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

Dazhi Yang
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.C.S. (Master of Computer Science)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Event-Driven Enhanced Coverage for Sensor Networks
TITRE DE LA THÈSE / TITLE OF THESIS

Prof. Jiying Zhao
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

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

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Chung-Hong Lung

Adulmotaleb El-Saddik

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

EVENT-DRIVEN ENHANCED COVERAGE FOR SENSOR NETWORKS

by

Dazhi Yang

A thesis submitted to
the Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of
the requirements for the degree of

Master of Computer Science

Ottawa-Carleton Institute of Computer Science
School of Information Technology and Engineering
University of Ottawa

Ottawa, Ontario, Canada

April 2006

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: 978-0-494-18481-3
Our file Notre référence
ISBN: 978-0-494-18481-3

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

Abstract

In recent years, wireless sensor networks are getting lots of attention due to their extensive applications, such as environment monitoring, battlefield surveillance, and intelligent home. However, they are characterized by extremely limited resources, such as energy and computing capability. To address this, my thesis develops a novel scheduling algorithm called EDEC (Event-Driven Enhanced Coverage) specifically for scheduling sensing units. The EDEC reduces the sensing redundancy by separating the monitoring phase and event observing phase, decreasing sensing overlap among densely-deployed sensors, and employing Busy-Tone technology. In the meanwhile, the original sensing capability of a sensor network, i.e. coverage degrees, can be maintained. The experimental results show that the EDEC can greatly prolong the network lifetime without decreasing the network capability, and the load can be evenly distributed to all nodes. In addition, the EDEC is flexible enough to be integrated with many existing communication protocols without too many changes to their original designs.

Acknowledgement

First of all, I would like to thank my supervisor, Professor Jiying Zhao, for his encouragement and guidance during the research and writing of this thesis. He presented a lot of patience, and gave me great instructions and comments on the research.

I also want to express my greatest debt and gratitude to my wife and my son, Fan and Eagle. Without their continuous supports and encouragements, this work would not have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, the continuing improvements in VLSI (Very Large Scale Integration), MEMS (Micro-Electro-Mechanical System) and wireless radio technologies leverage the emergence of the tiny wireless networked sensors which combine low power signal processing, low power computation, and low power, low cost wireless networking capability in a compact system. These sensors usually depend on battery to operate and are equipped with wireless radios to be able to communicate with each other [1][2][3].

1.1 Wireless Sensors

A sensor is any device that maps a physical quantity from the environment to a quantitative measurement [4]. Compared to the traditional sensor, a modern sensor is usually considered to have the following features: embedded microcontrollers, microprocessors, and analog-to-digital and digital-to-analog converters (ADCs and DACs), the ability of digital communication [5]. IEEE defines a smart sensor as a sensor "that provides functions beyond those necessary for generating a correct representation of a sensed or controlled quantity. This function typically simplifies the integration of the transducer

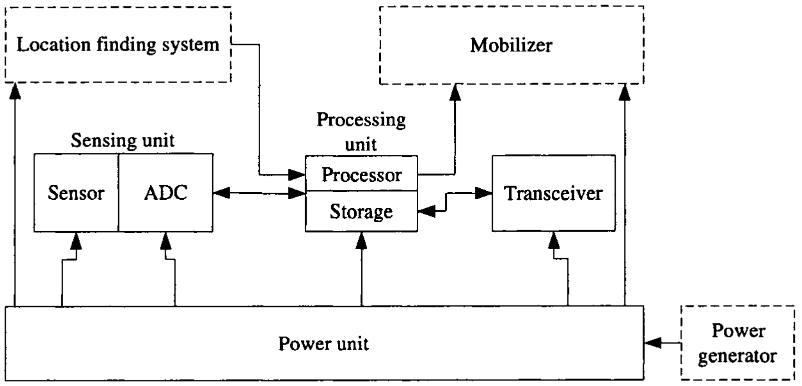


Figure 1.1: The structure of a sensor node [1].

into applications in a networked environment" [6]. The most basic capability of a modern sensor (or so-called smart sensor) hence is the ability to communicate in non-analog fashion.

Figure 1.1 gives a common view of the structure of modern sensors. Generally, sensors are equipped with sensing unit, processing unit, transmitting unit and power unit. In some applications, sensors are also equipped with location finding system (LFS), mobilizer, and power generator [1]. LFS makes sensors aware of geographic information, mobilizer makes sensors to be able to move around the interesting phenomena, and power generator makes sensors to be able to replenish their energy.

Figure 1.2 shows a sensor called "μAMPS-I DSP module" which is developed by MIT μAMPS Project. While it is small in size, it can be integrated to accomplish complex tasks as Figure 1.3 presents.

1.2 Wireless Sensor Networks (WSN)

Wireless sensor networks are to deploy tens, hundreds, or even thousands of sensors into the environment to observe the interesting phenomena. These sensors are usually able

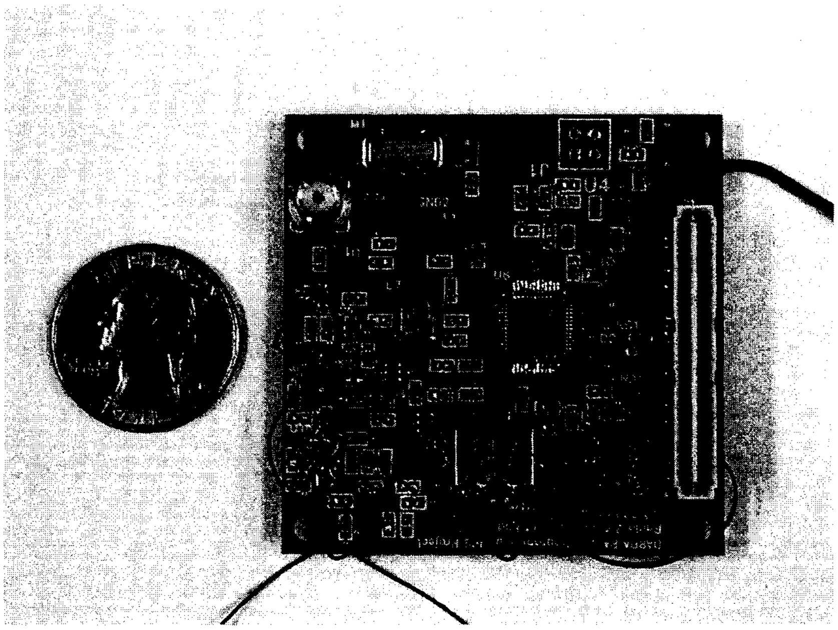
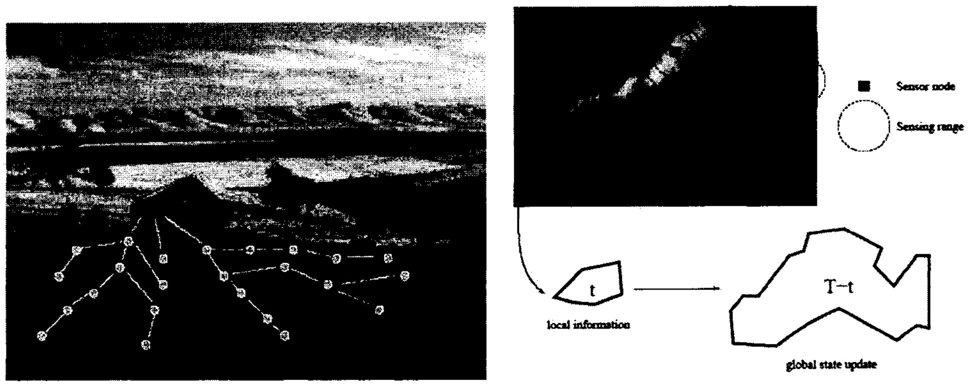


Figure 1.2: A small sensor node [7].

to communicate among themselves for data fusion and data aggregation in wireless fashion and route the processed and compressed data to BS (Base Station). As the previous section says, networking is the most important capability for a modern sensor. The reasons behind this and the motivations to network these sensors mainly include but are not limited to: [8]

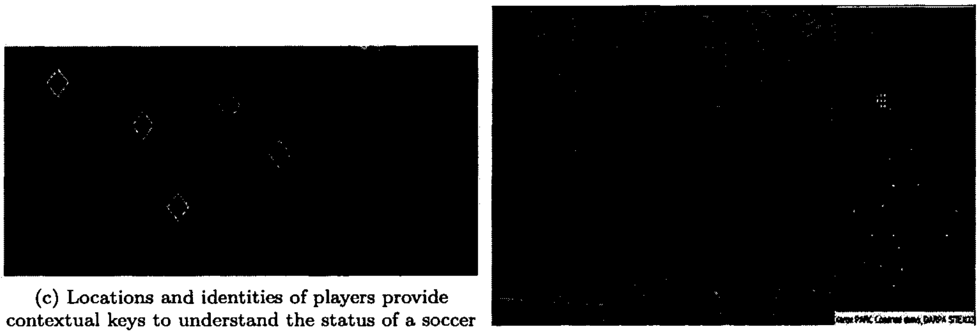
1. Cost saving: Simplified service, operation and maintenance, reduced wiring and rapid start up can be expected. Elimination of large, unwieldy bundles of cables and the associated mess is an obvious advantage.

2. Reliability: A complex, expensive and big sensor could be easily broken and stop providing services due to equipment failure, destructive natural phenomena, or premeditated attacks; however individual sensor's failure would not affect the overall task of a sensor network. This is one of the merits of distributed systems.



(a) Precision agriculture: sensors detect temperature, light levels and soil moisture.

(b) Wireless sensors measure the amount of chemical plume.



(c) Locations and identities of players provide contextual keys to understand the status of a soccer game even without seeing the ball position.

(d) Acoustic multi-target tracking.

Figure 1.3: A variety of sensor network applications.

3. More accurate measurement: Sensors in a network can be deployed multi-dimensionally in everywhere of an interesting area so that providing a full-coverage with accurate and precise measurements because each sensor delivers sensing data just around its vicinity.

Figure 1.4 gives a common scenario that sensor networks are organized. Sensor nodes are deployed in the target area to observe interesting events. They exchange the detected information in some way while doing data fusion and data aggregation so that data overlap and implosion [9] can be avoided and data amount decreases for saving the bandwidth. Then sensors route processed data to the Sink by either multi-hop way or

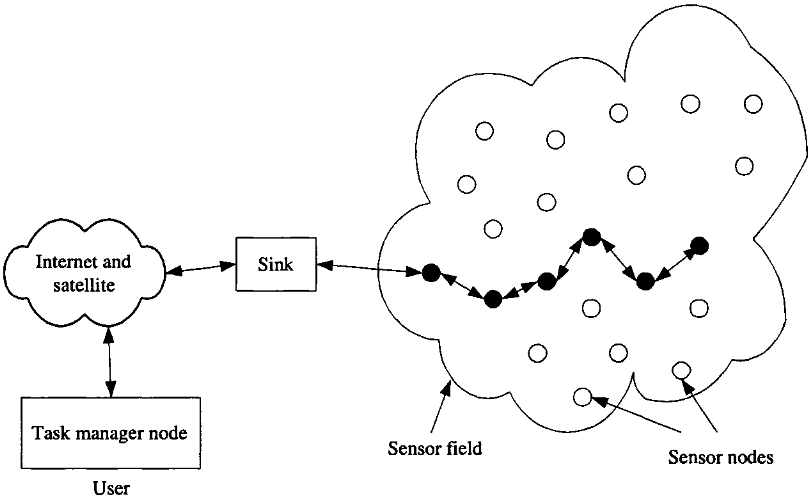


Figure 1.4: The sample structure of a sensor network.

direct transmission. Even if sensor networks use direct transmission, the transmission only happens between a few nodes and Sink because transmission between nodes and Sink are typically long distance hence energy intense. It is usually forbidden that all nodes directly transmit data to BS.

1.3 Applications and Scenarios

Sensor networks can be applied to a variety of applications. Actually, it induces a new era of ubiquitous computing in everywhere around us: everyday life, smart environment and so on.

For example, WSNs have been deployed for many environmental monitoring applications, which involve collecting data over time across a physical space. In [10], scientists have been monitoring the microclimate throughout the volume of redwood trees, helping form a sample of entire forests. Redwood trees are so large that entire ecosystems exist within their physical envelope. Traditionally, scientists have measured

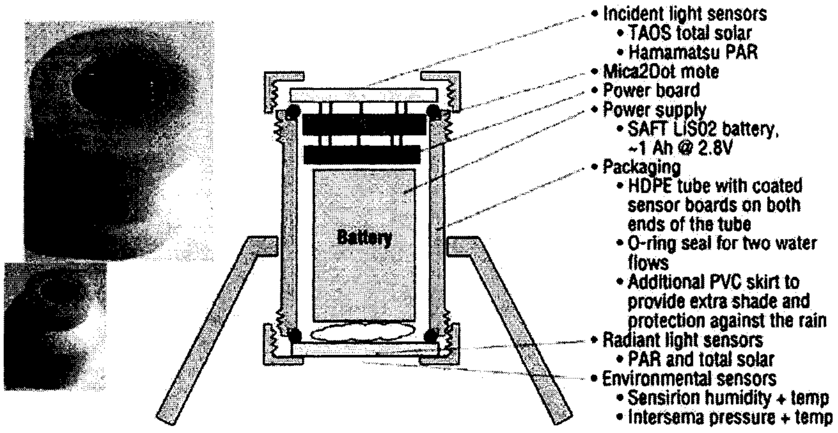


Figure 1.5: Wireless sensor node for environmental monitoring. An entire wireless microweather station fits in a tube about the size of a film canister.

ecosystem by moving a suite of instruments, weighing around 30 lbs, up and down a tree with serial cables.

However, WSN makes this much simpler and easier. Figure 1.5 shows a modern sensor used for environmental monitoring, which is about the size of a film canister. There are different sensor units for light, solar radiation, relative humidity, barometric pressure, and temperature. The weather-protected center of the tube contains a small computer, data storage, battery, and low-power radio to collect data, process it, and route information among the nodes to the Base Station. This provides a cost-effective means of obtaining simultaneous measurements at many points in the tree.

Figure 1.6 shows a temperature profile over three days, collected from 16 nodes at four elevations in a 35-meter high tree.

Another type of sensor network applications is object tracking. For example, protection of personnel has a high priority in peacekeeping missions. When a military unit establishes a base, they want to set up a security system as quickly as possible. Currently the army has different sensor systems, including footstep detectors, a battle-

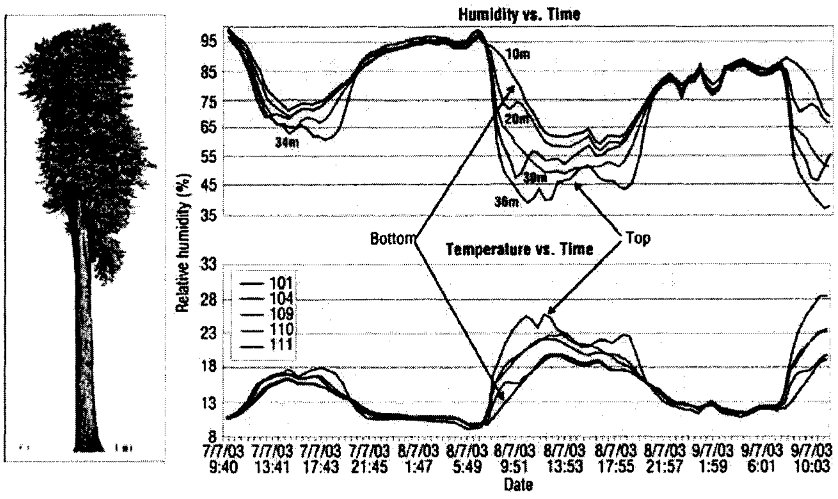


Figure 1.6: WSN climate data. The WSN samples climate data every five minutes and computes an average temperature at each elevation.

field radar and trip wires. These systems have several disadvantages including single points of failure and constant personnel attention. Thus, there is a need for a system that is easy to deploy and does not require an existing infrastructure or constant checkup by humans. More specifically, this system should report the location of the object and possibly other information about a detected object, such as speed and direction. Information about the detected object should be acquired and shared among the sensor nodes. This shared information will be used to generate precise information about the object and to improve the reliability of the object detection and classification (Individual sensor's reading might be inaccurate or even wrong due to environmental interferences).

Paper [11] describes an object tracking application for the Ministry of Defense. A passive infrared (PIR) sensor is chosen to detect objects, especially humans. This PIR sensor is connected to a Mica2 node, a commercially available sensor node. The PIR sensor is able to detect humans and provide an estimation of the direction of movement. There are software modules for the PIR sensor, detecting movement, sending messages (multicast), and a top-level module for the tracking algorithm. These modules are written in nesC for the TinyOS operating system, that runs on the Mica2 nodes. Experiments with the object tracking system show that the system can track an object, calculate its speed, and report this information to a base station. The speed calculation is relatively accurate, but can be influenced by the orientation of the sensor.

Besides environment monitoring and object tracking, sensor networks can also be applied to fields such as [1][3][12]:

- Health (e.g. vital sign monitoring)
- Home (e.g. motion detection)
- Manufacturing (e.g. assembly line fault-detection)
- Entertainment (e.g. virtual gaming)
- Digital lifestyle (e.g. parking spot tracking)

These applications share common features while they also have unique characteristics. In the following paragraph, we will classify them according to different perspectives.

Sensor networks can be classified as static and dynamic. The redwood tree monitoring example is a static application of WSN. The phenomena, sensors themselves and observer (Base Station) are all static. Previous studies have shown that localized algorithms can be used in an effective way [12]. The sensors in localized algorithms communicate with nodes in their locality. An elected node relays a summary of the local observations to the observer, perhaps through one or more levels of hierarchy. Such algorithms extend the lifetime of the sensor network because they trade-off local computation for communication [4].

In dynamic sensor networks, either the sensors themselves, the observer, or the phenomenon are mobile. For example, a plane might fly over a field periodically to collect information from a sensor network. This is a mobile observer. Another example is traffic monitoring implemented by attaching sensors to taxis. This belongs to mobile sensors. The object tracking example is an instance of mobile phenomena.

Whenever any motion occurs, it may break the connection from the observer to the phenomenon. If a sensor which is part of the route between observer to phenomenon moves out of the range, either the observer or this concerned sensor must take the initiative to rebuild a new route. The observer-initiated approach is a reactive approach, where path recovery action is only taken after observer realizing a broken path. Another sensor-initiated approach is a proactive approach where path recovery operations begin in anticipation of a future broken path.

As the previous sections stated, communication is one of the most important factor in sensor network field. Generally, communication within a sensor network can be classified into two categories: application and infrastructure. Application communication relates to the transfer of sensed data (or information obtained from it) with the goal of informing the observer about the phenomena. Within application communication, there are two models: cooperative and non-cooperative. Under the cooperative sensor model, sensors communicate with other sensors to realize the observer interest. In-network data processing is an example of co-operative sensors [13][14][15]. Non-cooperative sensors do not cooperate for information dissemination.

Infrastructure communication refers to the communication specifically for configuring, maintaining, and optimizing operations. Because of the ad hoc nature of sensor networks, sensors must be able to discover paths to other sensors of interest and to the observer regardless of sensor mobility or failure.

As infrastructure communication represents the overhead of the protocol, it is important to minimize this communication while ensuring that the network can support efficient application communication. However, investing in infrastructure communication can reduce application traffic and optimize overall network operation under certain conditions.

On the other hand, sensor networks can be classified in terms of the data delivery required by the application (observer) interest as: continuous, event-driven, observer-initiated and hybrid. In the continuous model, the sensors communicate their data continuously at a pre-specified rate. The authors of paper [16] showed that clustering is most efficient for static networks where data is continuously transmitted. In the event-driven data model the sensors report information only if an event of interest occurs. In this case, the observer is interested only in the occurrence of a specific phenomenon or a set of phenomena. In the observer-initiated (or request-reply) model, the sensors only report their results in response to an explicit request from the observer (either directly, or indirectly through other sensors). Finally, the three approaches can coexist in the same network; we refer to this model as the hybrid model.

For any of the above models, we can classify the routing approach as: flooding (broadcast-based), unicast, or multicast.

It is likely that the interaction between the data delivery model from the application and the routing model employed by the network protocol will significantly impact the performance of the network [12]. For example, considering a scenario where a sensor network is deployed for intrusion detection, the data delivery model is event driven, if the network level routing model is flooding based, problems raise: (1) the probability of loss of critical information; and (2) the latency in event reporting [12].

From an operational perspective, it is interesting to see the relationship between the properties of ad hoc routing protocols and the sensor network classification. The more dynamic the network is, the better the reactive approaches are.

We believe that it is useful to decouple the application communication used for information dissemination from the infrastructure communication used to configure and optimize the network. This separation will aid network designers in selecting the appropriate sensor network architecture that will best match the characteristics of the communication traffic of a given application.

1.4 Design Analysis

While sensor networks share many features with ad hoc networks, researchers also found many distinctions between them as the research is going on. To deeply understand these distinctions will be very helpful in designing and developing a sensor network.

Table 1.1 shows some of the differences.

Table 1.1: The distinction between WSN and MANET.

Sensor Network	**Mobile Ad-hoc Network**
Often use broadcasting	End-to-end communication
Sensors are deployed to co-operate for one target	Ad-hoc generally does not support co-operative dissemination
Number of nodes is relatively high	Number of nodes is relatively low
Individual sensor evolves stronger in power, computational capacity, and memory but is still limited compared to desktop or handheld	Unlimited in power, computational capacity and memory
Low-level radio frequency communications (AM/FM)	Bluetooth, 802.11 and ultrawideband (UWB)
Flooding and other improved communication protocols	TCP (UDP) / IP communication protocols

Sensor networks often have several orders of magnitude higher number of nodes than in other networks. Their addresses are defined as location-based, or capability or attribute based. In order to prolong the network lifetime, sensor nodes are densely deployed. Sensors could be mobile. However, its mobility issue is much different than other ad hoc networks because observers are not interested in sensor mobility, instead, observers care about how the sensor mobility could influence observed phenomena. Sensor networks are often regarded as "active" network, because nodes not only relay the message but also make decision of route selection based on their own local knowledge. All these constraints and characteristics lead to a summary of design factors that are important to the following sections.

1. Fault Tolerance

The failure of sensor nodes should not affect the overall task of the sensor network. This is the reliability or fault tolerance issue. Fault tolerance is the ability to sustain sensor network functionalities without any interruption due to individual sensor node failure.

2. Scalability

The density can range from a few sensor nodes to a few hundred sensor nodes in a region, which can be less than 10 meters in diameter.

3. Production Costs

Since sensor networks consist of a large number of sensor nodes, the cost of a single node is very important to justify the overall cost of the network.

As a comparative example, Bluetooth radio system is less than US$10. The cost of a sensor node should be much less than US$1 in order for the sensor network to be feasible for many WSN applications [1].

4. Topology

We can examine the deployment and put topology in three phases:

- Pre-deployment and deployment: Sensor nodes can be thrown from planes or placed one by one by human or robot.
- Redeployment of additional nodes: Additional sensor nodes may be added into

the network due to replacement or task re-scheduling.

- Post-deployment: After deployment, topology may change due to sensor's position, reachability, available energy and malfunctioning.

Thus, sensor network design should be able to address dynamic topology issue.

5. Power Consumption

In many application scenarios, replenishment of power resources might be impossible. The lifetime of the whole sensor network shows a strong dependence on battery lifetime.

Power consumption can be classified into three categories: sensing, communication, and data processing. Some sort of trade-off among computation and communication should be considered because experiments show that the execution of 3,000 instructions by a sensor's processor consumes the same amount of energy as sending a bit over 100 meters by radio [12]. This trade-off encourages the design of most existing sensor network communicating protocols.

1.5 Contributions

This thesis develops a novel scheduling algorithm for wireless sensor networks. Its main contributions can be summarized as:

1. It separates the tasks of optimizing network communication and scheduling sensing units such that the scheduling algorithm can be optimized specifically for sensing units.
2. It separates the monitoring phase and event observing phase such that coverage degree is low during the monitoring phase where it can be as high as original during the event observing phase.

3. A communication module implemented by busy-tone (BT) technology has been introduced specifically for scheduling sensing units. Therefore, the EDEC proposed in this thesis is flexible enough to be integrated with many existing communication protocols of wireless sensor networks.
4. A coverage calculation rule and the operation of such rule have been introduced such that the original coverage degree can be preserved even though a number of nodes turn off their sensing units.

1.6 Publication Generated from the Thesis Work

1. D. Yang, D. Wang, and J. Zhao, Event-driven enhanced coverage for sensor networks, Proc. Wireless Networks and Emerging Technologies (WNET 2006), Banff, Alberta, Canada, July 3-5, 2006.

1.7 Thesis Organization

The rest of the thesis is organized as follows. Several important topics – sensing coverage, network lifetime, and routing protocols – will be discussed in Chapter 2. Chapter 3 will propose the EDEC (Event-Driven Enhanced Coverage) algorithm. Chapter 4 will introduce the simulation models and give the experimental results and evaluations. Chapter 5 will conclude the thesis and point out our future research.

Chapter 2

Related Work

In this chapter, we will introduce several important topics which draw the most attention in recent years in the field of wireless sensor networks.

2.1 Coverage

Currently, many new algorithms have been proposed to try to set up sensor networks to provide good services for observing the physical phenomena. But, how well are these networks observing the physical phenomena? Coverage is one of the most important metrics for measuring the service quality. Generally speaking, we all try to achieve the goal that each location in the physical space of interest is within the sensing range of at least one sensor.

The reasons why coverage is an important measurement are:

- Many applications need to achieve required coverage, or hope to know the current coverage;

  For example, the well-known problem, Art Gallery Problem, asks to determine the number of observers and their placement, necessary to cover an art gallery room such that every point is seen by at least one observer. Here, it requires 100% coverage. On the other hand, some applications may desire to obtain the current coverage so that they can know how much confidence they could have on the data provided by their sensor networks.

- Some applications require location-based data;

  For example, user may be interested in the geographic position where a specific event occurs. Coverage algorithms often contain the function to discover neighbor nodes with their relative distance and angular information. This is because coverage is intrinsically a global issue based on all nodes' location information [17].

- Coverage is the most important decisional factor for the scheduling algorithms;

  Many nowadays sensor networks take advantage of dense deployment to prolong the network lifetime. This is because the sensor node is easy to fail, and highly limited on energy. Thus, if they keep active all the time sensors will drain out their energy quickly, therefore stop service earlier. To densely deploy sensors and then correctly schedule them so that only a few of them are active during one period while others are sleeping to save energy, sensor network's lifetime will be prolong greatly. However, how to schedule the nodes is the key issue and coverage is the most important decisional factor for such scheduling algorithms.

- Randomly distribution issue impacts the coverage;

  Randomly distribution is one of the most attractive characteristics among modern sensor networks while it is also the biggest challenge which makes coverage problem complex. We could imagine how easy it would be if we can pre-determine each sensor's position to give a full coverage on the target region. However, many sensor networks are deployed to unreachable places, such as battlefield and seabed. We will have to distribute the sensors randomly. In this way, 100 percentage coverage is hardly achievable theoretically. However, by post-deploying extra sensors to the "vacant" area, full coverage can be obtained [17].

2.1.1 Classification of Coverage

While the topics of coverage problem can be extensive, the most discussed concerns from the literature can be classified in the following types: area coverage, point coverage, and barrier coverage [18] as Figure 2.1 shows.

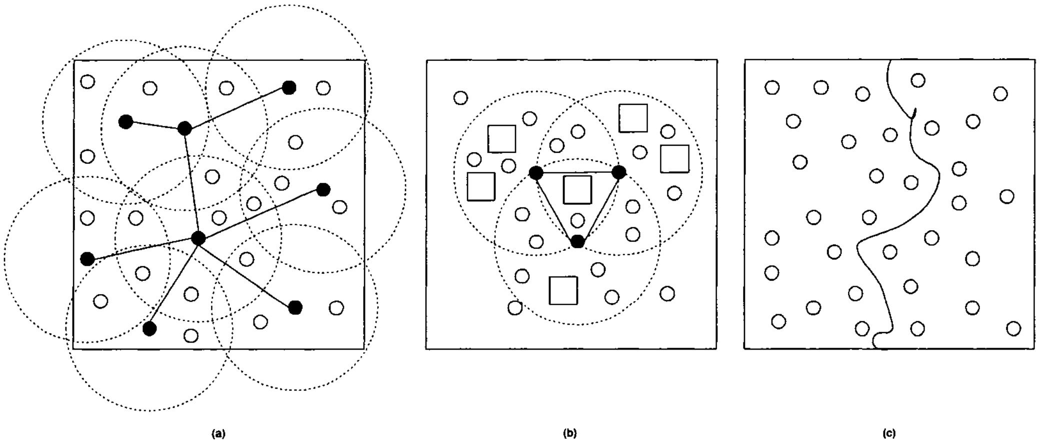


Figure 2.1: (a) area coverage, (b) point coverage, and (c) barrier coverage [18].

In Figure 2.1 (a), sensors are deployed to monitor the rectangle area. An event occurring at any point within that rectangle should be in at least one sensor's sensing range. This is called "area coverage". In Figure 2.1 (b), sensors are deployed to monitor a few points which are represented by small rectangles. As long as each of these small rectangles is in at least one sensor's sensing range, the effective sensing coverage is achieved. This is called "point coverage". In Figure 2.1 (c), sensors are deployed to monitor a barrier. Thus, sensors should be deployed along that barrier to assure that any point of that barrier is in at least one sensor's sensing range. This is called "barrier coverage".

Among different types of coverage, area coverage is the much more focused topic and we will concentrate on it, too.

2.1.2 Nodes Coverage

In paper [19], Carle et al. developed an algorithm, Area-Dominant Set (ADS), for coverage problem from an interesting perspective. The ADS is a subset of nodes which have the same coverage as what all nodes can achieve, and they can maintain the network connectivity, too. Apparently, the ADS does not improve the coverage of the given sensor network. Instead, it tries to save energy as much as possible without compromising the original coverage degree. Its simulation results show a big benefit of energy saving because of this redundancy prune. Besides, the authors' perspective is novel that a node is defined to be covered when its sensing sphere is covered by its neighbors. Thus, instead of computing how area is covered which might be complex geometrical problems, we just need to compute each node and its neighbors.

However, it only satisfies 1-degree coverage while some critical applications may require k-degree coverage for more reliability. (k-degree coverage is defined as for any point p in the target area, at least k nodes' sensing spheres cover it.) In addition, it tightly combined communicating unit and sensing unit together to produce the ADS. But we believe that sensing and communicating are different tasks. To separate them will increase the flexibility for either communicating design or sensing design.

2.1.3 Geometrical Computation of Coverage

In paper [17], Meguerdichian et al. gave an important proposal which cited the famous geometry concepts (Voronoi diagram and Delaunay triangulation) to compute the coverage.

In 2-Dimensional plane shown in Figure 2.2, the Voronoi diagram of a set of discrete points partitions the plane into a set of convex polygons such that all points inside a polygon are closest to only one point. This construction effectively produces polygons with edges that are equidistant from neighboring points. Figure 2.2 shows an example of a Voronoi diagram for a set of randomly placed points.

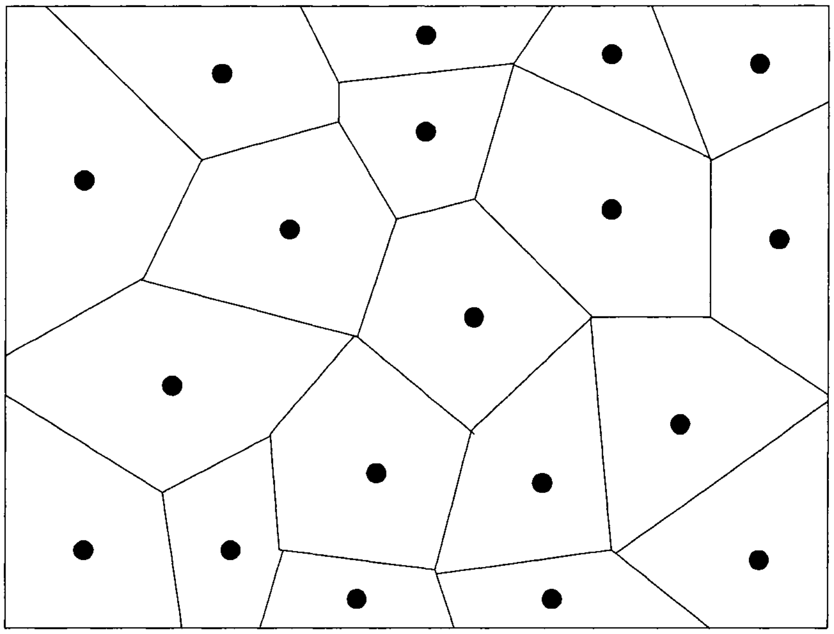


Figure 2.2: Voronoi Diagram of a set of randomly placed points in a plane [17].

Another structure that is directly related to Voronoi diagrams is the Delaunay triangulation. The Delaunay triangulation can be obtained by connecting the sites in the Voronoi diagram whose polygons share a common edge. It has been shown that among all possible triangulations, the Delaunay triangulation maximizes the smallest angle in each triangle. Also, neighborhood information can be extracted from the Delaunay triangulation since sites that are close together are connected. In fact the Delaunay triangulation can be used to find the two closest sites by considering the shortest edge in the triangulation.

The authors used these properties of the Voronoi diagram and Delaunay triangulation to solve for best and worst case coverages.

The authors defined the worst case as "Breach-Based Coverage" which means trying to find a $PATH$ through field A, with end-points I and F and with the property that for any point p on the $PATH$, the distance from p to the closest sensor is maximized. Then with Voronoi diagram, we can easily say that this path should be some edges of those convex polygons. Because if any point p on the $PATH$ deviates from Voronoi line segments, by definition, it must be closer to at least one sensor. Thus, the rest problem becomes how to choose one of the edges as the part of the $PATH$ in each step from I to F, which is apparently easier than constructing Voronoi diagram [17]. After finding this type of path, if new sensors can be deployed or existing sensors can move such that this path is no longer the biggest breach, then the worst-case coverage is improved.

On the other hand, if one tries to move in among the sensors while minimizing distance from sensors, one must attempt to travel along straight lines connecting sensor nodes. Since the Delaunay triangulation produces triangles that have minimal edge lengths among all possible triangulations, the best path must lie on the lines of the Delaunay triangulation of the sensors. Then we choose the path with biggest "support-weight" as the place where we should probably put more sensors. Here, "support weight" is the maximum distance from the closest sensors that an agent traveling on any path through the field A (from I to F) must encounter at least once. If additional sensors can be deployed or existing sensors can moved such that support weight is decreased, then the best-case coverage is improved.

Another elegance of this algorithm is that by analyzing a given field and selecting the proper number of sensor nodes, certain levels of coverage can be expected even when sensor deployment cannot be performed according to an exact plan.

As the authors claimed, the simulation results show that the conversion to graphs and weight assignments can be accomplished in linear time and therefore do not add any significant overhead to the computation.

2.1.4 Practice for "Critical Density Threshold"

In paper [20], Adlakha et al. evaluated the critical number of nodes required for target detection in a sensor network. The physical characteristics of sensors and target have been used to derive the R_1 and R_2, where R_1 is defined as the sensing range inside which any event occurring can be surely detected while R_2 is defined as the sensing range outside which any event occurring cannot be detected at all. Using these radii, the authors estimated the critical density for coverage in a sensor network.

The physical characteristics of sensors and target discussed in [20] are summarized as:

- Sensor model. The sensing ability of a sensor decreases as the distance between the target and the sensor increases, as shown in the following equation:

  $$S(s,p) = \frac{\lambda}{||s-p||^k} \tag{2.1}$$

  where the constant λ depends on the sensor calibration, s is the sensor, p is the target, k is the signal decay factor, and $||s-p||$ is the Euclidean distance between s and p.

- Exposure model. This basically means how to detect the signal. Some applications may declare detection when the received signals are above an pre-determined threshold; the other applications may declare the detection when the variation of detected signals is above an threshold.
- Detection model. An individual sensor can make a decision if it can announce the detection of an occurring event. Or, at least *k* sensors co-operate together to decide if an event is occurring.
- Target characteristics. The target can be static or moveable. And it can move in constant speed or variable speed, straight line movement or complicated moving tracks.

Another interesting discussion is the two new definitions of sensing radius proposed by the authors. One is "complete influence" radius within which targets will be surely detected; the other one is "no influence" radius beyond which targets will not be able to be detected at all. The authors derived an equation to obtain the desired radius given features listed above. This is valuable because the sensing range is the base to analyze coverage or design the density of sensors. (The EDEC develops new definitions about different sensing ranges which are originated from this.)

In short, the most valuable contributions of this paper are the definition of the two sensing radius and revealment that in practical applications the sensing radiuses could be different and influenced by giving specific characteristics as listed above.

2.1.5 Coverage Problem in 3-Dimension

In paper [21], Ravelomanana et al. focused on extending coverage problem to 3-Dimension space. A common model of sensor network could therefore be defined by a pair n and R, where n homogeneous sensor nodes are randomly thrown in a given region R of volume $V=|R|$, uniformly and independently. The authors then analyzed the neighborhood discovery and code assignment problem within 3-Dimension background. In particular, the authors considered how to design a poly-logarithmic protocol to allow the nodes to almost surely discover their neighborhoods asymptotically, and how to color the nodes of a graph in such a way that any two of adjacent nodes are assigned two different colors.

Moreover, the authors gave results about the relationships among the sensing range, transmission range, the number of nodes, and the volume of the region to be monitored. In particular, these results showed how to quantify the minimum and maximum degrees of a network.

There are several interesting and important characteristics of the sensor networks proposed in this paper:

1. Degrees of the reachability graph. The degree of a node v represents the number of its neighbors in the graph. So what is the required value of the transmission range R_{TRANS} to have a reachability graph with a given minimum or maximum degree?

2. Diameter. The hop distance between two nodes u and v is defined as the length of the shortest path (in terms of the number of hops) between them. The diameter of a graph is the maximum value of hop distances between any two pair of nodes. So what is the typical diameter of the reachability graph of a random sensor network?

3. Degrees of coverage. It is called k-degree coverage if every point of the considered region is covered by at least k nodes. Given the volume V of R, what is the required value of the sensing range R_{SENSE} to achieve a specified coverage degree k? More precisely, the authors were interested in values of the sensing range R_{SENSE} such that "each point p of the considered region R is covered by at least k spheres of radius R_{SENSE}". Similarly, degrees of connectedness are interesting depending on the

transmission range.

The authors gave a quantitative analysis on a randomly-distributed sensor network about coverage problem. However, the application-dependent design is encouraged in sensor networks because of their varieties in terms of physical characteristics of events, environments, and sensors themselves [20]. For example, given a specific sensing capacity, signal-noise ratio, exposure model, decision model, and target characteristic, the performance using application-specific protocols could be better than using comparably-generic algorithms in paper [21]. However, the most valuable part of this paper is to give a general guideline on designing a sensor network and present a general relationship among different factors quantitatively.

2.2 Lifetime of WSN

Besides the coverage, the lifetime of the network is another important metric for a WSN. Because the current wireless sensors usually operate on battery, they have extremely limited capacities. Furthermore, replenishing or replacing battery of sensors is often impossible or very costly in many scenarios such as battlefield surveillance and seabed monitoring. Therefore, sensors tend to run out of their energy shortly and stop services so that the whole network's service would be affected. In recent years, many papers discussed this issue and contributed different algorithms on how to resolve the energy bottleneck of a WSN.

In paper [22], Handy et al. proposed a new approach to define the lifetime of a wireless sensor network, which are FND (First Node Dies), HNA (Half of the Nodes Alive), and LND (Last Node Dies).

FND is used to denote that the network quality decreases as soon as the first node dies. HNA denotes an estimated value for the half-life period of a sensor network. LND gives an estimated value for the overall lifetime of a sensor network.

This approach presents a general perspective for the lifetime of a sensor network in terms of the number of the dead nodes. However, most sensor network applications do not care when an individual node dies or how many nodes are currently dead. Instead, they are often interested in the time when the network quality starts to decrease due to the dead nodes, or the time when the amount of the effective monitoring data starts to be lower than some pre-defined value due to the dead nodes.

In paper [4], Heinzelman designed a simulation model for a sensor network. In this model, the total initial energy of the network is known by deploying 100 nodes and assigning 2J [1] energy to each node. The unit energy for activities of sensors such as transmission and data processing are pre-defined. During the simulation, the energy is removed from a sensor whenever it is doing one of those activities. A sensor is dead when it runs out of its energy and then cannot do any activity. The whole simulation will end when the number of alive nodes are less than 5. By this model, the authors compared the two communication protocols – MTE and LEACH – in terms of the network lifetime given same total amount of energy. In addition, the energy efficiency can also be compared in the form of "pJ/bit".[2]

The second measurement, lifetime, may conflict with the first measurement, energy efficiency. For example, the LEACH algorithm assumes all nodes in one cluster observe the same event such that high ratio of data fusion can be achieved. While data fusion itself consumes energy, the greater saving can be obtained from avoiding long distant communicating with large amount correlated data. This way seems to compromise the energy efficiency from view of base station, but it actually gives much longer lifetime of the whole network.

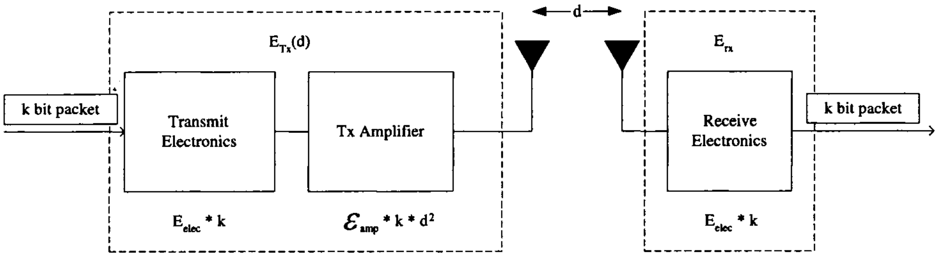
---

[1] J stands for Joule. pJ and nJ are also used in the rest of the thesis, where $1nJ = 10^{-9}J$ and $1pJ = 10^{-12}J$.

[2] pJ/bit represents the total energy consumed for 1 bit data.

The sources of energy consumption can be classified as sensing, communication, and data processing. Before we discuss their relationship, we look at the physical properties of radio communication.

Paper [4] proposed a simple model where the radio dissipates $E_{T_x-elec} = 50$ nJ/bit to run the transmitter or receiver circuitry and $E_{T_x-amp} = 100$ pJ/bit/m^2 for the transmit amplifier to achieve an acceptable performance as shown in Figure 2.3.



Operation	Energy Dissipated
Transmitter Electronics ($E_{Tx\text{-}elec}$) Receiver Electronics ($E_{Rx\text{-}elec}$) ($E_{Tx\text{-}elec} = E_{Rx\text{-}elec} = E_{elec}$)	50nJ/bit
Transmit Amplifier	100pJ/bit/m^2

Figure 2.3: Energy Dissipation.

In fact, it is very hard to try to model radio transmission in reality. The authors stated that when distance keeps increasing, radio signal attenuation becomes larger hence the amplifier may have to become much bigger, as shown in the following equations:

$$E_{Tx}(l,d) = E_{Tx-elec}(l) + E_{Tx-amp}(l,d) \tag{2.2}$$

$$E_{Tx}(l,d) = \begin{cases} lE_{elec} + l\varepsilon_{friss-amp}d^2 & d < d_{crossover} \\ lE_{elec} + l\varepsilon_{two-ray-amp}d^4 & d \geq d_{crossover} \end{cases} \tag{2.3}$$

$$E_{Rx}(l) = E_{Rx-elec}(l) = lE_{elec} \tag{2.4}$$

where E_{Tx} is the transmission energy, l is the size the message in terms of bits, d is the distance between transmitter and receiver, E_{elec} is the electronics energy depending on factors such as the digital coding and modulation, $\varepsilon_{friss-amp}$ and $\varepsilon_{two-ray-amp}$ are signal attenuation, and E_{Rx} is the receiving energy.

According to this model, we can conclude that in order to be energy-efficient, not only long distance but also number of sending or receiving need to be optimized. (We assume that the transmission is symmetric that transmission of Node A to B consumes same energy as Node B to A.)

Sometimes, we need to make a good balance between long distance communication and the number of communications. For example, when distance is 1 meter, according to this radio model, amplifier is 0.1 nJ and circuit including sending or receiving is about 50 nJ. The consumed energy of amplifier could be neglected. However, when distance increases to 100 meters, the amplifier becomes 1000 nJ and the circuit is still 50nJ.

In paper [23], Lindsey et al. proposed an interesting computation to induce the most optimal up-bound of sensor network's lifetime. In their protocol, PEGASIS, there are totally 100 nodes within the given area $50meters \times 50meters$, base station resides in remote about [25 meter, 150 meter], each node will be assigned to communicate with BS once every 100 rounds. Besides, node will send and receive locally once in each round. An initial energy in each node is set to be 0.25J. The long distant communication uses d^4 and intra-network communication uses d^2. Thus, the network can run 1100 rounds approximately. Considering some overhead such as data processing and static energy dissipation, the optimal should be less than 1000 rounds. By this model, PEGASIS simulation shows that it can achieve 800 rounds which are close to the optimal solution.

Finally, different sensor products have a little bit different physical properties which might influence the performance of the protocols. For example, UC-Berkeley developed three different sensors: Pico-Radio, Mote, and SmartDust.

2.3 Scheduling

The EDEC algorithm proposed in this thesis is actually a scheduling algorithm. As discussed in Section 2.2, power constraint can be the bottleneck of WSNs while units of a sensor, such as radio module and sensing device can be the main sources of energy consumption. Thus, we should consider to schedule different units so that some of them may turn off the circuits to avoid idle consumption.

Scheduling has been taken advantage of in earlier time by wireless network. For example, 802.11 MAC Layer has power save mode which enables the radio NIC to conserve battery power when there is no need to send data [24]. With power save mode on, the radio NIC indicates its desire to enter "sleep" state to the access point via a status bit located in the header of each frame. The access point takes note of each radio NIC wishing to enter power save mode, and buffers packets corresponding to the sleeping station. In order to still receive data frames, the sleeping NIC must wake up periodically (at the right time) to receive regular beacon transmissions coming from the access point. These beacons identify whether sleeping stations have frames buffered at the access point and waiting for delivery to their respective destinations. The radio NICs having awaiting frames will request them from the access point. After receiving the frames, the radio NIC can go back to sleep.

In WSNs, scheduling can be setup in different network layers. The following table lists a few existing scheduling algorithms running on Application Layer, Network Layer, and MAC Layer.

Table 2.1: Scheduling Algorithms in Different Layers [24][25][26].

Approach	**Protocol Layer**
TinyDB Duty Cycling	Application
S-MAC Scheduled Listening	MAC
Flexible Power Scheduling	Network
A Coverage-Preserving Node Scheduling Scheme	above Network
SPAN	between Network and MAC

When designing a scheduling algorithm, we should consider the following factors:

- The scheduler should allow as many nodes as possible to turn off working units such as their radio receivers most of the time, since an idle receive circuit can consume certain amount energy.

- On the other hand, it should assure as much capacity as the original network, such as the network connectivity or the coverage degree. This implies that enough nodes must stay awake to form a backbone.

- A good scheduling algorithm should inter-operate correctly with protocols either in other network layers or for other types of units. For example, it should not make many assumptions about the link layer's facilities for sleeping; it should work with any link layer that provides for sleeping and periodic polling, including 802.11's ad hoc power save mode.

In paper [27], Xu et al. proposed GAF protocol which is Geographic Adaptive Fidelity. GAF divided the network area into fixed zones to form a virtual grid. Inside each zone, nodes collaborate with each other to play different roles. For example, nodes will elect one sensor node to stay awake for a certain period of time, and then the rest will go to sleep. This node is responsible for monitoring the field and reporting data to the BS on behalf of the nodes in the zone. Hence, GAF conserves energy by turning off unnecessary nodes in the network without affecting routing requirement. Each node uses its GPS-indicated location to associate itself with a point in the virtual grid. There are three states defined in GAF: discovery, for determining the neighbors in the grid; active, for reflecting participation in routing; and sleep, for turning off the radio.

In paper [26], Tian et al. proposed a Coverage-Preserving Node Scheduling Scheme. This algorithm allows as many nodes as possible to be turned off in most of the time while it is still able to preserve the initial sensing coverage. It pursues ideal cases that a set of working nodes can cover the same monitored area as if all nodes are in working mode.

Most of the algorithms schedule the nodes from the communication perspective without considering the WSN's sensing coverage, while paper [26] uses the coverage as the main criteria. In fact, in wireless sensor networks, the main role of each node is sensing. Unusual event could happen at any time at any place. Therefore, if we turn off nodes only according to their participation in data forwarding, certain areas in the deploying area may become "blind points". Important events may not be detected. Paper [26] comes up with a solution to answer two questions which are basic and common in most scheduling algorithms. First, what is the rule that each node should follow to determine whether it should turn itself off or not? Second, when should nodes make such decisions?

In paper [24], Chen et al. proposed SPAN which adaptively elects "coordinators" from all nodes in the network. SPAN coordinators stay awake continuously and perform multi-hop packet routing within the ad hoc network, while other nodes remain in power-saving mode and periodically check if they should wake up and become a coordinator.

SPAN achieves four goals. First, it ensures that enough coordinators are elected so that every node is in radio range of at least one coordinator. Second, it rotates the coordinators in order to ensure that all nodes share the task of providing global connectivity roughly equally. Third, it attempts to minimize the number of nodes elected as coordinators, thereby increasing network lifetime without suffering a significant loss of capacity. Fourth, it elects coordinators using only local information in a decentralized manner.

However, SPAN focuses only on communication, and it does not consider the coverage at all. Thus, it is not suitable for sensor network applications with the monitoring coverage requirement.

2.4 Energy Efficient Routing Protocols

Routing is a challenging issue in WSNs. Routing protocols for WSNs need to setup paths between multi-nodes and BS while handling dynamic scenarios of sensor nodes, BS or the phenomenon as discussed in Section 1.3. The communication along these paths could be the biggest source of energy consumption. Therefore it caught considerable research attention.

Figure 2.4 presents an overview of the OSI layers which have been simplified and narrowed down because of sensor networks' specific features and functions. It is proved to be effective to make use of crossing layer interactions for many sensor network scenarios. For example, when application decides to enter into sleeping mode, it would be beneficial to notify power management such that power management can adjust the power saving schedule for lower layers such as network layer and link layer. Therefore, analysis on routing technique helps us design interaction between scheduling algorithms and network layer protocols.

In this section, we analyze different routing protocols in network layer, focusing on two points: energy efficiency and energy even distribution. Uneven energy consumption

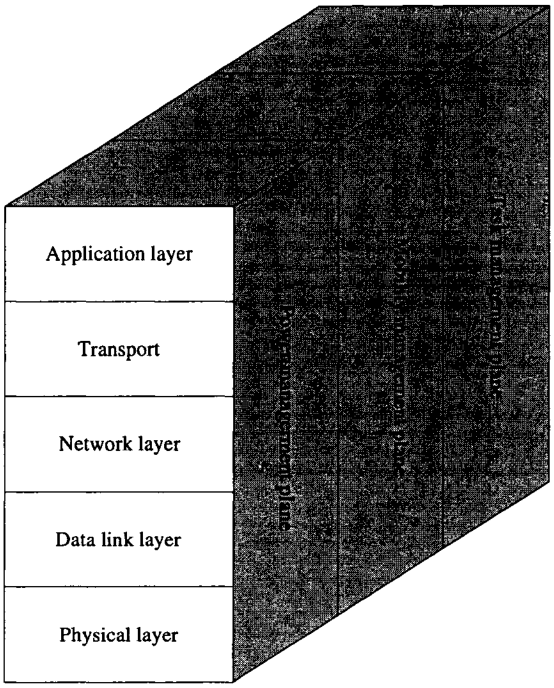


Figure 2.4: Protocol stack of WSN [1].

could make some sensor nodes die much faster even though all nodes could be deployed with the same amount of power. This often leads to break coverage of the whole sensor network. According to these perspectives, we have two criterions to route messages.

- Available power of each node;
- The energy required for transmission along the routes.

Therefore, we could have four ways to select a route as shown in Figure 2.5:

1. Maximum available power route. The route having greatest available energy will be the candidate. According to the energy even distribution principle, this route should be selected as it contains the greatest energy. In Figure 2.5, the route

between node T and Sink should be passing node C, node B, and node A, because their total available power ("PA") is the highest.

2. Minimum energy route. This route consumes the least energy along it. According to energy efficiency principle, this route should be selected as it consumes the least energy. In Figure 2.5, the route between node T and Sink should be passing node B and node A, because the total consumed power is the least (The sum of a_1, a_2, and a_7 is 3).

3. Minimum hop route. This route contains the least number of nodes along it. As mentioned in Section 2.2, it is energy inefficiency that too many nodes in a short path join routing, because sensor circuit running, radio transmitting and receiving are also sources of energy consumption. In Figure 2.5, the route between node T and Sink should be passing node D.

4. Maximum minimum available power node route. In this path, the least energy of a node along the path should be greater than any other nodes' energy in other paths. In Figure 2.5, there is no such a route between node T and Sink that satisfies this criterion.

These criterions are used in many existing power-saving protocols. In the following sections, we will introduce them based on the network structure: flat routing, hierarchical routing, and location-based routing [15].

2.4.1 Flat Routing

2.4.1.1 Flooding

Each node broadcasts the packets it received or sensed until the destination is arrived or the maximum hops are reached [28].

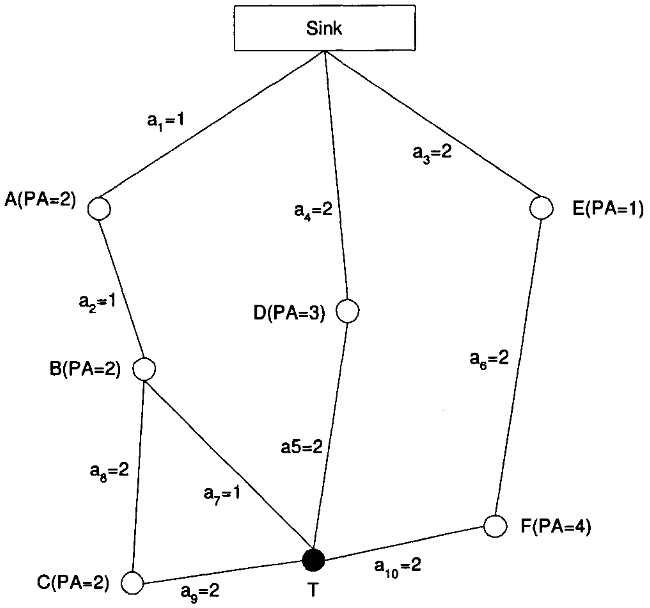


Figure 2.5: Power aware protocols.

Advantages of flooding:

- There is no need for topology maintenance and complex route discovery algorithm;
- Generally, it can ensure the packets receiving by destination.

Disadvantages of flooding:

- Implosion. One node often has to receive the same data packets from many different neighbors.
- Collision. Once an event is occurring, sensors around it start to transmit almost at the same time, which could result in serious collision.
- Overlap. Data packets from sensor nodes often have common contents when those sensors are observing the same area.

- Energy inefficiency. Because of the above disadvantages, energy is consumed inefficiently.

2.4.1.2 Gossiping

To avoid the weakness of flooding, gossiping makes each sensor node randomly choosing only one neighbor to receive the packet [1]. But this often results in taking longer time to propagate the message to all sensor nodes or arrive the destination.

2.4.1.3 Sensor Protocols for Information Via Negotiation (SPIN)

SPIN is a three-stage protocol as sensor nodes use three types of messages, ADV, REQ, and DATA, to communicate [9][28]. ADV is used to advertise new data, REQ is used to request data, and DATA is the actual message itself. In the very beginning, the BS sends queries to certain regions and waits for data from the sensors located in the selected regions. Since data is being requested through queries, attribute-based naming or meta-data is necessary to specify the properties of data. Figure 2.6 shows SPIN operation procedure. Node A starts by advertising its data to node B. Node B responds by sending a request to node A. After receiving the requested data, node B sends out advertisements to its neighbors, who in turn send requests back to node B.

SPIN can resolve the overlap and implosion issues in WSNs. Figure 2.7 shows the implosion issue where node D gets two copies from its neighbors node B and node C. As SPIN uses ADV and REQ as well as meta-data, node D will know that it already received the same data such that node D will not respond with REQ, therefore avoiding the implosion issue. Figure 2.8 presents the overlap issue where node C gets overlapping data from node A and node B because they have overlapping sensing area. As SPIN uses meta-data to identify different data, as long as the application-specific meta-data can be setup, the overlap issue can be resolved.

The other advantage of SPIN is that topological changes are localized since each node needs know only its one-hop neighbors. SPIN provides more energy savings than flooding, and meta-data negotiation could almost halve the redundant data. However, SPIN's data advertisement mechanism cannot guarantee delivery of data. For example, assuming that the nodes that are interested in the data are far away from the source node, and the nodes between source and destination nodes are not interested in that data, such data will not be delivered to the destination at all.

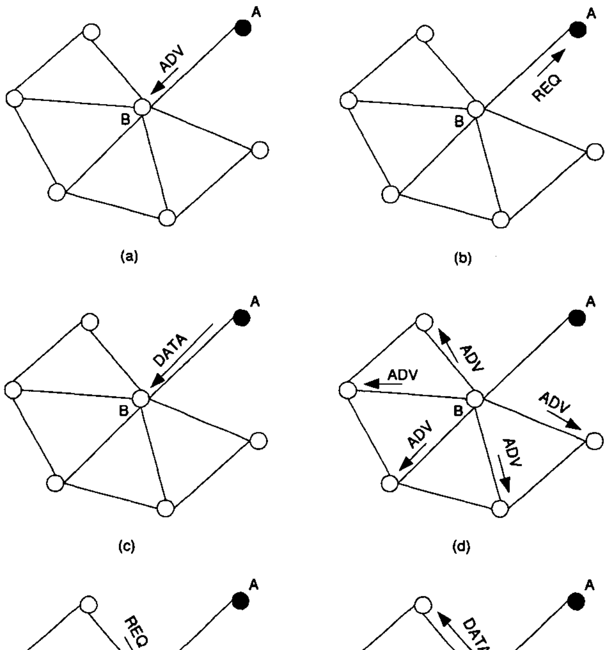


Figure 2.6: How SPIN spread the data from one node to the whole network.

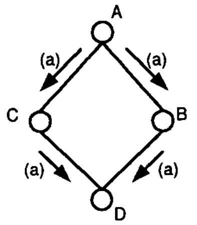


Figure 2.7: Implosion.

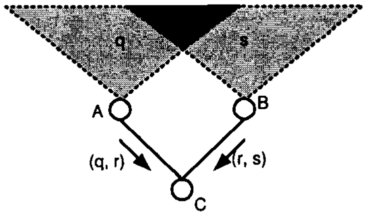


Figure 2.8: Overlap.

2.4.1.4 Directed Diffusion

Directed diffusion [14][29] is a novel data-centric routing protocol for sensor networks. It has following features: reinforcement of the empirically best path, and in-network data aggregation and caching. These features can enable highly energy-efficient and robust dissemination in dynamic sensor networks, while minimizing the node configuration.

Directed diffusion consists of several elements. Data is named using attribute-value pairs. A sensing task (or a subtask) is disseminated throughout the sensor network as an interest for named data. This dissemination sets up gradients within the network designed to "draw" events. Events start flowing towards the originators of interests along multiple paths. The sensor network reinforces one, or a small number of these paths (Shown in Figure 2.9).

Directed diffusion is different from SPIN because all communication in directed diffusion is neighbor to neighbor while each node has the capability to perform data aggregation and caching. Unlike SPIN, there is no need to maintain global network topology in directed diffusion. (SPIN assumes that BS could be any node within the network therefore the data has to be spread to all nodes. This is why global network topology has to be maintained; but directed diffusion only needs to keep the paths between source and BS.) However, directed diffusion may not be applied to applications that require continuous data delivery to the BS (For instance, environmental monitoring). This is because the query-driven on-demand data model may not help in such areas. Moreover, matching data to queries might require some extra overhead at the sensor nodes. This makes it unsuitable for one-time queries, as it is not worth setting up gradients for queries that use the path only once.

In directed diffusion, each node has same capacity to do aggregation, because the sources could be from different nodes. But in SPIN, the overlap data is simply cast aside in some nodes nearby source.

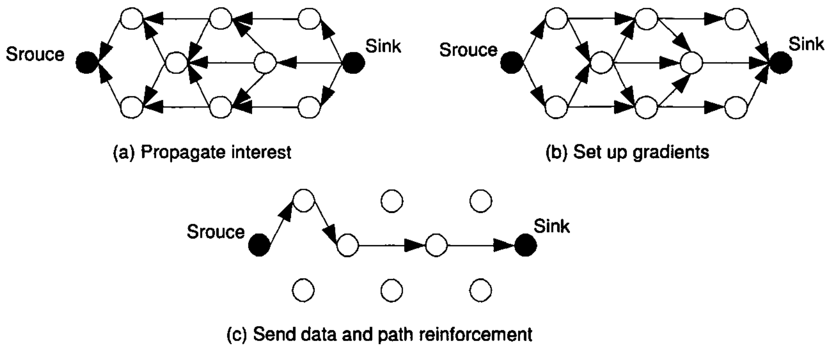


Figure 2.9: Directed Diffusion.

2.4.1.5 Minimum Cost Forwarding Algorithm for Large Sensor Networks (MCFA)

In paper [30], each node maintains the least cost estimate from itself to the sink node. When receiving a message, the node checks if it is on the least cost path between source node and sink. If so, it will forward the message by broadcasting. The idea behind this algorithm is somewhat similar to the natural gravity field that drives waterfalls from top of mountain to the ground. At each point water always flows along the shortest path. By this algorithm, in an initial phase, sink sends out an advertisement message with the cost set to zero. Each node sets cost as infinity. Then once receiving the advertisement message, the node checks if cost in that message plus cost of current link is smaller than its own cost. If so, the node will update the advertisement message with new cost as well as its own cost estimate and then broadcast the advertisement message. Finally, all nodes will get their own estimates of the least cost between themselves and sink.

The MCFA assures the energy efficiency that message always takes the shortest path. However, in some applications, it will break the load even distribution principle, because every node always takes the same path. When there is some hot spot where interesting events happen more frequently, nodes along some paths will die much faster. MCFA does not address this issue.

2.4.1.6 Other Flat-based Routing Protocols

In paper [31], the authors proposed another variant of directed diffusion, called gradient-based routing (GBR). The key idea in GBR is to memorize the number of hops when the interest is diffused through the whole network. As such, each node can calculate a parameter called the height of the node, which is the minimum number of hops to reach the BS. The difference between a node's height and that of its neighbor is considered the gradient on that link. A packet is forwarded on a link with the largest gradient.

In the GBR, three different data dissemination techniques have been discussed:

- A stochastic scheme, where a node picks one gradient at random when there are two or more next hops that have the same gradient;
- An energy-based scheme, where a node increases its height when its energy drops below a certain threshold so that other sensors are discouraged from sending data to that node;
- A stream-based scheme, where new streams are not routed through nodes that are currently part of the path of other streams.

The main objective of these schemes is to obtain balanced distribution of the traffic in the network, thus increase the network lifetime.

In paper [32], the network is treated as a huge distributed database system. The key idea is to use declarative queries in order to abstract query processing from the network layer functions, such as selection of relevant sensors. The BS is responsible for generating a query plan that specifies the necessary information about the data flow and in-network computation for the incoming query, and sends it to the relevant nodes.

There are some other flat routing protocols presented in [33][34].

2.4.2 Hierarchical Routing

Hierarchical or cluster-based routing methods have the advantages on scalability and energy efficient communication in WSNs. In a hierarchical architecture, higher-energy nodes can be used to process and send the information, while low-energy nodes can only be used to perform the sensing. The creation of clusters can also greatly contribute to overall system scalability and lifetime because lower energy consumption within a cluster is needed for routing as well as better data aggregation.

2.4.2.1 Low-Energy Adaptive Clustering Hierarchy (LEACH)

The LEACH [4][16][35] randomly chooses some sensor nodes as cluster heads. All other sensor nodes in the network select one of the heads as its own head. Instead of communicating with base station, sensor nodes communicate with their head nodes. Head nodes are responsible for relaying messages to base station (Shown in Figure 2.10). There are two phases in LEACH. In the initial phase, all sensor nodes are competing to be cluster heads. According to conditions such as available energy, a few of nodes are selected as heads and then each head node is broadcasting an advertisement globally. One node may receive advertisements from several head nodes and chooses one of them as its own head node and then informs that head node its decision. Finally, the head node will schedule time slots for all its member nodes for communication in a TDMA manner. The second phase is the steady phase. During this phase, nodes are constantly communicate with cluster head nodes and cluster head nodes communicate with BS. Generally, steady phase is much longer than initial phase. At the end of steady phase, the network re-starts the whole procedure. LEACH is energy-efficient, especially for static sensor network. The authors claimed that it could extend the life of sensor network by a factor of 8 compared to multi-hop routing protocols.

2.4.2.2 Threshold Sensitive Energy Efficient Sensor Network (TEEN)

The TEEN [36] employs the cluster formation strategy of LEACH. But it adopted a different way in data transmission phase. When the monitored value exceeds a threshold α, nodes just store the value and wait for their own time slots to transmit. If the monitored value increases above another threshold β, nodes will transmit the value

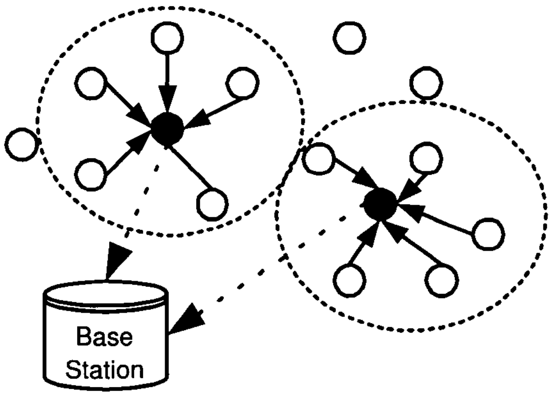


Figure 2.10: How the LEACH manages the communications between nodes and BS.

right away.

The APTEEN (Adaptive Periodic TEEN) [37] is another improvement for LEACH. It offers a lot of flexibility by allowing the user to set "count time" (the maximum time period between two successive reports sent by a node). Thus threshold values for energy consumption can be controlled by changing count time. The main drawback of this scheme is the additional complexity required to implement the threshold functions and count time.

2.4.2.3 Power-Efficient Gathering in Sensor Information Systems (PEGASIS)

PEGASIS [23] is an enhancement over the LEACH protocol. The basic idea of this protocol is that in order to extend network lifetime, nodes need only communicate with their closest neighbors, and they take turns in communicating with the BS. When the round of all nodes communicating with the BS ends, a new round starts. Unlike LEACH, PEGASIS avoids cluster formation and uses only one node in a chain to transmit to the BS instead of multiple cluster head nodes. To locate the closest neighbor node in PEGASIS, each node uses the signal strength to measure the distance to all neighboring nodes and then adjusts the signal strength so that only one node can be heard. A chain in PEGASIS will consist of those nodes that are closest to each other and form a path to the BS. The aggregated form of the data will be sent to the BS by any node in the chain, and the nodes in the chain will take turns sending to the BS. Its good performance is achieved through the elimination of the overhead caused by dynamic cluster formation in LEACH. Although the cluster forming overhead is avoided, PEGASIS still requires dynamic topology adjustment since a sensor node needs to know about the energy status of its neighbors in order to know where to route its data. Note also that PEGASIS introduces excessive delay for distant nodes on the chain. Finally, although in most scenarios sensors will be fixed or immobile as assumed in PEGASIS, some sensors may be allowed to move and hence seriously affect the protocol functionality.

2.4.2.4 Other Hierarchical Routing Protocols

There are many other hierarchical routing protocols such as minimum energy communication network (MECN) [38], and self-organizing protocol (SOP) [39].

The main idea of MECN is to find a sub-network that will have fewer nodes and require less power for transmission between any two particular nodes. In this way, global minimum power paths are found without considering all the nodes in the network. This is performed using a localized search for each node.

The SOP was proposed to build architecture to support heterogeneous sensors. Furthermore, these sensors can be mobile or stationary. Some sensors probe the environment and forward the data to a designated set of nodes that act as routers. Router nodes are stationary and form the backbone for communication. Collected data are forwarded through the routers to the more powerful BS nodes. Each sensing node should be able to reach a router in order to be part of the network.

2.4.3 Location-based Routing

Location-based routing makes sensor nodes addressed by means of their locations. The distance between neighboring nodes can be estimated on the basis of incoming signal strengths. Relative coordinates of neighboring nodes can be obtained by exchanging information between neighbors [40][41] or directly by communicating with a satellite using GPS if nodes are equipped with a small low-power GPS receiver [27].

2.4.3.1 SPAN

SPAN [24] is a position-based algorithm which selects some nodes as coordinators based on their positions. The coordinators form a network backbone that is used to forward messages. A node should become a coordinator if two neighbors cannot reach each other directly or via one or two coordinators (three-hop reachability). In the next chapter, we will introduce SPAN more in detail because the EDEC can be integrated with SPAN to optimize the sensor network scheduling.

There are other location-based routing protocols presented in [42][43].

Chapter 3

The Proposed Event-Driven Enhanced Coverage Algorithm

Wireless sensors usually run on the battery. The limitation of battery of each sensor is hence the biggest constraint of a sensor network. Applications always hope the network can provide service as long as possible, while individual sensor tends to drain out its energy quickly and then stops working. Straightforwardly, there are two ways to solve this problem. One is to improve energy efficiency of sensor running, from data gathering process to data transmission to BS. In the previous chapter, we introduced many works that contribute to this. However, the requirement of network lifetime is often much more than what optimized energy efficiency can achieve. Therefore, there comes another way which is to increase the number of sensors of a network so that the total amount of energy of the network can be increased.

However, deploying a lot of redundant nodes will compromise the energy efficiency because the redundant nodes keep active all the time as the other nodes. And they will drain out their energy as fast as other nodes. Therefore, a scheduling algorithm is needed to arrange redundant nodes to be in power saving mode and wake them up only when it is necessary.

The other reason why a scheduling algorithm is important to WSN is due to the deployment of sensors. Wireless sensor networks are often used to undertake monitoring tasks or object tracking in an interested field. As the sensing range of each sensor is limited, different parts of this field needs to be monitored by different sensors. If we use pre-determined way to deploy sensors such as manually, it is easy to assure that each piece of the area is monitored by at least one sensor node. However, this is not always the case. For example, in hostile environments such as seabed and battlefield, it maybe impossible to precisely deploy sensors. Thus randomly deploying sensors is necessary in many applications. Uniformly distribution of sensors is hard to achieve by random deployment which means some parts of the area may have a lot of redundant sensors where other parts of the area are vacant. This breaks another important metric of WSN: full coverage. Paper [17] proposed a good way to find out these vacant areas and post-deploy extra nodes to repair the broken coverage. These extra nodes and those original redundant nodes may compromise the energy efficiency, too. Thus, with this un-uniform deployment, a scheduler is needed to let the redundant nodes keep in power saving mode.

As we mentioned, sensing coverage is another important quality of WSN besides the network lifetime. To achieve the full coverage, sensing unit of each sensor needs to co-operate together to assure that each point of the interesting field is effectively monitored by at least one sensor. Thus, this bring another energy consumption source because sensing units could consume quite amount of energy too. As WSN is one category of ad hoc wireless network, a lot of papers discussed how to design the energy-efficient communicating protocols for ad hoc wireless network, and many papers discussed how to adapt these protocols specifically for WSN [23][24][32][36][37]. However, most of them are not dedicated to scheduling sensing units energy-efficiently such that existing communication protocols can be integrated flexibly. As introduced in Section 1.1, a sensor typically consists of three main parts: power unit which provides the power, communicating unit which sends and receives data, and sensing unit which senses the environment and generates data. Most researches are on the power unit and communicating unit, and a few papers focus on how to combine all of them together to design a specific protocol for all parts of a sensor, power unit, sensing unit, and communicating unit [19]. In this thesis though, the proposed algorithm explores the possibilities of designing a protocol specifically for sensing unit and power unit and can combine with other existing communicating protocols of WSN which optimize the communicating aspect of the network.

Therefore, a scheduling algorithm which is able to handle redundant nodes in terms of communicating units as well as sensing units is necessary. It should separate the scheduling into two independent sub-tasks, one is to schedule sensing units, and the other one is to schedule communication units. Thus, sensing units are scheduled to be in working mode or sleeping mode such that at any time, a subset of sensing units exists to be working without breaking the sensing coverage where the number of nodes in this subset is minimum. Furthermore, the network connectivity can be maintained by other existing power-aware protocols so that at any time point, the communication units of another subset of nodes keep in working status.

To meet these requirements, we propose a new algorithm EDEC (Event-Driven Enhanced Coverage). This algorithm is able to schedule redundant sensing units of sensors. It has the following features:

1. Because of the random distribution of sensors, or intended deployment of redundant sensors in expectation of longer lifetime, redundant sensors need to be well scheduled. The EDEC takes advantage of this redundancy and makes good scheduling so that nearly maximum number of sensing units in network keep their sensing units in

sleeping mode without compromise the full coverage of the original network.

2. Moreover, not like [19][21][26][44], which are dedicated to achieve 1-degree or k-degree coverage by only one sensing range, the EDEC separates the sensing range for monitoring from another type of sensing range for event observing. This can avoid wasting a great deal of energy on maintaining full coverage while no events happen at all. The sensing range used for monitoring can be much bigger than the sensing range used for observing. Thus, the number of nodes that are used to achieve full coverage could be much less. However, the working nodes in this network must be sparse. When events occur, the nodes around them could be too far to provide anticipated precise and accurate sensing data, although these nodes are able to tell that some interested events are happening. Therefore, some additional nodes which are in sleeping mode need to be waken up so that the event is within shorter sensing range.

3. The EDEC takes advantage of the BT channel technology to implement the listening and notifying functionality. As in the EDEC, sensors need to wake up nodes around an event to let this event be within at least one node's R_f range, a listening and notifying mechanism is necessary. The BT channel technology is very simple and cheap where the sleeping nodes do not need to periodically check if an event in their R_f range is happening. This is very beneficial because if we let sleeping nodes periodically check if it should change its status, how to choose this period becomes a problem. If this period is too long, the latency could be serious while if the period is too short, the energy efficiency would be compromised badly.

4. The EDEC separates the design of communication of WSN from the design of sensing of WSN. This makes the EDEC to be able to combine with a lot of existing protocols dedicated to improving communicating aspect of WSN, such as [4][9][24].

5. The EDEC can potentially increases coverage degree. Although the EDEC only provides low coverage degree because it uses coarse sensing range, the coverage degree

of partial network, or we called precise and accuracy of partial network, can be very high when those areas have occurring events. This is not like [26] which only satisfies 1-degree coverage, the EDEC can achieve much higher coverage degrees as if all sensors in the network are actively monitoring the field.

6. For some types of WSN applications such as object tracking system, the EDEC is able to potentially decrease latency. Because when the sensors are detecting an interested object, they will wake up their neighborhood with pre-defined range to increase the observing accuracy. This range can be defined big enough so that at every direction that object could be going to move, there are enough alert nodes available to "catch" it with no detecting latency.

3.1 Eligibility Rule

3.1.1 Calculation of Eligibility

A lot of papers have been proposed for scheduling nodes into sleeping mode or working mode under some conditions. For instance, paper [27] put the interesting field as grids. In each grid, only one sensor will be in working mode; paper [24] put the sensors into sleeping mode when they find that turning on their communication units cannot improve the network capacity. Both of these two papers are more focusing on reducing communicating redundancy. Paper [45] proposed a probing algorithm in terms of sensing coverage. A sensor periodically sends out a probing message locally. If it gets reply message, it will close its sensing unit because there is working node in neighborhood; otherwise it will turn itself on. However, this algorithm cannot assure full coverage.

In the EDEC, we use a simple perimeter coverage scheme as presented in [26]. The sensing area of a sensor is defined as a disk, centered at this sensor with a radius equal to the sensing range. Nodes within certain range of a sensor are regarded as its neighbors. Furthermore, neighbors of a node can be classified into two types, neighbors of sensing units and neighbors of communicating units, because communicating radius can be different from sensing radius. Therefore, the group of neighbors of communicating unit of a sensor can be different from its "sensing neighbors". Figure 3.1 illustrates the definition of "sensing unit neighbor". In this figure, each black dot represents a sensor, and the circle around it represents its sensing area. Nodes in (a) and (b) are not neighbors, even though nodes in (b) do have a small overlap (This can avoid complicated geometrical computation.). Nodes in (c) and (d) are neighbors. In the next paragraph, we will explain how to simplify the sensing coverage calculation. The EDEC only takes neighbors to calculate coverage.

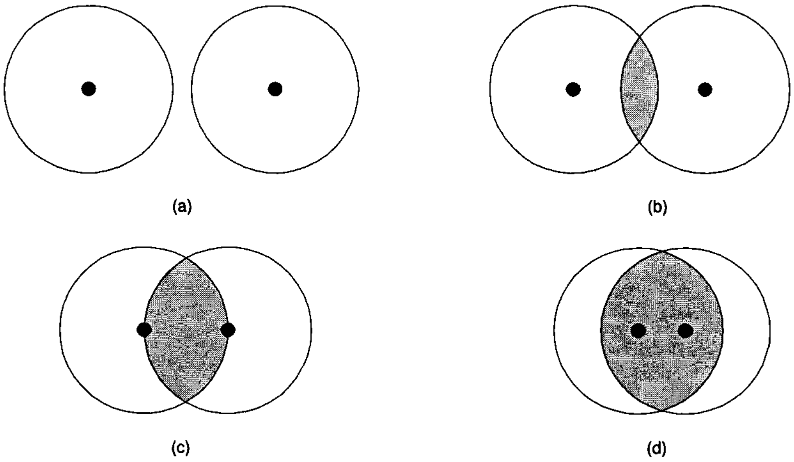
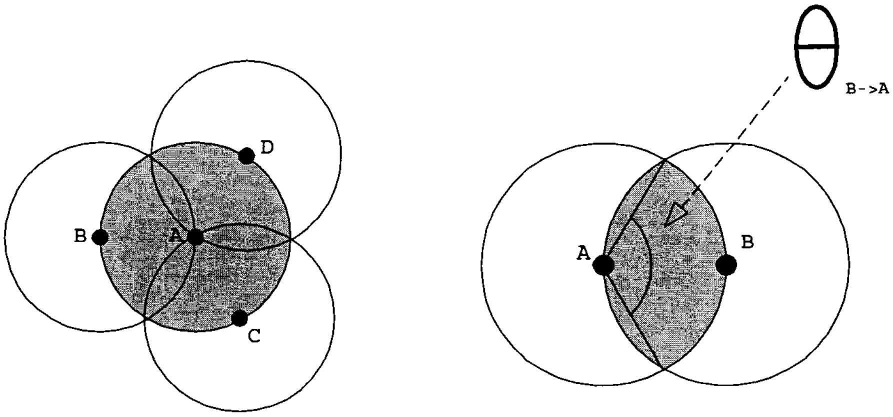


Figure 3.1: Sensing neighbors.

Figure 3.2 presents the sensing neighbors and the way to calculate their crescent-shaped overlaps. As shown in Figure 3.2 (a), $node_A$ is fully covered by its neighbors $node_B$, $node_C$, and $node_D$. In Figure 3.2 (b), we use angle $\theta_{B->A}$ to represent the shadow part of $node_A$ covered by its neighbor $node_B$. Although there is a small difference between shadow area and the angle, the use of θ can greatly simplify the calculation and it will not affect calculation accuracy and correctness. As shown in Figure 3.2 (a), as long as the sum of angles $\theta_{B->A}$, $\theta_{C->A}$, and $\theta_{D->A}$ equals to 360°, $node_A$ is regarded as fully covered. θ can be calculated by the following equation:

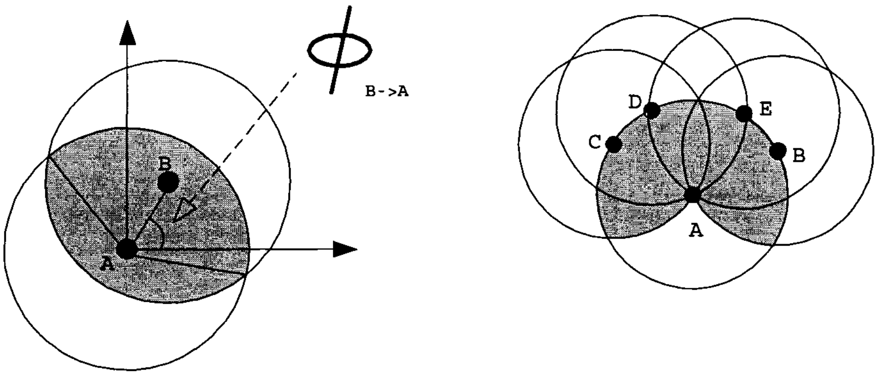
$$\theta_{B->A} = 2 \times arccos\left(\frac{d(A,B)}{2 \times r}\right) \quad 0 < d(A,B) \leq r \tag{3.1}$$

where d(A, B) is the distance between $node_A$ and $node_B$, r is the sensing radius. According to equation(3.1), $\theta_{B->A}$ is within range [120, 180).



(a) A is fully covered by B, C, and D.

(b) Angle is used to represent the covered area.



(c) Directional information is needed for coverage calculation.

(d) Overlap should be taken into account.

Figure 3.2: Eligibility calculation.

In Figure 3.2 (d), $node_A$ has sensing neighbors $node_B$, $node_C$, $node_D$, and $node_E$. The sum of the angles is greater than 360°. However, $node_A$ is not fully covered because of the overlaps among the shadow areas. Therefore, the direction information needs to be brought in to precisely remove the overlaps. In the 2-dimension graph, as long as nodes know their location information such as the values of x and y coordinates, the angle ϕ (shown in Figure 3.2 (c)) can be obtained to represent the direction of covered angle θ. Thus, all overlaps can be taken into account when calculating the coverage. The angle $\phi_{B->A}$ can be computed by the following equation:

$$\phi_{B->A} = arctg\left(\frac{y_B - y_A}{x_B - x_A}\right) \tag{3.2}$$

where x_A and y_A are the values of x and y coordinates of $node_A$, and x_B and y_B are the values of x and y coordinates of $node_B$.

The location information of a node is required for eligibility calculation. The EDEC assumes every node in the network knows its position represented by (x, y). In fact, this information plays a very important role in many WSN designs as [46] states "Sensing data without knowing the sensors location is meaningless.". It can be obtained by GPS (Global Positioning System), triangulation technology, or combination of both [14][15][27][44].

3.1.2 Eligibility Rule Operation

Eligibility rule is basically used to schedule the sensing units. It is only one subcomponent of the network hence needs to be integrated with other components. For example, it needs information provided by the communicating part of the network. One of the design goals of the EDEC is to avoid increasing the system complexity and take advantage of the existing WSN protocols. As we analyzed, location information and probably some other information such as energy can be obtained from communicating part of the network, because most of the communicating protocols of WSN exchange some control information either pro-actively or re-actively for infrastructure setup. In this way, the tasks of communicating and sensing are allocated to the proper network components and the interface for data between communicating part and sensing part can be created without too much update on original communication design.

On the other hand, when a sensor finds that it does not satisfy the eligibility rule, it enters into working mode; however, when a sensor finds that the eligibility rule is satisfied, it cannot decide to close sensing unit right away because of the "blind point" issue. Figure 3.3 presents such a scenario. $node_A$ is fully covered by $node_B$, $node_C$, $node_D$; and $node_C$ is fully covered by $node_A$, $node_E$, $node_F$. If both $node_A$ and $node_C$ enter into sleeping mode, the blind point embraced by white line will appear, the full coverage is then broken.

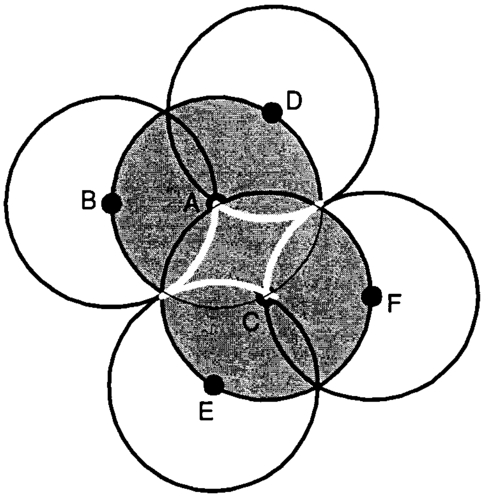


Figure 3.3: Blind point.

Papers [19][24][26][47] addressed such a problem by broadcasting and setting up back-off mechanism, when nodes compete to enter into sleeping mode. But this will increase the complexity of data interface between communicating unit and sensing unit and add overhead of network running. The EDEC, however, can adaptively address this issue by two different ways depending on the combined communication protocols:

1. Cluster-based communicating protocols, such as LEACH, usually have globally-broadcasting function for network setup. Therefore, what the EDEC only needs to do is to add the EDEC specific information in those broadcasting messages, every node

then will be able to keep a table with enough information of all other nodes, such as their location information. Then, the following procedure will be taken:

- Sort nodes by energy level, node ID, or combination of both;
- Apply eligibility rule to each node in that sorted list in a descending order;
- Each node will calculate out if it is eligible to shut down its sensing unit without any further exploring.

Sorting the nodes by energy and node ID is the way to increasing the load uniform distribution so that nodes with more resources should work longer with higher probability. And it also solves the "blind point" problem without further ad hoc broadcasting.

After this procedure, many sensing units will enter into sleeping mode while a few nodes keep their sensing units on to fully cover the field.

2. For communicating protocols without globally broadcasting, the EDEC requires two-hop neighbor's information. Figure 3.4 illustrates the concept of two-hop neighbor.

As shown in Figure 3.4, $node_B$ is the "sensing neighbor" of $node_A$ because it is in sensing range of $node_A$. Also, $node_C$ is the sensing neighbor of $node_B$, and it is not the sensing neighbor of $node_A$. Then, we call $node_C$ as the two-hop neighbor of $node_A$.

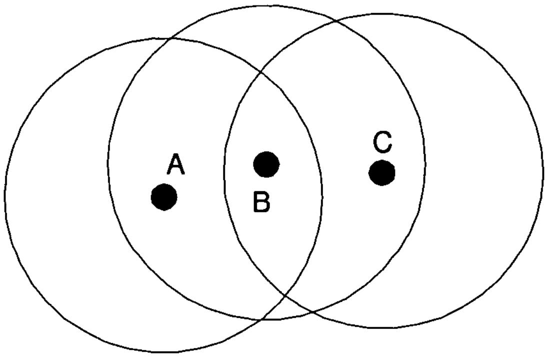


Figure 3.4: Two-hop sensing neighbor.

After nodes know locations of their neighbors and two-hop neighbors, they can make eligibility decision for most of the cases. In Figure 3.5 (a), $node_A$ is covered by $node_B$, $node_C$, $node_D$. But it cannot decide if it can enter into sleeping mode right away. Instead, it needs to obtain its two-hop neighbors' information, which are information of $node_E$ and $node_F$. Then, $node_A$ knows that its neighbors $node_B$, $node_C$, $node_D$ will not close their sensing units for sure. (We call the neighbors of $node_A$, such as $node_B$, $node_C$, $node_D$, "unclosable neighbor" because they will not close their sensing units in any conditions due to the full coverage requirement.) $node_A$ then will safely enter into sleeping mode. However, in Figure 3.5 (b), $node_A$ finds that among its neighbors that can fully cover it, there is one neighbor $node_D$ that is also covered by $node_A$, $node_E$, $node_F$. (We call the neighbors of $node_A$, such as $node_D$, "closable neighbor" because it could close its sensing unit under some conditions.) In this situation, $node_A$ will have to make decision based on the node ID and energy. For instance, if the current energy of $node_A$ is less than $node_D$, $node_A$ will choose to close its sensing unit according to the load balance principle; otherwise, it will keep its sensing unit on. In the case that both nodes have equal amount of energy, node ID plays the decisive role such that $node_A$ closes its sensing unit if its node ID is bigger than $node_D$; otherwise, it keeps on. (There could be some further solution for this situation by adopting simple broadcasting of the decision of becoming working node or sleeping node. This will increase the number of sleeping node by trading off with overhead of broadcasting. We will leave this to the future work.)

3.2 R_c and R_f of Full Coverage

We believe that providing full coverage is a task that is different from the task of observing occurring events with enough precision and accuracy. An application requires

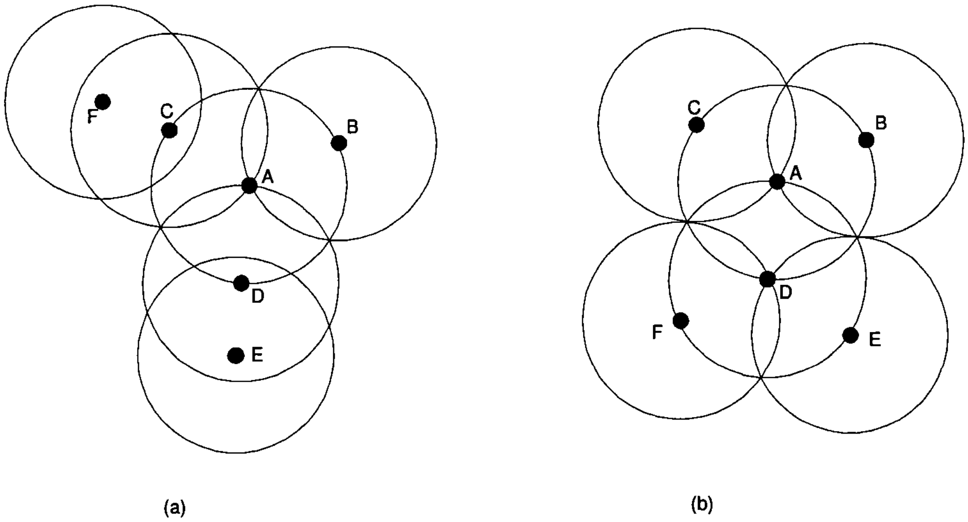


Figure 3.5: Two-hop neighbor coverage.

full coverage such that the observer should be able to be alerted whenever an event is happening at any point within the monitoring field. This does not necessarily requires the same accuracy and precision as the observer needs to provide when observing the event. However, most of the recent papers in WSNs regard both as the same task that the interested field has to be under full and accurate monitoring. In fact, this may waste a lot of resources to keep precise devices on when the happening frequency of interested events is very low. Recently, [20] proposed a set of new definitions of sensing range:

R1 is referred as radius of complete influence. "For a particular sensor, it is defined as the distance from the sensor such that all targets originating within this radius are surely detected."

R2 is referred as radius of no influence. "For a particular sensor, it is defined as the distance from the sensor such that the sensor cannot detect any target originating beyond this radius."

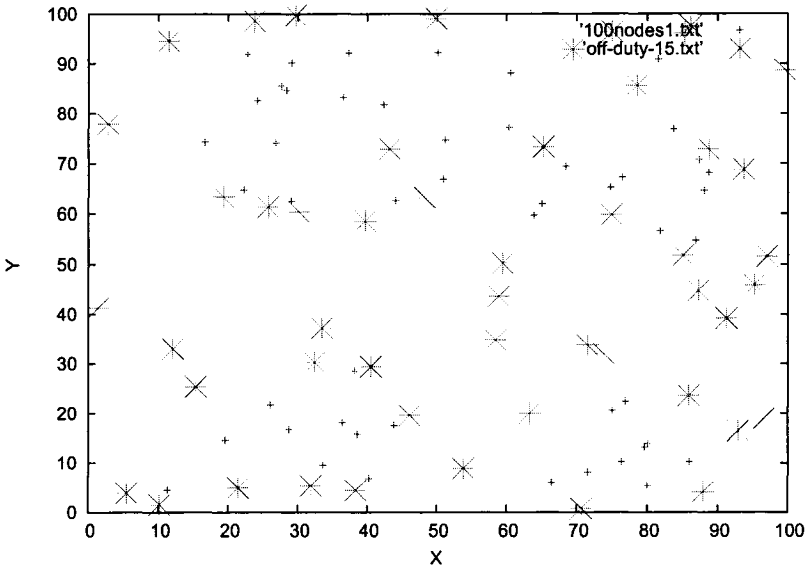
Originated from separation of R1 and R2, EDEC develops new definitions of sensing range:

R_c — Coarse sensing range. It is defined as the distance from the sensor such that all interesting events occurring within this radius can be surely detected.

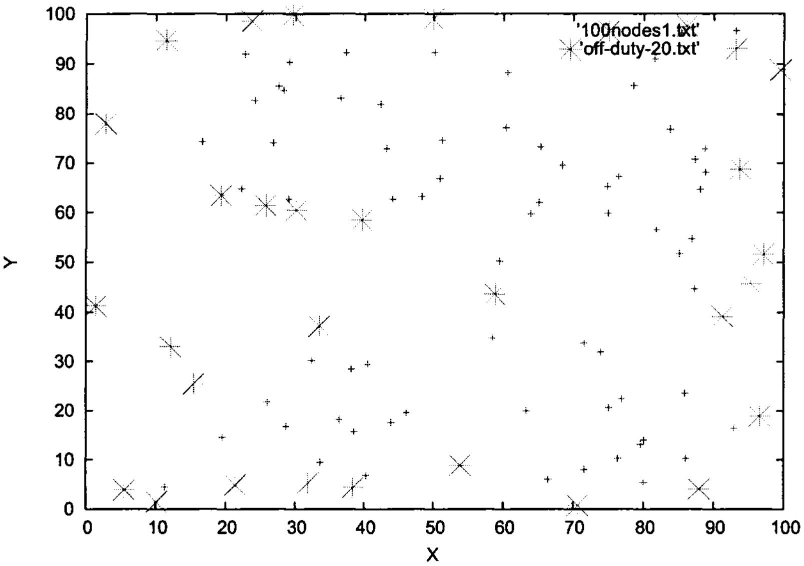
R_f — Fine sensing range. It is defined as the distance from the sensor such that all interesting events occurring within this radius cannot only be surely detected, but also be observed with enough accuracy and precision.

Apparently, R_c and R_f are highly application-dependent. They could have the same values in some applications where in many others, they can be much different. In this thesis, our target is to explore how this design can bring the benefit for WSN. In terms of full coverage, the number of nodes working for coverage using R_c could be much less than using R_f when R_c is bigger than R_f. Therefore, instead of using R_f, we tried to use bigger values of sensing radius for full coverage task.

In Figure 3.6, sign "*" represents the working node and sign "+" represents the sleeping node. As shown in Figure 3.6 (a) and (b), when the value of sensing radius increases, the working nodes are more sparse, which satisfies the EDEC's goal of maintaining a minimum number of working sensing units to implement the full coverage.



(a) R_c=15.



(b) R_c=20.

Figure 3.6: Different values of R_c in 100 nodes distribution.

In Section 4.3, we will present how different values of R_c and R_f will affect the system performance.

3.3 BT Module for Sensing Units

As stated in previous section, we use R_c to provide full coverage to monitor the interesting field. However, when some events happen, those sensing units in working mode may not be able to provide enough accuracy or precision, although they can detect the occurring of the event. Figure 3.7 presents such an example. In this figure, $node_A$ is in working mode while other nodes are in sleeping mode. When an event is happening, because it is in R_c range of $node_A$, $node_A$ can detect it. However, because it is on the edge of the R_f of $node_A$, $node_A$ may not be able to provide enough precision. Thus, $node_A$ needs to wake up its neighbors so that $node_B$ will be waken up to join the observing group. As the event is in the R_f range of $node_B$, enough precise information can be obtained as application needed.

Therefore, a listening-notifying mechanism is required for the EDEC to enhance the coverage degree when events happen. This is a communicating work that could be put in communicating protocols. However, it will increase the complexity of combining the EDEC with existing communicating protocols. Because a notifying mechanism not only requires simple data, but also potentially changes the original working flow of the existing communicating protocols. For example, a sensing unit detects an event, and requests its communicating unit to broadcast an alert to the neighbors so that the neighbors can wake up their sleeping sensing units. This requires the communicating unit of the monitoring sensor and all its neighbors to be in working mode; otherwise, the necessary neighboring sensing units could not be waken up in time. This conflicts with many power-saving communicating protocols because they are tending to schedule

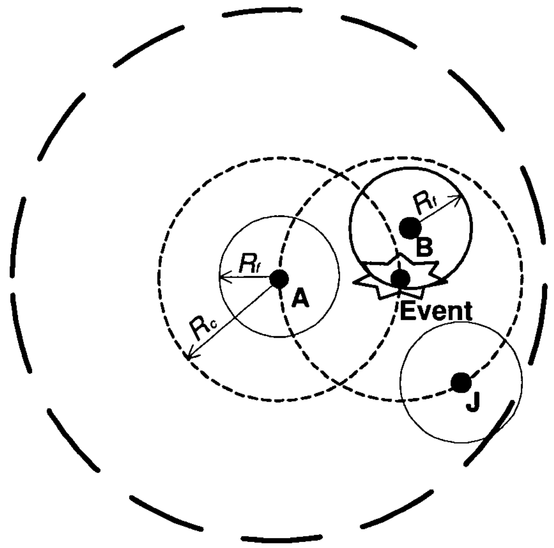


Figure 3.7: Listening and Notifying.

the radios in sleeping mode too for most of the time. Therefore, requiring the radios of neighbors to turn on for listening possibly conflicts with the communication schedules.

In addition, as stated in paper [48], existing communicating units are designed for data transmission and reception that require good channel quality, high speed, and thus complex and power consuming hardware. The task of listening and notifying, however, is only to get a binary information about whether a packet targeted at this node is coming. Therefore, the EDEC designed a separate communicating module specifically for listening and notifying. This module uses BT (Busy Tone) technology. BT technology is used to broadcast on the channel for specified duration. The advantages of using BT technology are as follows:

- No information is encoded in the BT.
- Radio system only has to detect energy on channel rather than decode packet.
- The hardware is simple.

- The detection time is short.
- There is no need to handle collisions.

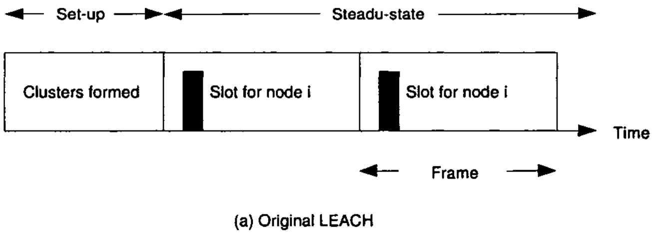
Plus, BT consumes much less power compared to data transmission module. Preliminary research showed that the wake-up radio may only take around 1μW, comparing to 10mW of a CDMA radio in monitoring mode [46][48][49].

3.4 Combining with Communication Protocols

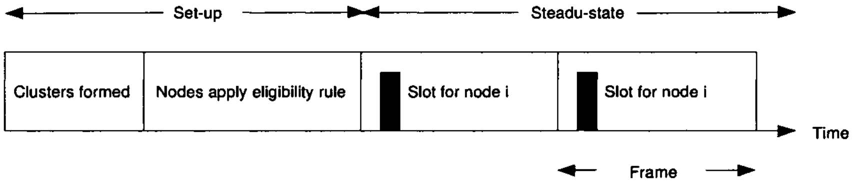
3.4.1 Combining with Hierarchical Protocols

The LEACH, Low-Energy Adaptive Clustering Hierarchy, is the most famous protocol of representation for hierarchical WSNs [4]. In this section, we will theoretically present how to combine the EDEC with the LEACH.

The LEACH is set to be globally round-based. This is popular for most of the hierarchical protocols, because they need an initial phase to set up the clusters throughout the network. In the network initialization phase, every node in the network will decide if it should be the cluster head. According to the design, only a few nodes will be chosen to be cluster heads. The rest of the nodes select one of the claimed cluster heads to be its cluster head and join that cluster. Ideally, nodes always choose to join the cluster where they are geographically close to each other. Nodes then broadcast this join-message globally so that cluster head nodes know what are their cluster members. Cluster heads then create time slots for each of their member nodes so that members can be in sleeping mode most of the time. Only at its own time slot, it will wake up and send data to cluster head. This TDMA schedules are broadcasted by cluster head nodes. Then, the network enters into steady phase which is much longer than setup phase. Cluster member nodes can only communicate with their cluster head nodes, and cluster head nodes are responsible for communicating with base station because this is typically long-distance communication, and it should be avoided as much as possible. Figure 3.8 (a) shows the time-line of LEACH operation, Figure 3.8 (b) shows the time-line after combining the EDEC with the LEACH. In fact, the only difference is in the set-up phase where every node adds a new process: eligibility calculation. There is no extra time-line except set-up and steady-state, and there is no extra time-line within set-up and steady-state. This makes the combination very smooth.



(a) Original LEACH



(b) Combination of EDEC&LEACH

Figure 3.8: Timeline.

When events happen, some monitoring sensing units will detect and wake up their neighbors so that more sensing units are turning on to observe the events. After events end, sensing units will return back to their original sensing status. This process is completely within the EDEC algorithm without involving LEACH at all, because the EDEC has a separate BT module to do notifying-listening work. This is why LEACH can keep almost unchanged on its own time-line. On the other hand, in original LEACH, every node will send data to its cluster head at its time-slot. However after employing the EDEC, nodes will send data only when they have effective data for detected events.

This is another benefit by decreasing a number of radio communications.

3.4.2 Combining with Flat Protocols

In this section, we present the theoretical analysis and design of combining the EDEC with the SPAN. The SPAN (A power saving technique for multi-hop ad hoc wireless networks) is proposed in paper [24]. It is a flat protocol focusing on data communicating.

The SPAN has some special features. First, "it adaptively elects coordinators from all nodes in the network. SPAN coordinators stay awake continuously and perform multi-hop packet routing within the ad hoc network, while other nodes remain in power-saving mode and periodically check if they should wake up and become a coordinator." The time-line of this procedure likes the EDEC's eligibility rule, but it is for communication units. Second, "SPAN elects coordinators using only local information in a decentralized manner — each node only consults state stored in local routing tables during the election process." Thus, each sensor in SPAN is independent, and distributed. It independently makes decisions such as when to broadcast, when to collect neighbor information, and when to enter sleeping mode. Accordingly, the EDEC cannot reply on global round-based design. Instead, each sensing unit in the EDEC should collect sensing information on its own, and make its own decision such as closing sensing unit or not. The following is the original operation of SPAN:

1. A node periodically broadcasts HELLO messages that contain the nodes status (for example, whether or not the node is a coordinator), its current coordinators, and its current neighbors. From these HELLO messages, each node can construct a list of its neighbors and coordinators, and for each neighbor, a list of its neighbors and coordinators.

2. Periodically, a non-coordinator node determines if it should become a coordinator

or not according to SPAN coordinator election rule.

3. Each coordinator periodically checks if it should withdraw as a coordinator according to SPAN coordinator withdrawal rule. This happens after a node has been a coordinator for some period of time.

To adapt the EDEC into the SPAN, the operations are modified as following:

1. In the SPAN's first phase when nodes broadcast HELLO messages, the EDEC will add some extra information in HELLO messages. This phase is actually an information-collecting phase, therefore the necessary information for the EDEC should be also arranged to be obtained in this phase.

2. In the SPAN's second phase when sensors periodically determine if they should become coordinators, the EDEC eligibility rule should be applied to make decision whether it needs to close its sensing unit or not. At this point, the sensors have necessary information for the SPAN and for the EDEC. The periodical determination can be regarded as "individual round" which helps nodes rotate status and distribute evenly both tasks: communicating and sensing.

3. A coordinator checks if it should withdraw, and also its sensing unit checks if it should turn off or turn on itself for the sensing task.

Table 3.1 presents the updated HELLO message in which node position is added for the EDEC. And another piece of required information can be either node ID or node available energy, or both. They are useful when the EDEC prioritizes the neighbors.

Figure 3.9 shows the updated time-line of the SPAN-EDEC. (Note that both communication unit and sensing unit are turned on in the initialization phase.) When there is an event occurring, sensing-active nodes will wake up all neighbor sensing units. This will not affect the SPAN, because the EDEC use a separated BT-channel to do this listening-notifying job. After the event ends, involved sensing units will return back to their original sensing status.

Table 3.1: Hello message for SPAN and EDEC.

Hello message
SPAN specific information
Is coordinator
Is tentative
Coordinator list
Neighbor list
EDEC specific information
Node ID
Node position
Node energy

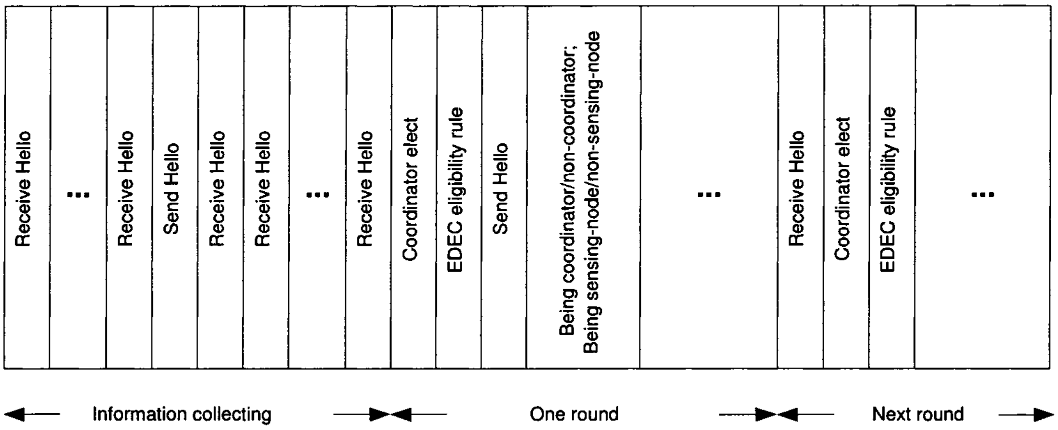


Figure 3.9: Timeline of the updated SPAN.

Figure 3.10 shows how the EDEC operates when combining with the SPAN.

3.4.3 Summary

From the discussion of Section 3.4.1 and Section 3.4.2, we can find that the interface between the EDEC and the communication protocols are fairly simple. The exchanged data crossing the interface are location information, remaining energy, probably plus node ID. To combine with the EDEC, the communication protocols do not need to change their process no matter they are cluster-based protocols or multi-hop protocols,

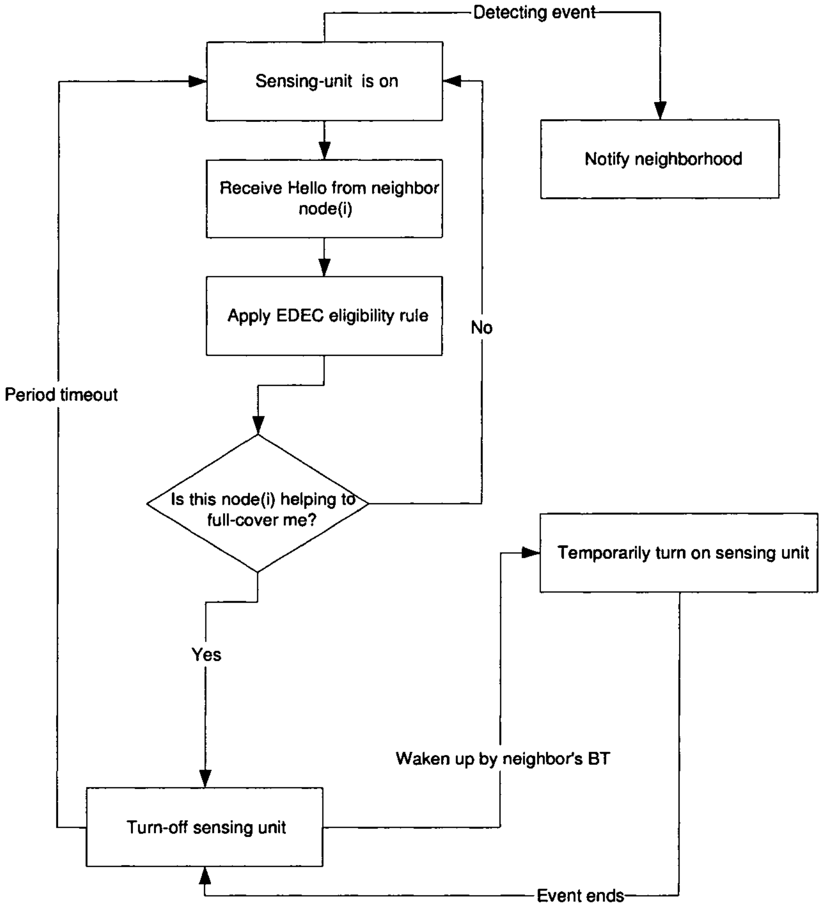


Figure 3.10: Flow graph of EDEC with SPAN.

globally organized or self-setup. This makes the EDEC flexible enough to adapt to different existing communication protocols for extensive sensor network applications.

Chapter 4

Experimental Results and Evaluations

This chapter presents the experiments design and results analysis of the EDEC algorithm. In simulation, the EDEC algorithm is adapted into LEACH (Low-Energy Adaptive Clustering Hierarchy) [4], and the simulation results are compared with original LEACH in terms of network lifetime, amount of effective data delivered, and energy dissipation. Although the proposed algorithm can be combined with any other communication protocols, we select LEACH because its NS-2 simulation code is available in the Internet and it is one of the most famous communication protocols suitable for wireless sensor networks [26].

The simulation is performed using Network Simulator 2 [50].

4.1 Simulation Models

This section describes the models that were used for channel propagation, communication energy dissipation, and computation energy dissipation.

4.1.1 Channel Propagation Model

In a wireless channel, the electromagnetic wave propagation can be modeled as falling off as a power law function of the distance between the transmitter and receiver. In addition, if there is no direct line-of-sight path between the transmitter and the receiver, the electromagnetic wave will bounce off objects in the environment and arrive at the receiver from different paths at different times. This causes multi-path fading. In [4], both the free space model and the multi-path fading model were used, depending on the distance between the transmitter and receiver as defined by the channel propagation model in NS-2. In this paper, we use the same channel propagation model as [4] for both communication radio and BT radio for the sake of comparability.

If the distance between the transmitter and receiver is less than a certain cross-over distance ($d_{crossover}$), the Friss free space model is used (d^2 attenuation), and if the distance is greater than $d_{crossover}$, the two-ray ground propagation model is used (d^4 attenuation). The cross-over point is defined as follows [4]:

$$d_{crossover} = \frac{4\pi\sqrt{L}h_r h_t}{\lambda} \tag{4.1}$$

where

$L \geq 1$ is the system loss factor not related to propagation,

h_r is the height of the receiving antenna above ground,

h_t is the height of the transmitting antenna above ground, and

λ is the wavelength of the carrier signal.

If the distance is less than $d_{crossover}$, the transmit power is attenuated according to the Friss free space equation as follows [4]:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2 L} \tag{4.2}$$

where

$P_r(d)$ is the receive power given a transmitter-receiver separation of d,

P_t is the transmit power,

G_t is the gain of the transmitting antenna,

G_r is the gain of the receiving antenna,

λ is the wavelength of the carrier signal,

d is the distance between the transmitter and the receiver, and

$L \geq 1$ is the system loss factor not related to propagation.

This equation models the attenuation when the transmitter and receiver have direct, line-of-sight communication, which will only occur if the transmitter and receiver are close to each other. If the distance is greater than $d_{crossover}$, the transmit power is attenuated according to the two-ray ground propagation equation as follows [4]:

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4} \tag{4.3}$$

where

$P_r(d)$ is the receive power given a transmitter-receiver separation of d,

P_t is the transmit power,

G_t is the gain of the transmitting antenna,

G_r is the gain of the receiving antenna,

h_r is the height of the receiving antenna above ground,

h_t is the height of the transmitting antenna above ground, and

d is the distance between the transmitter and the receiver.

In this case, the received signal comes from both the direct path and a ground-reflection path.

In the experiments described in this thesis, an omni-directional antenna is used with the following parameters [4][50]: $G_t = G_r = 1$, $h_t = h_r = 1.5m$, no system loss ($L = 1$), 914 MHz radios, and $\lambda = \frac{3\times10^8}{914\times10^6} = 0.328m$. Using these values, we get $d_{crossover} = 86.2m$.

4.1.2 Radio Energy Model

Different assumptions about the radio characteristics, including energy dissipation in the transmit and receive modes, will change the advantages of different protocols. Therefore, we choose the same radio energy model as [4] for comparability reason. To transmit a l-bit message over a distance d, the radio expends:

$$E_{Tx}(l,d) = E_{Tx-elec}(l) + E_{Tx-amp}(l,d) \tag{4.4}$$

$$E_{Tx}(l,d) = \begin{cases} lE_{elec} + l\varepsilon_{friss-amp}d^2 & d < d_{crossover} \\ lE_{elec} + l\varepsilon_{two-ray-amp}d^4 & d \geq d_{crossover} \end{cases} \tag{4.5}$$

and to receive this message, the radio expends [4]:

$$E_{Rx}(l) = E_{Rx-elec}(l) = lE_{elec} \tag{4.6}$$

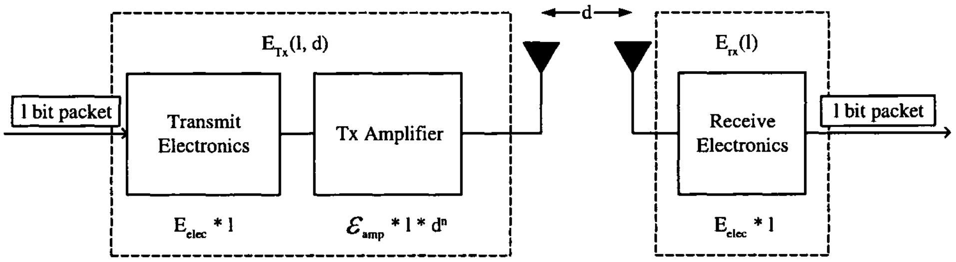


Figure 4.1: Radio energy dissipation model.

As shown in Figure 4.1, the electronics energy E_{elec} depends on factors such as the digital coding, modulation, and filtering of the signal before it is sent to the transmit amplifier. In addition, when using DS-SS, the electronics energy accounts for the spreading of the data when transmitting and the correlation of the data with the spreading code when receiving. Researchers have designed transceiver baseband chips that support multi-user spread-spectrum communication and operate at 165 mW in transmit mode and 46.5 mW in receive mode [4]. For the experiments described in this paper, we set the energy dissipated per bit in the transceiver electronics to be

$$E_{elec} = 50nJ/bit \tag{4.7}$$

The parameters $\varepsilon_{friss-amp}$ and $\varepsilon_{two-ray-amp}$ will depend on the required receiver sensitivity and the receiver noise figure. We take the deduced results from [4] and will use these radio energy parameters for the simulations described in this chapter:

$$\varepsilon_{friss-amp} = 10pJ/bit/m^2 \tag{4.8}$$

$$\varepsilon_{two-ray-amp} = 0.0013pJ/bit/m^4 \tag{4.9}$$

All the associated parameters are summarized in Table 4.1:

Table 4.1: Radio characteristics and parameter values.

Description	**Parameter**	**Value**
Cross-over distance for Friss and two-ray ground attenuation models	$d_{crossover}$	$\frac{4\pi h_r h_t}{\lambda}$
Transmit power	P_t	$\varepsilon_{friss-amp} R_b d^2 \quad d < d_{crossover}$ $\varepsilon_{two-ray-amp} R_b d^4 \quad d \geq d_{crossover}$
Receive power	P_r	$\frac{\varepsilon_{friss-amp} R_b G_t G_r \lambda^2}{(4\pi)^2} \quad d < d_{crossover}$ $\varepsilon_{two-ray-amp} R_b G_t G_r h_t^2 h_r^2 \quad d \geq d_{crossover}$
Minimum receiver power needed for successful reception	$P_{r-thresh}$	6.3nW
Radio amplifier energy	$\varepsilon_{friss-amp}$ $\varepsilon_{two-ray-amp}$	$\frac{P_{r-thresh}(4\pi)^2}{R_b G_t G_r \lambda^2}$ $\frac{P_{r-thresh}}{R_b G_t G_r h_t^2 h_r^2}$
Radio electronics energy	E_{elec}	50nJ/bit
Compute energy	E_{BF}	5nJ/bit
Bitrate	R_b	1Mbps
Antenna gain factor	G_t, G_r	1
Antenna height above the ground	h_t, h_r	1.5m
Signal wavelength	λ	0.325m
Cross-over distance for Friss and two-ray ground attenuation models	$d_{crossover}$	87m
Radio amplifier energy	$\varepsilon_{friss-amp}$ $\varepsilon_{two-ray-amp}$	$10pJ/bit/m^2$ $0.0013pJ/bit/m^4$

4.1.3 NS-2 Implementation

To implement the EDEC, we added several features to NS-2, an event-driven network simulator with extensive support for simulation of wireless network protocols. The extensions include supporting multiple wireless physical network interfaces, BT communication energy dissipation, and the protocol architecture discussed in this thesis. Program paradigms for these implementations can be found in Appendix A.

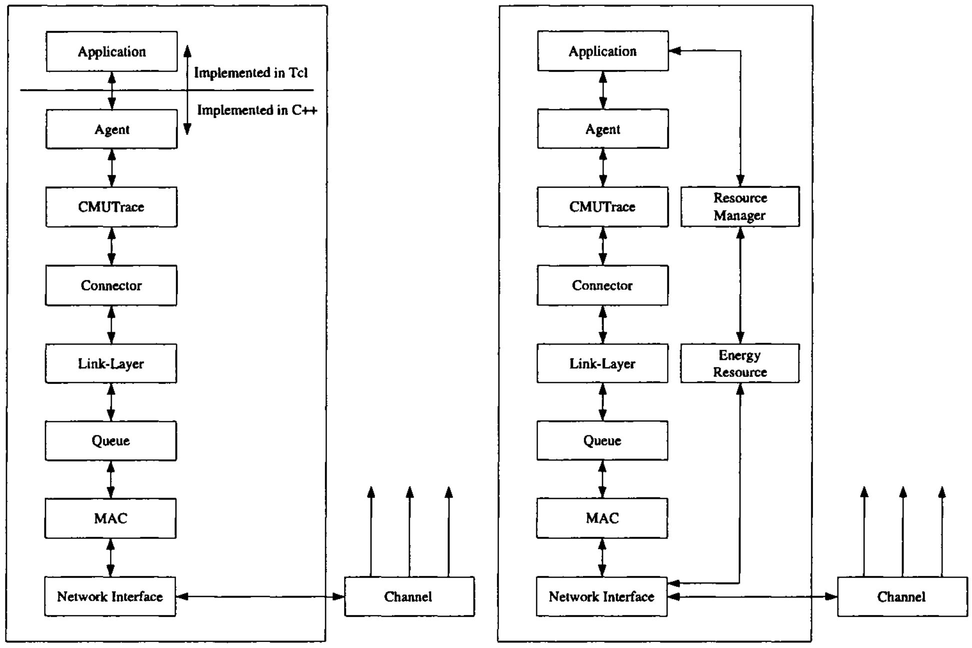
One of the key updates for the EDEC node is to have a separated BT radio model which is dedicated to waking up neighborhood when some events happen. In Figure 4.2, (a) represents original NS-2 node, (b) represents LEACH node which has some updates such as adding resource manager, (c) represents EDEC node which has separated set of networking components such as MAC layer, Physical Layer, and Channel layer. The signal coming from BT radio of a sensor node will trigger waking up its sensing unit as well as making its communication unit to have effective data to send when turning to its time slot.

4.2 Set-up of Experiments

In our experiments, the random 100-node network, shown in Figure 4.3 was used in a $100meter \times 100meter$ plane. Each black dot represents a sensor (We assume that all sensors are immobile during their lifetime after deployment.). The base station was placed at location (x=50, y=175) (not shown in the figure). For some experiments such as choosing R_c and R_f, up to 50 different random distributions were used to get an average result.

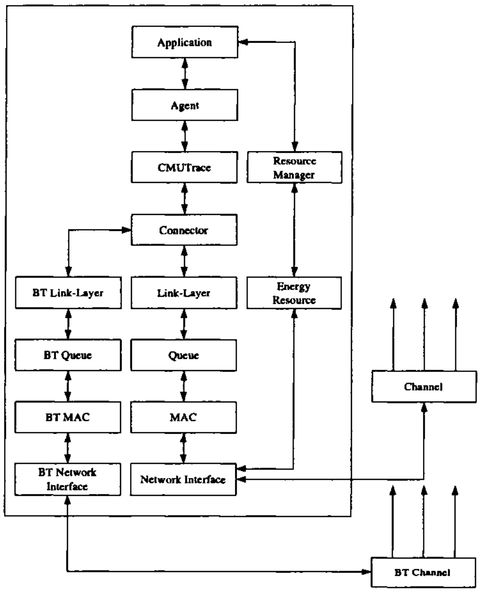
Table 4.2 lists the network parameters. For those parameters introduced from paper [4], we use the same values. For parameters from the EDEC, we will explain them in later sections.

As simulation starts, the first step is the cluster formation process which is the setup phase of the LEACH. According to the LEACH, each node will be either competing to be cluster head or choosing the best cluster to join by broadcasting ADV message at least once. In this ADV message, we added information needed by the EDEC such as location and available energy. Therefore, the eligibility is calculated such that every node knows whether or not it should turn off its sensing unit when this phase is finishing.



(a) NS-2 mobilenode.

(b) LEACH mobilenode.



(c) EDEC mobilenode.

Figure 4.2: Comparing different mobilenodes' structures.

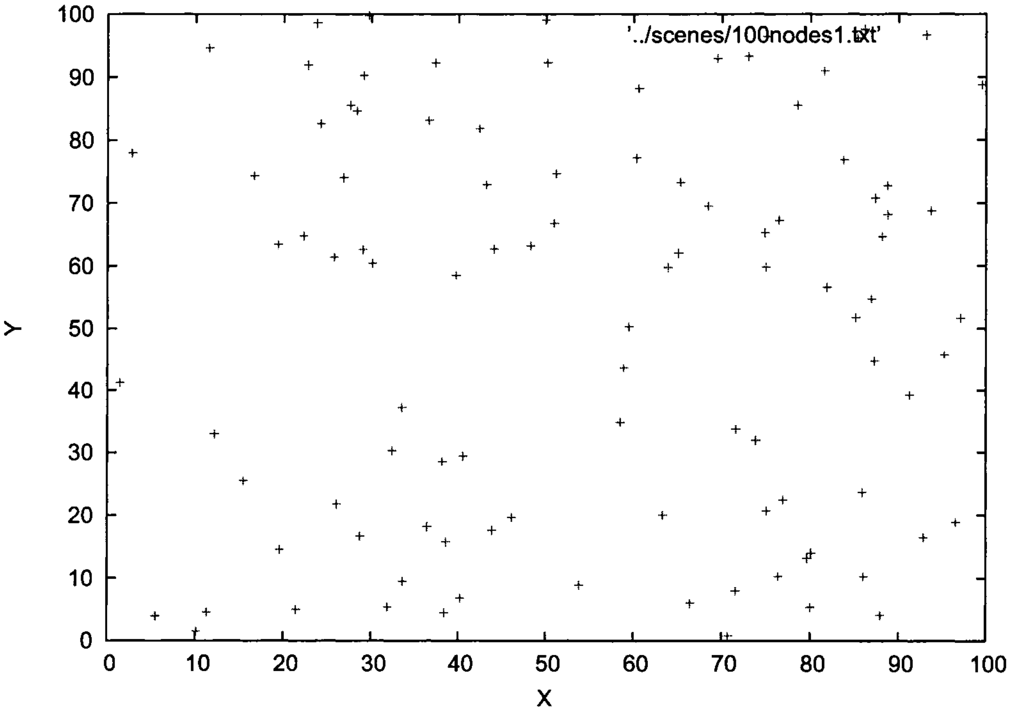


Figure 4.3: 100 nodes distribution.

Next phase is the steady phase while we designed a scheduler to "create" events at some nodes chosen randomly. Those nodes will broadcast using BT channel to wake up their sensing neighbors. Nodes, no matter their sensing units are in sleeping mode or not will set effective data for their communication units and remove the energy for running sensing units. The energy are also removed for BT notifying and receiving. The effective data are sent out when it turns to the node's TDMA time slot. This steady phase is going on until the LEACH decides to rotate the clusters. The whole simulation will end at 3600 seconds, or if less than 5 nodes are alive. On average, one experiment runs 25 minutes or a bit more as each node is given 2J initial energy. In next sections, we will present the specific experiments done for adjusting a few important parameters separately.

Table 4.2: Parameters of test network.

Parameter	Description	Value
R_c	Radius of coarse sensing coverage	20 meter
R_f	Radius of fine sensing coverage	10 meter
$E_{sensing-unit}$	Energy to run sensing unit per second	$0.001 J/second$
F_{event}	Frequency of event happening	50% (100% means events happen all the time.)
$E_{lasting}$	How many seconds that sensing unit takes to detect an event	1 second
Nodes	Number of nodes	100
Network size	A square	$100m \times 100m$
Base station location		(50, 175)
Radio propagation speed		$3 \times 10^8 m/s$
Processing delay		$50\mu s$
Radio speed		1Mbps
Data size	The data size in every packet sent by nodes	500 bytes

4.3 Choosing R_c and R_f

One of the key features of the EDEC algorithm is to schedule nodes' sensing units to be sleeping or working while the original sensing coverage is maintained by R_c instead of R_f. However, R_c and R_f are highly dependent on the physical characteristics of sensing units and the application scenarios. That is, R_f may not be able to be defined by us, instead, it is defined by the physical device. Therefore, instead of working out the specific value of R_f, we will focus on revealing how different values of R_c can affect the whole network's lifetime and energy efficiency, i.e. when R_f is fixed, how to adjust R_c so that the best result can be obtained. (In this simulation, we assume that all sensors are identical in terms of physical properties such as sensing range.)

In our simulation, we designed and implemented the EDEC eligibility rule to be able to take different values of R_c. Table 4.3 presents the samples we used for exploring the value of R_c by applying to different topologies (50 different random distributions). Note that the main goal of R_c is to make the number of active nodes as small as possible while not compromising the original sensing coverage.

Table 4.3: R_c and eligibility rule.

R_c	**Number of sleeping nodes**	R_c	**Number of sleeping nodes**
10	11	15	44
20	69	25	76
30	78	35	79
40	80	45	81
50	81	55	82
60	82	65	82
70	82	75	83

As we can see, by applying the EDEC eligibility rule, 11 nodes can be turned off when R_c equals to 10 meters. Also, by varying the R_c's value from 10 meters to 75 meters, the number of sleeping nodes increases which is in our expectation.

We also tested with other different parameters such as node density. Figure 4.4 shows that increasing R_c and increasing node density will increase the number of sleeping nodes which is consistent with our design.

However, active node number does not remain constant over different deployed node density when the sensing range are fixed. Instead, it increases as the the deployed node number increases as illustrated in Figure 4.5. This is actually due to the increasing of edge nodes (located at the boundary of the deployed area). Edge nodes do not have chance to be turned off because all their neighbors are located on one side of them. Intuitively, increasing edge nodes will increase the active node number, however, experimental result shows that the EDEC eligibility rule still effectively limits the number of active nodes. According to the figure, when the deployed node number is increased from 100 to 300, the number of active nodes only increases about 30%.

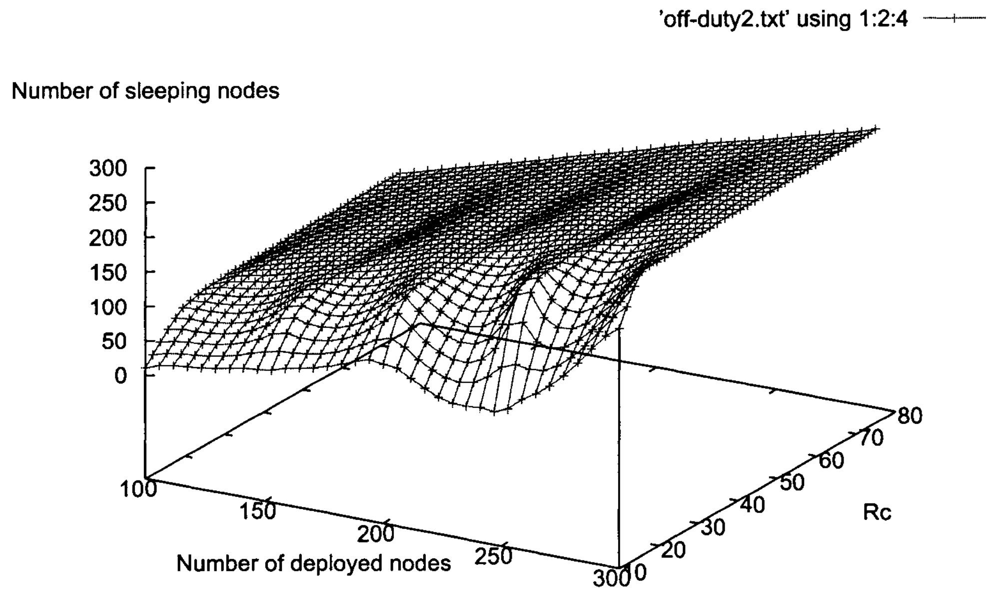


Figure 4.4: R_c with node density and eligibility rule.

4.4 Energy of Running Sensing Device

The very core ideal of the EDEC algorithm is to shut down sensing units of some nodes to save energy because they are redundant in terms of sensing coverage. Therefore, the energy used to run sensing device is a very important parameter of the network. For instance, if it is a very small amount compared to the energy running radio circuits, even negligible, using the EDEC to shut down them may not be applicable because a scheduling algorithm itself consumes energy on both processing and BT broadcasting. However, what percentage of the energy a sensing unit takes in a sensor really depends on specific application or types of sensors. Therefore, our task is to find out, when the energy of sensing unit varies, how performance of the EDEC is going to be.

The LEACH put energy of sensing unit to be zero, we modified this to different

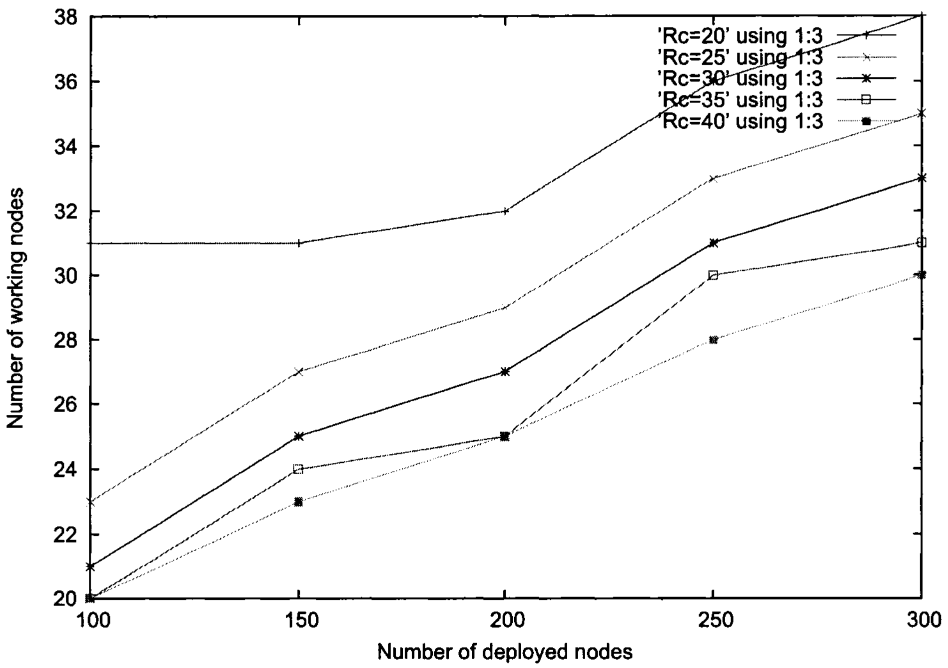


Figure 4.5: Edge nodes.

values when they are used in the EDEC. Table 4.4 presents the sample values we used for both LEACH and EDEC. (Note, we assume the event frequency for the EDEC is 0.25 at here; in next section, we will introduce how event frequency parameter will affect the performance of the EDEC.)

Table 4.4: Sample values of sensing energy.

$E_{sensing}$	**Total amount of sensing energy (LEACH)**	**Total amount of sensing energy (EDEC)**	**Ratio of** $\frac{LEACH}{EDEC}$
0.00005	1.03846	1.038	1
0.0005	21.04	16.06	1.31
0.001	32.51	22.27	1.46
0.005	96.81	58.22	1.663
0.01	138.77	81.68	1.699

As we can see, the more energy sensing unit consumes, the better performance the EDEC achieves compared to the LEACH, because the ratio of the LEACH's total amount energy of sensing over the EDEC's keeps increasing when unit energy of sensing increases. This implied that the EDEC takes better advantage of redundant sensing energy than the LEACH.

Figure 4.6 shows the number of nodes alive over simulation time given different values of $E_{sensing-unit}$. As we can see, for most of the values, the EDEC performs longer lifetime than the LEACH. It agreed with the theoretical analysis that the more percentage the energy of sensing unit takes, the much faster LEACH nodes died than EDEC nodes because the EDEC avoids overusing the energy of sensing units.

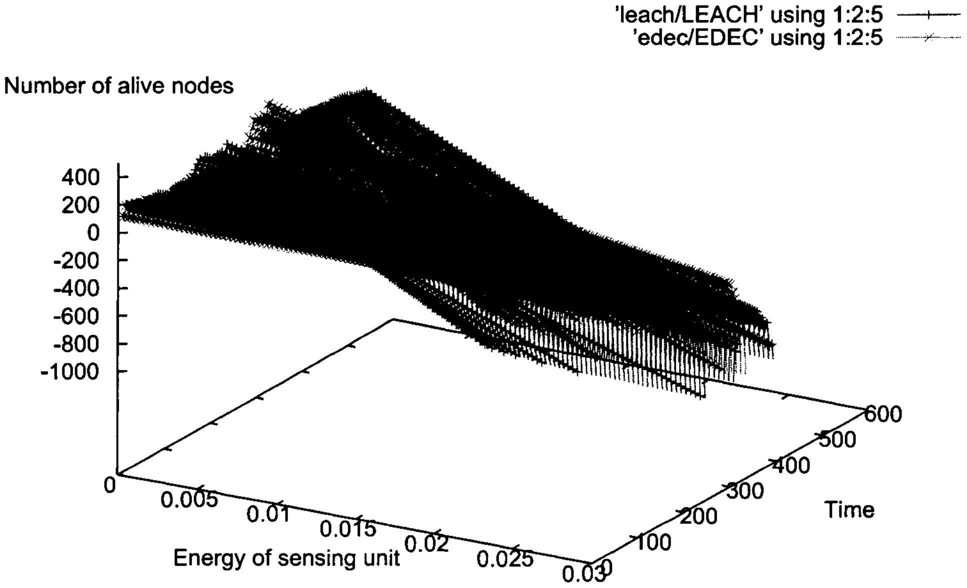


Figure 4.6: Network lifetime.

However, from Figure 4.6, we can observe that some of LEACH values are better than the EDEC. This is also consistent to our expectation because those values are all small amount energy ($\leq 0.00005J$) which implies that the overhead to run the

EDEC algorithm is higher than the benefits the EDEC can trade-off. But for most of the values of $E_{sensing-unit}$, the EDEC performs better. Therefore, for those applications where sensors need certain amount of energy to do sensing task, the EDEC will perform better, and as that amount goes higher, the benefits the EDEC can obtain will increase.

4.5 Frequency of Event

The EDEC is an event-driven algorithm. Only interesting events will let the network be "really active". This is the key reason why the EDEC can have longer lifetime because it stays at very low-level operation like "hibernation". When interesting events happen, the alert sensors will wake up neighborhood to co-monitor the environment. Therefore, we can theoretically conclude that the less frequent the events happen, the longer lifetime the network can live.

In the EDEC, we use frequency of event to simulate the real-life event happening. For the LEACH, the parameter is always set to be 1, because it assumes that in every time slot, each sensor has effective data. However, in the EDEC, we tried different values from 0.1 to 1, to see how the EDEC algorithm performs under different scenarios.

Figure 4.7 presents the relationships between network lifetime and frequency of events by applying different F_{event}.

As we can see from Figure 4.7, the simulation results are consistent with the theoretical conclusion. With different values of F_{event}, EDEC performs longer lifetime than LEACH. And as the values of F_{event} decrease, the network died slower. This is because the time during which the network stay at low-level operation becomes longer and longer so that the redundant consumed energy is effectively avoided.

As we observed Figure 4.7, when the F_{event} increases to 1, EDEC still performs a little bit better than LEACH. This is because we use another parameter $E_{lasting}$ to

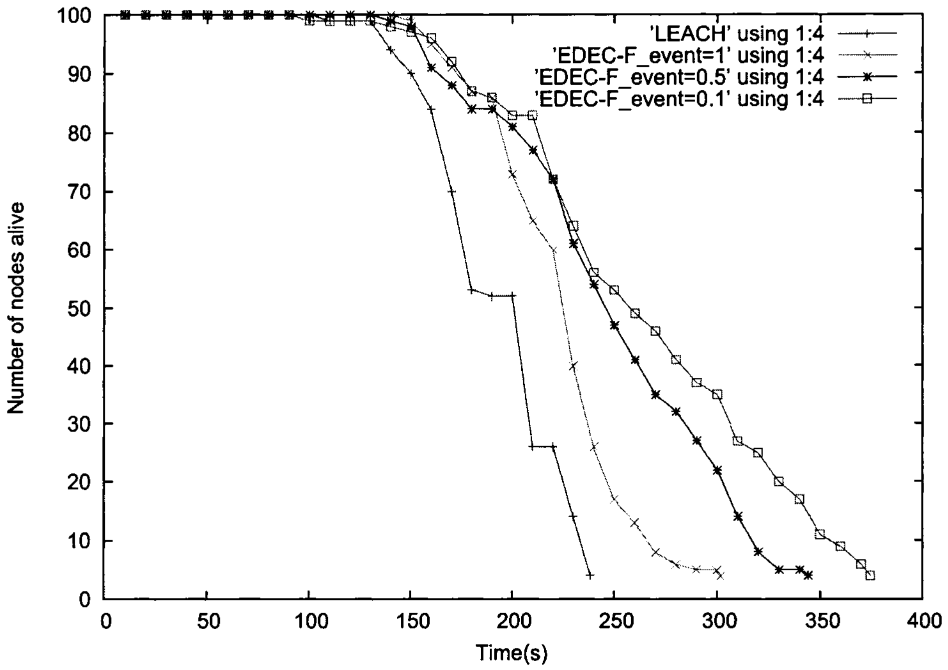


Figure 4.7: Network lifetime by different F_{event}.

control how long a sensing unit needs to take to observe an event. Even when events happen at each time slot, the sensing units could still sleep for a few seconds between events.

Therefore, the simulation results show that the EDEC performs better for event-driven monitoring tasks, especially for those event-sparse scenarios.

4.6 Other Simulation Results

In this section, we did the experiments to compare the LEACH and the EDEC in different perspectives.

Figure 4.8 presents the number of data items received at BS for the LEACH and the EDEC over time. For the comparability, we set the F_{event} to be 1, thus the EDEC always has event happening during each time-slot of each node, same as LEACH. The result is consistent with the expectation that both algorithms have almost the same number of data items over time. This is very important, as [4] states that "one application-independent method of determining quality is to measure the amount of data received at the base station." And Figure 4.8 shows that even though the EDEC schedules the sensing units to put some of them into sleeping mode, the ultimate quality of network of gathering effective data is not compromised. Moreover, the difference between LEACH and the EDEC in Figure 4.8 is that the EDEC has longer lifetime than the LEACH.

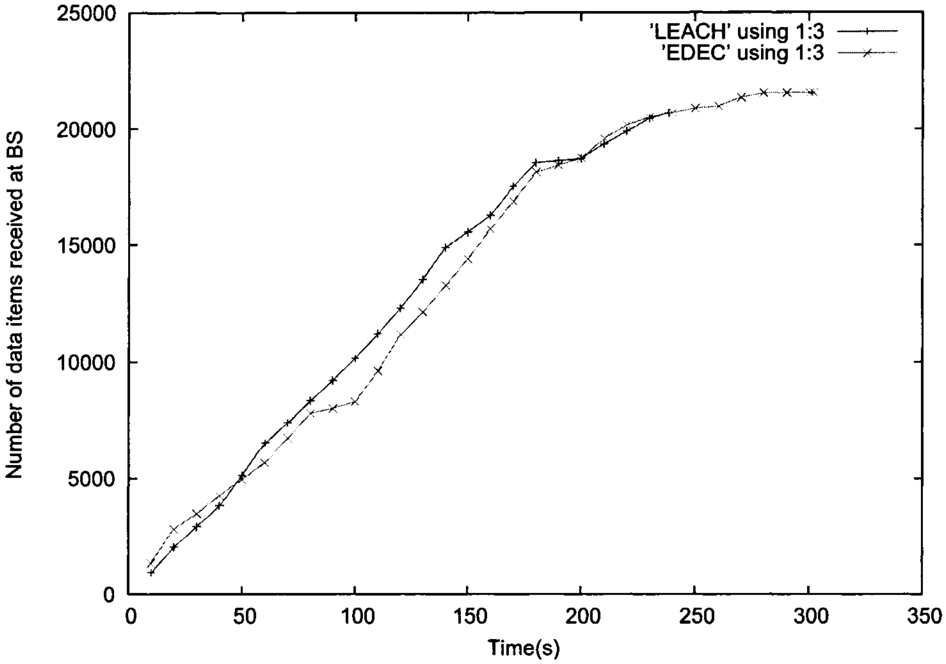


Figure 4.8: Data received at BS over time.

Figure 4.9 presents how the EDEC performs better than the LEACH in terms of energy saving over time.

Figure 4.10 shows the energy efficiency that how much effective data can be delivered by certain amount of energy. As in each data communication, the size of the data is fixed (500 bytes), we use the number of the data communications to represent the

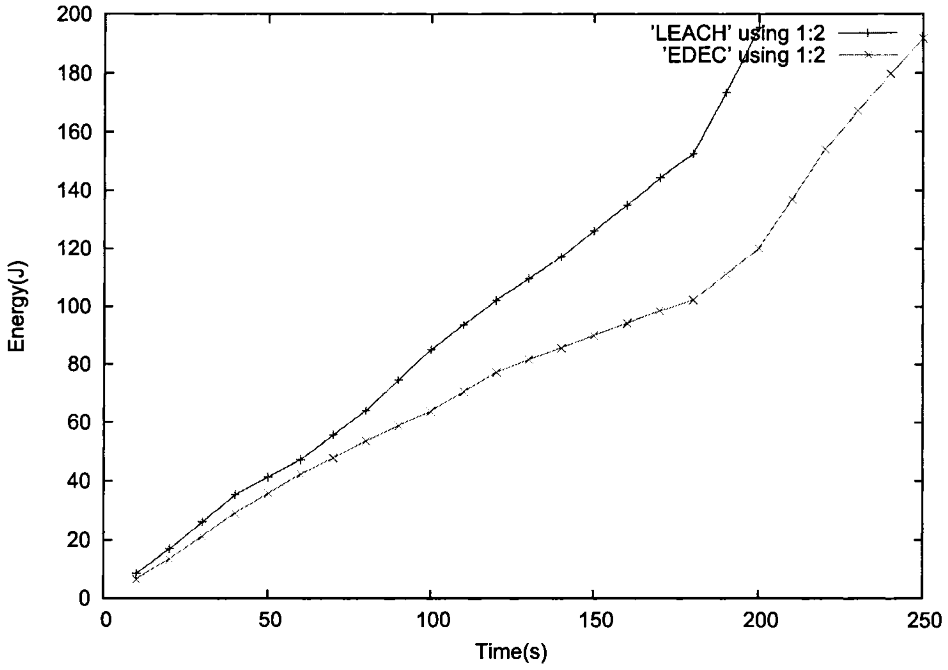


Figure 4.9: Energy dissipated over time.

output. It clearly shows that the EDEC performs about 30% better than LEACH. Moreover, this is based on the configuration of $\{F_{event}=1\}$ for the EDEC. For those sparse events scenarios, the energy efficiency can be expected much higher, because of the saving of energy of sensing units.

Figure 4.11 shows network throughput in terms of alive nodes over delivered data. As seen from figure, the EDEC nodes can produce as much as 50% more effective data than LEACH nodes.

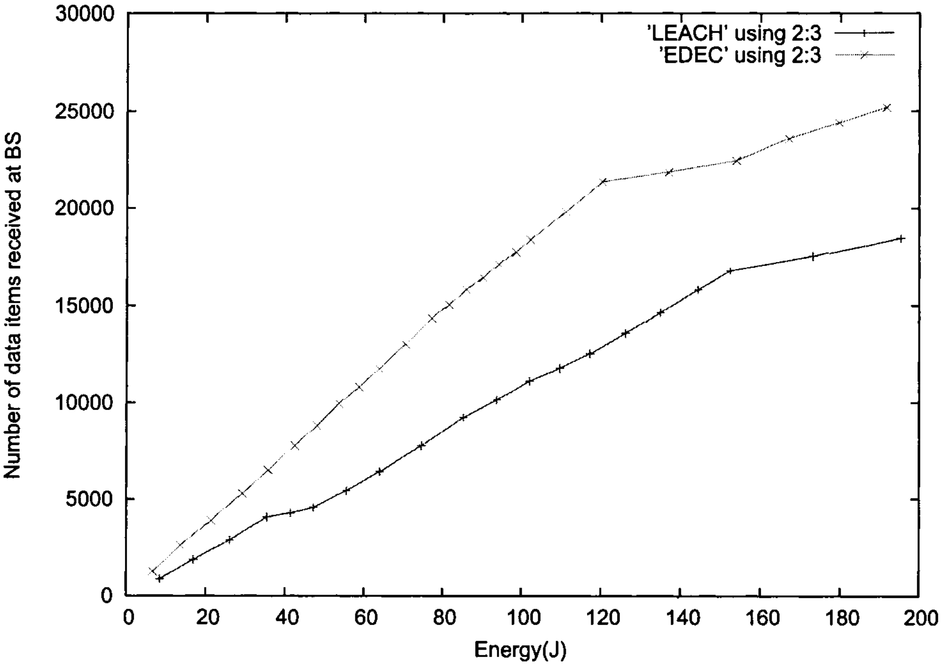


Figure 4.10: Total amount of data per given amount of energy.

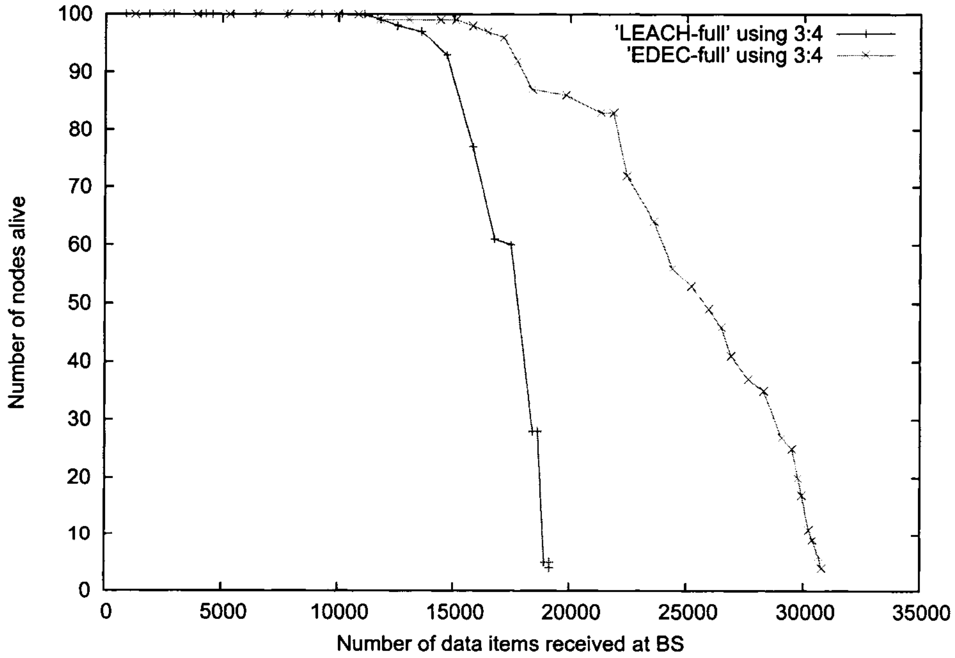


Figure 4.11: Number of alive nodes per amount of data.

Chapter 5

Conclusions and Future Works

We proposed a new scheduling algorithm for sensing units such that the whole network lifetime can be greatly prolonged while the original coverage degree can be maintained. Another attractive feature is that the EDEC separates the tasks of sensing and communicating so that it can adaptively combine with many existing communicating protocols without too many changes on their original designs. The EDEC also brings together the BT technology, coarse/fine coverage, and eligibility calculation without extra communications such that the energy consumed for sensing task is optimized. The experimental results have shown that the EDEC is effective compared to original communicating protocol LEACH in terms of energy efficiency and network lifetime.

In order to more accurately simulate the real environment, we need to create an event-driven simulator. Such a simulator would be able to measure the probability of missed detections and false alarms of events occurring in the environment, especially when testing different R_c values. Thereby, we can get a more realistic and application-specific determination of the network quality by taking account of the energy wasted on the false alarms. This is a subject for our future work.

In addition, the coverage calculation process can be explored more by adopting simple broadcasting which is to broadcast the decision of being a working node or being a sleeping node. This would increase the number of sleeping node therefore save even more energy. However, as Section 3.1.2 stated, this broadcast will bring the extra communication overhead. Moreover, as it needs to encode the message which indicates the node status, BT module cannot do it. Instead, communication units have to do this task and this probably influences the working flow of communication units. For example, when a communication unit is deciding to enter into sleeping mode, it has to consider the wake-up time when its sensing unit needs it to notify its neighboring sensing units that this sensing unit has made the decision to be in sleeping mode. Thus, there are tradeoffs between putting more sensing units in sleeping mode and increasing the network overhead as well as complexity to combine communication protocols and the EDEC scheduling. This is another subject for our future work.

Bibliography

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, A survey on sensor networks, IEEE Communications Magazine, Vol. 40, No. 8, pp. 102-114, August 2002.

[2] M. Kochhal, L. Schwiebert, and S. Gupta, Integrating sensing perspectives for better self organization of ad hoc wireless sensor networks, the Journal of Information Science and Engineering, Vol. 20, No. 3, pp. 449-475, May 2004.

[3] G. J. Pottie, and W. J. Kaiser, Wireless integrated network sensors, Communications of the ACM, Vol. 43, No. 5, pp. 51-58, May 2000.

[4] W. B. Heinzelman, Application-specific protocol architectures for wireless networks, PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Tecnology, June 2000.

[5] M. Clarkson, DA systems: smart sensors, Sensors Magazine, http://www.sensorsmag.com/articles/0597/smartsen/index.htm, May 1997, last visited April 5, 2006.

[6] IEEE1451 committee, IEEE Standard 1451.2-1997, Standard for a smart transducer interface for sensors and actuators - transducer to microprocessor commu-

nication protocols and transducer electronic data sheet (TEDS), Institute of Electrical and Electronics Engineers, Inc., September 26, 1997.

[7] A. Chandrakasan, and W. R. Heinzelman, MIT μAMPS Project, http://www-mtl.mit.edu/researchgroups/icsystems/uamps/research, last visited May 5, 2006.

[8] Y. Zhang, Y. Gu, V. Vlatkovic, and X. Wang, Progress of smart sensor and smart sensor networks, Proc. Intelligent Control and Automation (WCICA 2004), pp. 3600-3606, Hangzhou, China, June 14-18, 2004.

[9] J. Kulik, W. Heinzelman, and H. Balakrishnan, Negotiation-based protocols for disseminating information in wireless sensor networks, Wireless Networks, Vol. 8, pp. 169-185, 2002.

[10] D. Culler, D. Estrin, and M. Srivastava, Guest editors' introduction: overview of sensor networks, Computer, vol. 37, No. 8, pp. 41-49, August 2004.

[11] S. Vlaam, Object tracking in a multi sensor network, Master thesis, TNO Physics and Electronics Laboratory, Computer Engineering, Delft University of Technology, August 17, 2004.

[12] S. Tilak, N. Abu-Ghazaleh, W. Heinzelman, A taxonomy of wireless micro-sensor network models, ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 6, No. 2, pp. 28-36, April 2002.

[13] A. A. Ahmed, H. Shi, and Y. Shang, A survey on network protocols for wireless sensor networks, Proc. Information Technology: Research and Education (ITRE2003), pp. 301-305, Newark, New Jersey, USA, August 11-13, 2003.

[14] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks, Proc. the Sixth Annual

International Conference on Mobile Computing and Networking, pp. 56-67, Boston, MA, August 2000.

[15] J. N. Al-Karaki, and A. E. Kamal, Routing techniques in wireless sensor networks: a survey, IEEE Wireless Communications, Vol. 11, No. 6, pp. 6-28, December 2004.

[16] B. Heinzelman, P. Chandrakasan, and H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Communications, Vol. 1, No. 4, pp. 660-670, October 2002.

[17] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, Coverage problems in wireless ad-hoc sensor networks, Proc. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001), pp. 1380-1387, Anchorage, AK, USA, April 22-26, 2001.

[18] M. Cardei, and J. Wu, Coverage in wireless sensor networks, Handbook of Sensor Networks, CRC Press, 2004, ISBN: 0-8493-1968-4.

[19] J. Carle, and D. Simplot-Ryl, Energy-efficient area monitoring for sensor networks, Computer, Vol. 37, No. 2, pp. 40-46, Febuary 2004.

[20] S. Adlakha, and M. Srivastava, Critical density thresholds for coverage in wireless sensor networks, Proc. Wireless Communications and Networking (WCNC 2003), pp. 1615-1620, New Orleans, Louisiana, USA, March 16-20, 2003.

[21] V. Ravelomanana, Extremal properties of three-dimensional sensor networks with applications, IEEE Transactions on Mobile Computing, Vol. 3, No. 3, pp. 246-257, July 2004.

[22] M. Handy, M. Haase, and D. Timmermann, Low energy adaptive clustering hierarchy with deterministic cluster-head selection, Proc. Mobile and Wireless Commu-

nications Networks (MWCN 2002), pp. 368-372, Stockholm, Sweden, September 2002.

[23] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, Data gathering algorithms in sensor networks using energy metrics, IEEE Transactions on Parallel and Distributed Systems, Vol. 13, No. 9, pp. 924-935, September 2002.

[24] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, SPAN: an energy efficient coordination algorithm for topology maintenance in ad hoc wireless networks, Proc. ACM/IEEE Mobicom 2001, pp. 85-96, Rome, Italy, July 2001.

[25] B. Hohlt, The design and evaluation of network power scheduling for wireless sensor networks, PhD Dissertation, Computer Science Division UC Berkeley, May 2005.

[26] D. Tian, and N.D. Georganas, A coverage-preserving node scheduling scheme for large wireless sensor networks, Proc. 1st ACM Workshop Wireless Sensor Networks and Applications, pp. 32-41, Westin Peachtree Plaza, Atlanta, Georgia, USA, September 28, 2002.

[27] Y. Xu, J. Heidemann, and D. Estrin, Geography-informed energy conservation for ad hoc routing, Proc. 7th Annual International Conference on Mobile Computing and Networking, pp. 70-84, Rome, Italy, July 16-21, 2001.

[28] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks, Proc. the 5th Annual IEEE/Mobicom Conference, pp. 174-185, Seattle, WA, USA, August 15-20, 1999.

[29] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, Directed diffusion for wireless sensor networking, IEEE/ACM Transactions on Networking, Vol. 11, No. 1, pp. 2-16, February 2003.

[30] F. Ye, A. Chen, S. Lu, and L. Zhang, A scalable solution to minimum cost forwarding in large sensor networks, Proc. Computer Communications and Networks, pp. 304-309, Scottsdale, AZ, USA, October 2001.

[31] C. Schurgers, and M. B. Srivastava, Energy efficient routing in wireless sensor networks, Proc. Military Communications Conference (MILCOM2001), pp. 357-361, Vienna, Virginia, USA, October 2001.

[32] Y. Yao, and J. Gehrke, The cougar approach to in-network query processing in sensor networks, ACM SIGMOD Record, Vol. 31, No. 3, pp. 9-18, September 2002.

[33] M. Chu, H. Haussecker, and F. Zhao, Scalable information driven sensor querying and routing for ad hoc heterogeneous sensor networks, International Journal of High Performance Computing Applications, Vol. 16, No. 3, pp. 293-313, 2002.

[34] N. Sadagopan, B. Krishnamachari, A. Helmy, The ACQUIRE mechanism for efficient querying in sensor networks, Proc. IEEE International Workshop on Sensor Network Protocols and Applications (SNPA2003), pp. 149-155, Anchorage, AK, USA, May 2003.

[35] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, Proc. the 33rd Hawaii International Conference on System Sciences, pp. 8020, Hawaii, USA, January 4-7, 2000.

[36] A. Manjeshwar, and D. P. Agrawal, TEEN: a routing protocol for enhanced efficiency in wireless sensor networks, Proc. 15th International Parallel and Distributed Processing Symposium (IPDPS'01) Workshops, pp. 2009-2015, San Francisco, USA, April 23-27, 2001.

[37] A. Manjeshwar, and D. P. Agrawal, APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks, Proc. International Parallel and Distributed Processing Symposium (IPDPS 2002) Workshops, pp. 195-202, Marriott Marina, Fort Lauderdale, Florida, USA, April 15-19, 2002.

[38] V. Rodoplu, T. H. Meng, Minimum energy mobile wireless networks, IEEE Transactions on Selected Areas in Communications, Vol. 17, No. 8, pp. 1333-1344, August 1999.

[39] L. Subramanian, and R. H.Katz, An architecture for building self-configurable systems, Proc. the 1st ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2000), pp. 63-73, Boston, Massachusetts, USA, August 11, 2000.

[40] N. Bulusu, J. Heidemann, and D. Estrin, GPS-less low cost out door localization for very small devices, IEEE Transactions on Personal Communications, Vol. 7, No. 5, pp. 28-34, Octobor 2000.

[41] A. Savvides, C.-C. Han, and M. Srivastava, Dynamic fine-grained localization in ad hoc networks of sensors, Proc. the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM 2001), pp. 166-179, Rome, Italy, July 16-21, 2001.

[42] 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, Vol. 12, No. 10, pp. 1023-1032, October 2001.

[43] F. Kuhn, R. Wattenhofer, and A. Zollinger, Worst-case optimal and average-case efficient geometric ad hoc routing, Proc. the 4th ACM International Symposium

on Mobile Ad Hoc Networking & Computing, pp. 267-278, Annapolis, Maryland, USA, 2003.

[44] D. Tian, and N. D. Georganas, Connectivity maintenance and coverage preservation in wireless sensor networks, Proc. the 2004 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2004), pp. 1097-1100, Niagara Falls, Ontario, Canada, May 2-5, 2004.

[45] F. Ye, G. Zhong, S. Lu, L. Zhang, Energy efficient robust sensing coverage in large sensor networks, Technical Report.

[46] J. Rabaey, M. Ammer, J. da Silva, D. Patel, and S. Roundy, PicoRadio supports ad hoc ultra-low power wireless networking, IEEE Computer, Vol. 33, No. 7, pp. 42-48, July 2000.

[47] A. Cerpa, and D. Estrin, ASCENT: adaptive self-configuring sensor networks topologies, IEEE Transactions on Mobile Computing, Vol. 3, No. 3, pp. 272-285, July-September 2004.

[48] C. Guo, L. C. Zhong, and J. M. Rabaey, Low power distributed MAC for ad hoc sensor radio networks, Proc. IEEE GlobeCom 2001, pp. 2944-2948, San Antonio, November 25-29, 2001

[49] M. Miller, and N. Vaidya, Power save mechanisms for multi-hop wireless networks, Proc. First International Conference on Broadband Networks (BROADNETS'04), pp. 518-526, San Jose, California, USA, October 25-29, 2004.

[50] http://www.isi.edu/nsnam/ns/, last visited May 5, 2006.