a mobile ad hoc network*, In Proceeding of ACM International Conference on Mobile Computing and Networking(MOBICOM'99) (August 1999).

[71] S.Slijepcevic and M.Potkonjak, *Power efficient organization of wireless sensor networks*, In Proceeding of IEEE International Conference on Communications ICC'01 (June 2001).

[72] I. Stojmenovic, M. Seddigh, and J. Zunic, *Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks*, IEEE Transactions on Parallel and Distributed Systems (2002), no. 1, 14–25.

[73] T.Clouqueur, V.Phipatanasuphorn, P.Ramanathan, and K.K.Saluja, *Sensor Deployment Strategy for Target Detection*, In Proceeding of First ACM Wireless Sensor Network and Application Workshop (October, 2002), 42–48.

[74] T.He, J.A.Stankovic, C.Lu, and T.Abdelzaher, *Speed: A stateless protocol for real-time communication in sensor networks*, In Proceeding of 23rd International Conference on Distributed Computing Systems (2003).

[75] D. Tian and N.D. Georganas, *A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks*, In processing of ACM wireless sensor network and application workshop 2002 (September 2002).

[76] Toussaint, *The relative neighborhood graph of a finite planar set*, Pattern Recognition (1980), no. 4, 261–268.

[77] T.Yan, T.He, and J.A.Stankovic, *Differentiated Surveillance for Sensor Networks*, In proceeding of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003) (November 2003).

[78] X. Wang, G.Xing, Y.Zhang, C.Lu, R.Pless, and C.D.Gill, *Intergrated Coverage and Connectivity Configuration in Wireless Sensor Networks*, In proceeding of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003) (November 2003).

[79] R. Wattenhofer, L.Li, P.Bahl, and Y. M.Wang, *Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks*, In Proceeding of Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001) (2001).

[80] Y. Xu, J. Heidemann, and D. Estrin, *Geography-informed Energy Conservation for Ad Hoc Routing*, In Proceeding of ACM International Conference on Mobile Computing and Networking(MOBICOM'01) (2001), 70–84.

[81] Y. Xu, J. Heidemann, and D. Estrin, *Adaptive Energy-Conserving Routing for Multihop Ad hoc Networks*, Technical Report 527, USC/ISI. http://www.isi.edu/ johnh/PAPERS/Xu00a.html. Page accessed on October 2002 (Oct.2000), 70–84.

[82] X.Y.Li, G.Calinescu, and P.Wan, *Distributed construction of planar spanner and routing for ad hoc wireless networks*, In proceeding of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) (2002).

[83] F. Ye, G. Zhong, S. Lu, and L. Zhang, *Energy Efficient Robust Sensing Coverage in Large Sensor Networks*, Technical Report, http://www.cs.ucla.edu/ yefan/coverage-tech-report.ps (Page accessed in October 2002).

[84] Y.Yu, R.Govindan, and D.Estrin, *Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks*, Technical Report UCLA/CSD-TR-01-0023,UCLA Computer Science Dept. (2001).