Constraint-Minimizing Logical Topology for Wireless Sensor Networks

by

Quazi Mamun
M. Sc. in Global Information and Telecommunication Studies
Waseda University, Tokyo, Japan.

A thesis submitted for fulfillment of the
requirements for the degree of
Doctor of Philosophy (0190)

Clayton School of Information Technology
Monash University

March 2011

Typeset in Palatino by TEX and LATEX 2ε.

Declaration

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

....

Quazi Ehsanul Kabir Mamun

March 02, 2011

Dedicated to the memory of my father.

Acknowledgments

Although my name is printed on the cover of this thesis, the word ”I” does not appear within its chapters. I do this to pay tribute to the myriad contributions of my supervisors and mentors, and the support of my family and friends.

To my supervisors Sita, and Srini: Thank you for your constant guidance and endless patience towards the completion of this thesis. Without your insights and encouragement, this thesis would not have been completed.

To my parents: I owe you for your love, affection and sacrifice. Thank you for giving me the freedom and opportunity to pursue my studies, and forgive me for not being there, when you needed me the most.

To my wife Nipa: Thank you for believing in me, and for your patience during so many late nights, many of them unanticipated.

To my daughters Ardii and Aurii: Every morning when you wake me up, I feel joy and inspiration. Thank you for things which you don’t even know you have done for me - I can even write another thesis.

To my three sisters: You are always happy in my every success, thanks a lot for your moral supports.

To Noriako Sato: My mentor of English, the assistance you gave me in writing this thesis is without match. Your dedication, support, and hard work make this thesis readable. I am at a loss for words regarding your benevolence.

To Helen Borowsky: Thank you so much for proofreading my thesis. You have done an excellent job.

To Monash University and the staffs of Clayton School of IT: Thanks for providing me with an excellent research environment, and the necessary resources to undertake this research. Gary, thank you for your time and the intense work you did to fix my PC when it crashed.

To Mortuza: Thanks for teaching me so many things. You offered much advice, and encouragement that was a great source of comfort to me.

Many of my fellow colleagues and friends took time away from their own over-extended lives to provide feedback on this thesis, and for this, I am extremely grateful. I hesitate to include such a list, as I will surely omit some who deserve mention. Unrestrained comments from many anonymous reviewers for my several technical papers provided equal parts of useful feedback, criticism, and disbelief.

I want to thank all of my friends who made life in Melbourne enjoyable, and sociable. Subrata, I will always aspire to match your level of optimism, clarity of purpose, and general zest for life. Rokon, Nabeel, and Daniel, thanks for all the good times.

Finally, I would like to express my profound gratitude to the Almighty for blessing me with the talents and abilities to undertake this research.

Quazi Mamun,
March, 2011.

Abstract

Wireless sensor network (WSN) is currently one of the most promising areas in the field of information and communication technologies (ICT). This new technology has unlimited potential for numerous applications in different areas, including environmental research, medical application, military, transportation, entertainment, crisis management, homeland defense, and smart spaces. However, several constraints of the sensor nodes are the main obstacles for designing efficient protocols for WSNs. The major constraints of WSNs include energy consumption severity, low quality of communication, limited computational resource, and scalability.

In developing various application protocols for WSNs, such as data collection and dissemination protocols, routing protocols, security or synchronization protocols, various researchers have focused primarily on reducing the aforementioned constraints of WSNs. So far, their approaches tend to focus on developing protocols first, and then to use them on different topologies to implement the protocols. This thesis, however, proposes a novel approach for protocol design paradigm. To contend with the constraints of WSNs, this thesis argues to develop the logical topology before the protocols are designed, because logical topology inherently defines the logical structure and the communication abstraction among the sensor nodes, and thus logical topology governs protocol design. An optimized logical topology facilitates the ability of designers to design efficient protocols, which allow the sensors to communicate with each other with little overheads, lower energy consumption, longer lifetime of sensor nodes, and reduced latency. It is thus more intuitive to approach the constraints-minimizing problem of WSNs from the topological point of view. Moreover, the use of logical topology is inevitable when the sheer number of sensor nodes, their unattended deployment, and hostile environment preclude reliance on physical configuration or physical topology. In such circumstances, designers need to rely only on the logical topologies. Hence, this thesis aims to design an efficient logical topology/structure, with which other application protocols can be designed.

Starting with a discussion on the influences of logical topology on different constraints of WSNs, this thesis proposes how sensor nodes should be connected and managed in an efficient way, so that the WSNs are able to provide the services required by the users, and at the same time overcome different constraints. A hierarchical multi-chain oriented logical topology is proposed. This topology exploits the advantages of chain oriented topology, and at the same time, overcomes the problems of chain oriented topology. The proposed logical topology restricts the sensors to communicate only with their successive nodes along the chains, which saves vast energy, and thus helps to lengthen the lifetime of sensor networks. To make the proposed logical topology more efficient, three adaptations to the basic multi-chain oriented topology are proposed and designed. These adaptations are i) scheduling the chain members, ii) creating localized chains, and iii) collecting data by employing mobile data collectors. The first adaptation proposes a scheduling method, which allows a considerable number of deployed sensors to be turned off by sacrificing only a negligible amount of coverage area. The second adaptation proposes a localized chain construction method using *Voronoi* tessellation technique. This scheme not only saves energy, but also reduces interference in the network. The third and final adaptation describes an efficient data gathering scheme, which conserves energy, provides network connectedness, and reduces latency. Extensive simulation experiments are performed to evaluate the performance of each of the adaptations.

Furthermore, to test the proposition on logical topology, a number of application protocols are designed using the hierarchical structure and the communication abstraction of the proposed logical topology. All the applications, one for data collection, one for data dissemination, and the other for data transfer in WSNs, show promising results in respect of energy consumption, network lifetime, as well as latency.

Abbreviations

BS	Base Station
CH	Cluster Head
CHIRON	Chain-based HIerarchical ROutiNg protocol
CL	Chain Length
COSEN	Chain Oriented SEnsor Network
EBCRP	Energy-Balanced Chain-cluster Routing Protocol
ECR	Energy-efficient Chain-cluster Routing protocol
ED	Event Detection
GAF	Geographical Adaptive Fidelity
LEACH	Low-Energy Adaptive Clustering Hierarchy
MDC	Mobile Data Collector
MREF	Maximum Residual Energy First
MTE	Minimum Total Energy
PEAS	Probing Environment and Adaptive Sleeping
PECAS	Probing Environment and Collaborating Adaptive Sleeping
PEGASIS	Power-Efficient GAthering in Sensor Information Systems
SDMA	Spatial Division Multiple Access
SecCOSEN	Secured Chain Oriented SEnsor Network
SPE	Spatial Process Estimation
SPIN	Sensor Protocol for Information via Negotiation
TTL	Time To Live
VD	Voronoi Diagram
WSN	Wireless Sensor Network

Contents

List of Figures

List of Tables

Chapter 1

Introduction

'In 1951, a man could walk inside a computer and by 2010 . . . ,
computers are beginning to "walk" inside of us.'
– C. Gordon Bell, Bell's Law for the birth and death of computer
classes: A theory of the computer's evolution, MSR-TR-2007-146.

1.1 Preamble

With the popularity of laptops, cell phones, PDAs, GPS devices, RFID, and intelligent electronics in the post-PC era, computing devices have become cheaper, more mobile, more distributed, and more pervasive in daily life. Using the commercial on-the-shelf (COTS) components, it is now possible to construct a wallet-size embedded system with the equivalent capability of a 90's PC. Such embedded systems can be supported with scaled down Windows or Linux operating systems. From this perspective, the emergence of wireless sensor networks (WSNs) is essentially the latest trend of Bell's Law toward the miniaturization and ubiquity of computing devices.

WSN technology is expected to have a significant impact on our lives in the twenty-first century [Akyildiz et al., 2007]. This is because of the increasing advances in the past decade in the areas of microelectronics, sensing, analog and digital signal processing, wireless communication and networking [Warneke and Pister, 2002]. Wireless sensor networks are made up of a large number of inexpensive devices that are networked via low power wireless communications [Akyildiz et al., 2002]. It is this networking capability that fundamentally differentiates a sensor network from a mere collection of sensors, and this enables cooperation, coordination and collaboration among sensor assets. Figure 1.1 depicts how a sensor network collects data

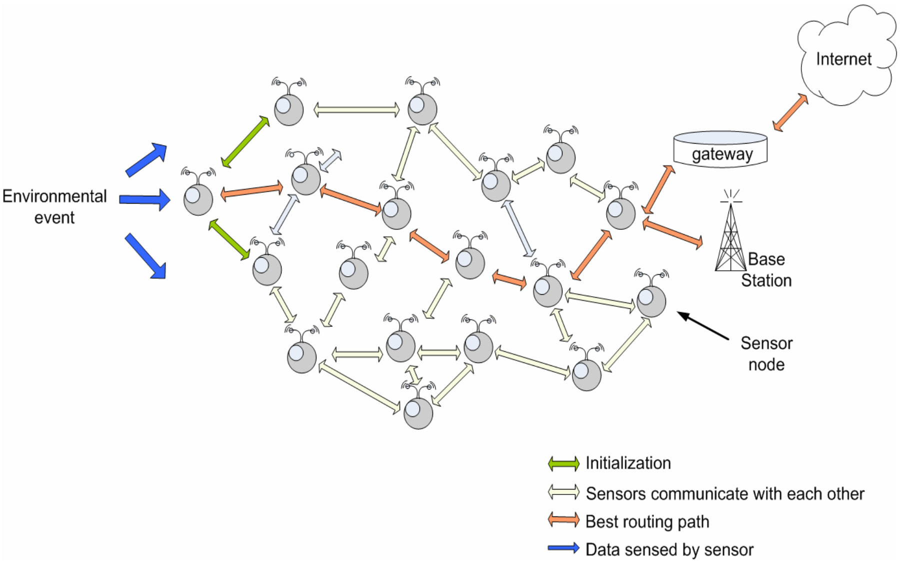


Figure 1.1: A wireless sensor network: environmental data are sensed and delivered to the base station/gateway using best routing path.

from the environment and passes the data to the user/Internet using sink nodes and gateways. Various applications of sensor networks have been proposed in areas such as environmental monitoring [Ye et al., 2009; Mahdy, 2008; Zhigang and Hui, 2009; Martinez et al., 2004], natural disaster prediction and relief [Shen et al., 2008], homeland security [Haupt et al., 2007; Lee and Reichardt, 2005], healthcare [Ming et al., 2009; Kim et al., 2008; Mascarenas et al., 2009], manufacturing [Evans, 2005; Zurawski, 2009], transportation [Benliang et al., 2006; Peng et al., 2009; Mamun et al., 2006], mining [Wang et al., 2009], home appliances [Kim et al., 2007; Suh and Ko, 2008; Baeg et al., 2007] and entertainment [Verdone et al., 2008].

However, at the same time as WSNs enjoy enormous application potentials in different fields, they suffer from various constraints. The most important constraint is limited energy, which is caused by the failure to replace batteries or power sources, and by the absence of wires [Walsh et al., 2008]. Unfortunately, limited energy of WSNs causes network lifetime and connectedness problems. These severities largely affect the designing of protocols. Other major constraints that affect designing different algorithms and protocols for WSNs are unreliable and low quality communication, limited computational resource, and scalability. For this reason, minimizing the

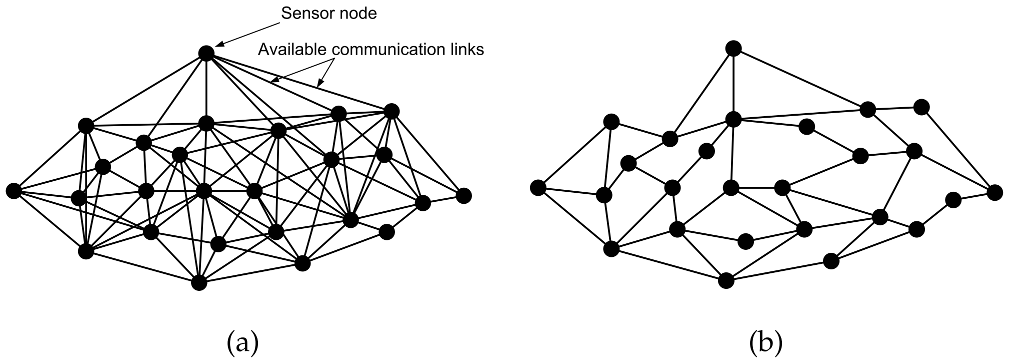


Figure 1.2: Communication graph and logical topology of WSN. (a) Communication graph shows all available communication links among the sensor nodes; and (b) Logical topology is created by selecting a set of edges from the communication graph.

constraints of WSNs is the underlying theme of this present work. In this thesis, the constraint minimizing problems of WSNs are studied from topological point of view.

Network topologies can be physical or logical. Physical topology refers to the physical design of a network including the devices, location and cable installation. On the other hand, logical topology refers to how data are actually transferred in a network as opposed to its physical design. Usually WSNs are formed by a large collection of power-conscious wireless-capable sensors without the support of pre-existing infrastructure, possibly by unplanned deployment. With a sheer number of sensor nodes, their unattended deployment and hostile environment very often preclude reliance on physical configuration or physical topology. It is, therefore, often necessary to depend on the logical topology.

The logical topology of a wireless sensor network is formed by the communication graph of the network. A communication graph of a WSN is an undirected graph $G = (V, E)$ where V denotes the sensors deployed, and E denotes the available communication links among the sensor nodes. An example of a communication graph is shown in Figure 1.2(a), whereas Figure 1.2(b) shows a logical topology, which is created from the communication graph by selecting a subset of the edges.

From this point of the thesis, 'topology' refers only to 'logical topology'. The words 'topology' and 'logical topology' are used interchangeably, if otherwise not specified.

1.2 Motivation

Wireless sensor networks differ fundamentally from general data networks, such as the Internet, and they require the adoption of a different design paradigm. WSNs are often application specific, and they are designed and deployed for special purposes. Thus the network design must take into account the specific intended applications. For battery-operated sensors, energy conservation is one of the most important design parameters, since replacing batteries is difficult in many applications, if not impossible [Yebari et al., 2008]. As a result, sensor network design must be optimized so that the energy conservation is efficient and thus extends the network lifetime. The energy and bandwidth constraints and the potential large-scale deployment pose challenges to efficient resource allocation and sensor management.

In many applications, the key features of a sensor network are wirelessness and random deployment. Sensor nodes are usually placed in hostile or unattended environments [Yebari et al., 2008], such as battle fields, sea beds, deep forests etc. Thus the use of wired node-to-node connections is highly difficult, if not infeasible. It can be logistically challenging and also expensive, especially when hundreds of sensors are envisioned. For a large scale WSN, the most important concerns are limited energy, unreliability, network connectedness, latency, scalability, and communication overhead [Paschalidis et al., 2007; Zhao et al., 2004a; Quan et al., 2007; Cai et al., 2006]. All of these issues are nearly inevitable topics in wireless sensor network design as they impose strict constraints on the network operations. To address these issues, logical topology provides an effective approach to contend with these constraints of WSNs, because it is logical topology that governs the sensor nodes to communicate with each other.

Many contemporary researchers have devoted themselves in designing the protocols/algorithms to contend with the constraints of WSNs. Numerous protocols such as routing protocols, data aggregation protocols, base station (BS) positioning protocols etc. have been designed to solve the constraint minimizing issues. While various theories and applications have been proposed, this thesis argues that logical topology of WSNs should be considered before designing the protocols, as logical topology inherently defines the communication paths among the sensor nodes. As the main tasks of a sensor network are to sense the events, generate data, and to disseminate these data to the BS or sink node(s), the following aspects are crucial:

i) to determine and establish the route of the dissemination process,

ii) to determine which node is responsible for the transfer of data to the BS,

iii) to choose the node where data fusing can take place, and

iv) to decide which node precedes which other nodes in respect of data sending or receiving.

Logical topology of WSNs actually deals with all the issues listed above. It is thus more intuitive to approach the constraints-minimizing problem of WSNs from the topological point of view, and to design such a topology according to the conservation requirements. This thesis argues that topology provides an intuitive way to address those issues. To support this argument, the following sections discuss the inherent nature of topology in various levels of WSNs.

Topology plays a vital role for WSNs. Energy consumption is proportional to the number of packets sent or received. The receiving cost depends on packet size, while the transmission energy depends on the distance between the nodes. As topology inherently defines the type of routing paths, indicates whether to use broadcast or unicast, determines the sizes and types of packets and other overheads, choosing the right topology helps to reduce the amount of communication needed for a particular problem. Thus energy can be saved. An efficient topology, which ensures that neighbours are at a minimal distance, reduces the probability of message being lost between sensors. A topology can also reduce the radio interference, thus reducing the waiting time for sensors to transmit data. Moreover, topology facilitates data aggregation, which greatly reduces the amount of processing cycles and energy, thus giving a longer lifetime for the network.

In addition, topology inherently defines the size of a group, how to manage new members in a group or how to deal with members who have left the group. With the awareness of the underlying network topology, more efficient routing or broadcasting schemes can be achieved. Furthermore, the network topology in WSNs can be changed by varying the nodes' transmitting ranges and also by adjusting the wake /

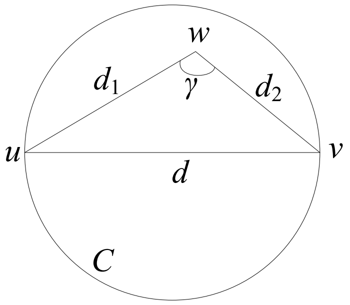


Figure 1.3: Topology and energy consumption.

sleep schedule of the nodes. Therefore, more energy can be saved if the network topology is maintained in an optimal manner. Below are two elaborations which affirm the significance of logical topology in WSNs in this context.

1.2.1 Topology and energy consumption

As outlined previously, to achieve efficient use of the scarce energy resources available to sensor network nodes is one of the fundamental tasks of the network designer. Since nodes consume a considerable amount of energy to transmit/receive messages, reducing the energy consumed for radio communications is an important issue. Suppose a node u must send a packet to a node v, which is at distance d (see Figure 1.3). Node v is within node u's transmitting range at maximum power, so direct communication between u and v is possible. However, there also exists a node w in the region C circumscribed by the circle of diameter d that intersects both u and v (see Figure 1.3). Assume Euclidian distance between nodes u and w, $\delta(u, w) = d_1$ and Euclidian distance between nodes v and w, $\delta(v, w) = d_2$. Since the values of both d_1 and d_2 are less than d, sending the packet using w as a relay is also possible. There is a question of which of the two alternatives is more convenient from the energy-consumption point of view. To answer this question, specific wireless channel and energy consumption models can be referred to.

For simplicity, assume that the radio signal propagates according to the free space model, and that the issue is to minimize the transmit power only. For these assumptions, the power needed to send the message directly from u to v is proportional to d^2. In case the packet is relayed by node w, the total power consumption is proportional

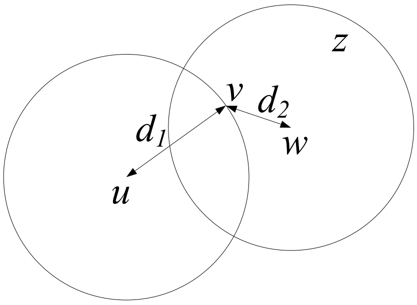


Figure 1.4: Conficting wireless transmission.

to $d_1^2 + d_2^2$. Consider the triangle $\triangle uwv$, and let γ be the angle opposite to side uv. By elementary geometry, it is found that $d^2 = d_1^2 + d_2^2 - 2d_1 d_2 cos\gamma$

Since the circle C contains the point w, $cos\gamma \leq 0$ and thus, $d^2 \geq d_1^2 + d_2^2$, it follows that, from the energy-consumption point of view, it is better to communicate using short, multi-hop paths between the sender and the receiver.

The above observation gives the first argument in favor of topology; instead of using a long, energy-efficient edge, communication can take place along a multi-hop path composed of short edges that connects the two endpoints of the long edge.

1.2.2 Topology and network capacity

Contrary to the case of wired point-to-point channels, wireless communications use a shared medium, the radio channel. The use of a shared communication medium implies that particular care must be made to avoid concurrent wireless transmissions from corrupting each other. A typical conflicting scenario is depicted in Figure 1.4. In this figure, node u is transmitting a packet to node v using a certain transmit power P. At the same time, node w is sending a packet to the node z using the same power P. Since $\delta(v,w) = d_2 < \delta(v,u) = d_1$, the power of the interfering signal received by v is higher than that of the intended transmission from u, and the reception of the packet sent by u is corrupted.

Note that the amount of interference between concurrent transmissions is strictly related to the power used to transmit the messages. This important point can be clarified with the following example. Assume that the node u must send a message to the node v, which is experiencing a certain interference level λ from other concurrent radio communications. For simplicity, treat λ as a received power level, and assume that a packet sent to the node v can be correctly received only if the intensity of the received signal is at least $(1+\eta)^{\lambda}$, for some positive η. If the current transmit power P used by the node u is such that the received power at the node v is below $(1+\eta)^{\lambda}$, the correct message reception is ensured by increasing the transmit power to a certain value $P' > P$ such that the received power at the node v is above $(1+\eta)^{\lambda}$. This seems to indicate that increasing transmit power is a good choice to avoid packet drops due to interference.

On the other hand, increasing the transmit power at the node u increases the level of interference experienced by the other nodes in u's surrounding. So, there is a tradeoff between the 'local view' (u sending a packet to v) and the 'network view' (reducing the interference level in the whole network). In the former case, a high transmit power is desirable, while in the latter case, the transmit power should be as low as possible. Then the following question arises: how should the transmit power be set, if the designer's goal is to maximize the network traffic carrying capacity?

In order to answer this question, an appropriate interference model can be used. The simplest model of interference is the 'Protocol Model' used in [Gupta and Kumar, 2000] to derive upper and lower bounds on the capacity of *ad hoc* networks. In this model, the packet transmitted by a certain node u to a node v is correctly received if $\delta(v,w) \geq (1+\eta)\delta(u,v)$, where any other node w is transmitting simultaneously, and where $\eta > 0$ is a constant that depends on the features of the wireless transceiver. Thus, when a certain node is receiving a packet, all the nodes in its interference region must remain silent in order for the packet to be correctly received. The interference region is a circle of radius $(1+\eta)\delta(u,v)$ centered around the receiver. In a sense, the area of the interference region measures the amount of wireless medium consumed by certain communications. Since concurrent non-conflicting communications occur only outside each other's interference region, this is also a measure of the overall network capacity.

Suppose that the node u must transmit a packet to the node v, which is at distance d (see Figure 1.5). Furthermore, assume that there are intermediate nodes $w_1, w_2, \cdots, w_k$ between the node u and the node v and that:

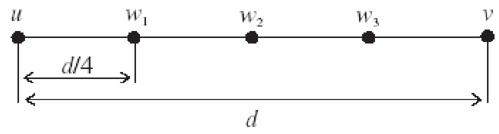


Figure 1.5: Topology and network capacity.

$$\delta(w_1, w_2) = \cdots = \delta(w_k, v) = \frac{d}{k+1}$$

Now, if the node u wants to send data to the node v, there remain two routes - i) u can send data directly to v, or ii) u can use the multi-hop path $u, w_1, w_2, \ldots, v$. Which of them is better in terms of network capacity can be easily identified by considering the interference range(s) in the two scenarios. In case of direct transmission, the interference range of node v is $(1+\eta)d$, corresponding to an interference region of the area $\pi d^2(1+\eta)^2$. In case of multi-hop transmission, the areas of the interference regions of each short, single-hop transmission need to be summed. The interference region for any such transmission is

$$\pi\left(\frac{d}{k+1}\right)^2(1+\eta)^2$$

There are $(k+1)$ regions to consider overall. Hence, by Holder's inequality,

$$\sum_{i=1}^{k+1}\left(\frac{d}{i+1}\right)^2 = (k+1)\left(\frac{d}{k+1}\right)^2 < \left(\sum_{i=1}^{k+1}\frac{d}{i+1}\right)^2 = d^2$$

Thus, it can be concluded that, from the network capacity point of view, it is better to communicate using short, multi-hop paths between the sender and the destination.

The above observation is the other motivating reason for a careful design of the network topology: instead of using long edges in the communication graph, it is better to use multi-hop paths composed of shorter edges that connect the endpoints of the long edge.

All the discussions and examples of this section infer that logical topology plays a vital role for wireless networks, especially for resource-constraint WSNs. Moreover, logical topology facilitates WSNs in many ways to overcome different constraints such as minimizing energy consumption, maximizing lifetime, reducing interference, making networks scalable etc. That is the reason why this thesis considers logical topology as the best way to approach the constraints minimizing problems of WSNs.

1.3 Objectives of the Thesis

In the last section it is shown how logical topology contributes to minimizing constraints and maximizing output for WSNs. When an application protocol is designed, the designer should always consider the resource-constraint nature of WSNs. Thus, it is argued in this thesis that logical topology should be considered before designing the application protocols. This expresses the other aim of the thesis, which is to propose a protocol design paradigm for WSNs. In doing so, four different scenarios are considered in order to derive more detailed objectives. Each of the scenarios describes a particular use of WSNs, and these scenarios cover most of the usages of WSNs, if not all. The scenarios are:

- **Scenario 1:** The user needs to collect data periodically from the deployed sensor nodes. This is the most common use of WSNs, and this includes such tasks as collecting data from a target field or monitoring a target field.

- **Scenario 2:** The user needs to disseminate some information to all deployed sensor nodes. For example, the user needs to set a new threshold value for the sensors, which are being used to detect forest fire. Another example of this scenario is that the user wants to disseminate secret keys in the WSN, so that the sensor nodes can communicate with each other in a secured way.

- **Scenario 3:** A sensor or group of sensors deployed in a part of the target field need to communicate with another sensor or a group of sensors deployed in another part of the target field. For example, if a group of sensors G_1 detect some event e in location x, G_1 needs to communicate with another group of sensors G_2,. The G_2 is in location y, and would trigger an action in response to the message received from G_1 for event e.

- **Scenario 4:** The user needs the deployed sensors to perform some action, e.g., the user needs to synchronize the clocks of the sensor nodes in the target field.

Note that, in all the scenarios mentioned above, deployed sensor nodes need to communicate with each other. To solve the problems encountered in the above mentioned scenarios, the following approach has traditionally be taken: designing a data collection protocol for Scenario 1 [Rothery et al., 2008; Zhang et al., 2009; Yang et al., 2009]; designing a data dissemination protocol for Scenario 2 [Hamida and Chelius, 2008; Zhang and Wang, 2008]; designing a routing protocol for Scenario 3 [Al-Karaki and Kamal, 2004; Zhaohua et al., 2010]; and designing a time synchronization protocol for Scenario 4 [Sivrikaya and Yener, 2004; Wang and Wang, 2007]. Traditionally, in constructing these protocols, researchers have used different logical structures, and different communication abstractions. However, this thesis approaches the protocol designing problems in a different way. This thesis argues that logical structure and communication abstraction (i. e., logical topology) should be built first, because in each scenario, all the sensors are just communicating with each other and doing nothing else. Thus if the logical topology is built first, this makes it easier to devise the rest of the designing parts. Moreover, a well-designed logical topology helps to contend with the constraints of WSNs. For this reason, this thesis primarily aims to *design a constraint minimizing logical topology for WSNs.*

Designing a logical topology encompasses many issues, such as node scheduling, logical structure of the topology, communication model, data collection and dissemination techniques etc. This thesis thus focuses on achieving the following design specific objectives:

- To design a logical topology for WSNs. The logical topology would demonstrate the way the sensor nodes should be managed so that the WSN is able to provide connected coverage to the entire area of interest. At the same time, the logical topology should preserve energy, have increased lifetime, retain scalability, and experience reduced overheads for different types of communications.
- To design a node scheduling protocol to select nodes from the deployed sensor nodes. The selected nodes by the protocol will then participate to construct the logical topology of the sensor network.
- To design an algorithm with which the selected nodes can identify themselves in a group to construct the logical topology. The design has to be fully distributed, because a centralized algorithm needs global synchronization overhead, and is not scalable to large-populated sensor networks [Zhao and Raychaudhuri, 2009].

- To design an efficient topology structure construction algorithm, and here topology structure can be defined as the type of organization of the sensor nodes. For example, for a cluster based logical topology, topology structure construction algorithm means an algorithm for designing the clusters of the sensor nodes.

- To design an efficient data collection method. The method has to be designed in a way to help conserve energy, and at the same time to take less convergence time.

Moreover, after designing logical topology, this thesis aims to evaluate the performance of the constructed logical topology. The objectives specific to this evaluation are:

- To identify the performance metrics to evaluate different topologies of WSNs. These metrics will be used to compare different aspects of WSNs, such as energy consumptions rate, lifetime patterns, communication overhead, latency and scalability of the networks.

- To design and apply different types of application protocols to run on the top of the logical topology to evaluate the efficiency of the logical topology.

It should be noted that logical topology problems are sometimes confused with routing problems in WSNs. The aim of this thesis is not to design a routing protocol but to construct a logical topology for WSNs. Although logical topologies of wireless networks inherently define routing paths, the problem is not limited to delivering data from the source to destination node(s). The logical topology designs the logical structure, and gives the communication abstraction, while routing protocols can be established on the basis of the logical abstraction. Moreover, the thesis aims to design a logical structure of deployed sensors, with which other protocols (e.g., data dissemination or data collection protocols, time synchronization protocols, event synchronization protocols and other different application protocols) can be designed. This thesis argues that an optimized logical topology facilitates the designers to design efficient protocols, which allow the sensors to communicate with each other with little overheads, comparatively low energy consumption, longer lifetime, reduced latency and other improved facilities. The scope of logical topology problems is comparatively bigger, as it includes network connectivity, node communication, group and network management and scalability, whereas routing protocols only deal with finding paths from source to destination.

1.4 Research Scope of the Thesis

In this section, the scope of the research of this thesis is identified. As WSNs are very much application dependent, this thesis asserts the types of WSNs applications for which the research applies.

There are different types of application classifications for WSNs. One of the possible classifications of WSNs applications distinguishes applications according to the type of data that must be gathered in the network. Almost any application, in fact, could be classified into two categories: event detection (ED) and spatial process estimation (SPE) [Buratti et al., 2009]. In the first case, sensors are deployed to detect an event, for example, a forest fire, an earthquake etc. [Shastry et al., 2005; Quek et al., 2007; Toriumi et al., 2008]. Whereas in SPE, WSNs aim at estimating a given physical phenomenon (e.g., the atmospheric pressure in a wide area, or the ground temperature variations in a small volcanic site), which can be modeled as a bi-dimensional random process. In this case, the main issue is to obtain the estimation of the entire behaviour of the spatial process based on the samples taken by sensors that are typically placed in random positions [Simic and Sastry, 2003; Nordio et al., 2008; Dardari et al., 2007; Behroozi et al., 2008].

Another possible classification of WSNs applications can be on the basis of how frequent the sensors sense, and send the data to the BS. According to this classification the applications are divided into four categories: i) Continuous / Periodical data gathering, ii) Event-driven data gathering, iii) On demand data gathering and iv) Hybrid type data gathering

Besides the application classifications of WSNs, the sensors nodes construct different types of networks too. One classification is based on the similarity of the sensor nodes. A sensor network is called homogeneous if all the sensor nodes are of similar type, whereas a network is called heterogeneous if the sensor nodes are of different types. The network can also be classified based on the number of sensor nodes in the network. For example, a network where the number of sensor nodes does not exceed hundred can be classified as small/normal-sized WSNs. On the other hand, if the number of nodes in a WSN exceeds hundreds, or even thousands, it is called a large-scale network.

The research scope of this thesis is based on WSNs of the following categories: i) Large Scale sensor networks, ii) Either ED or SPE category, iii) Continuous / Periodical data collection category and iv) Homogeneous sensor networks category.

The research of this thesis is *not* intended for the following types of WSNs: i) Body Sensor Networks, ii) Vehicular sensor networks, iii) Machine-to-Machine, iv) Acoustic (underwater) Sensor network, or v) Interplanetary Sensor Networks etc.

1.5 Contributions of the Thesis

The thesis investigates different logical topologies that are found as the underlying structures of different protocols, such as flat topology, cluster based topology, chain oriented topology and tree based topology. The thesis then identifies the evaluation metrics and compares all topologies using the same performance metrics. Using the comparative evaluation, it is argued in this thesis that chain oriented topology has more potential to minimize the constraints of WSNs than any other topologies. The main contributions of the thesis are outlined below:

- This thesis demonstrates that the issues of WSN constraints are well-addressed by the logical topology. Arguing that a well-designed topology facilitates designing application protocols to minimize different constraints, this thesis brings a novel notion in application design paradigm of WSNs.

- This thesis proposes a multi-chain oriented logical topology for WSNs. By designing a multi-chain oriented logical topology and comparatively evaluating the topology with other topologies using various metrics, it is demonstrated that several WSN constraints can be addressed by the logical topology. In particular, the proposed multi-chain oriented topology saves a significant amount of energy in comparison to other topologies for both data collection and dissemination applications, and thus assists the sensors to produce longer lifetime of the network. The proposed topology is scalable and also helps the sensor nodes to send data to the BS more quickly.

Besides designing the multi-chain oriented logical topology, three adaptations of the basic multi-chain oriented logical topology are also proposed in this thesis. Various protocols are devised for the adaptations, which are mentioned below:

- A novel node scheduling algorithm is proposed to select the chain members for the proposed logical topology. The proposed scheduling algorithm saves a significant amount of energy by compromising a small amount of coverage area.
- This thesis then analyzes the limitations of the basic multi-chain oriented logical topology, and shows how to overcome those limitations using a tessellation method.
- A distributed algorithm is then proposed to construct tessellation planes. The proposed algorithm is fully distributed and has very little communication overhead.
- A chain construction algorithm is then presented to create a single chain in each tessellation plane. The algorithm guarantees minimum energy consumption by the created chains.
- To save more energy, an efficient data collection method is then proposed. The proposed method employs mobile data collectors in the sensor network. Thus burdens of sending messages to distant nodes/BS are taken out from the sensor nodes. As a result, the lifetime of the network is extended significantly.

Moreover, to show the applicability of the multi-chain topology, different application protocols are designed. These application protocols are designed using the logical structure and the communication abstractions of the proposed logical topology, and applied on the top of the proposed topology. The application protocols are chosen to follow different aspects of WSNs. The first aspect to test is the data collection process. The second aspect to test is the data dissemination process, while the third aspect is to test the data transfer from a source node to the destination node in the WSN. All the application protocols provide excellent results, and warrant further application protocols to be designed using the proposed logical topology.

1.6 Thesis Organization

The rest of the thesis is organized as follows. First, Chapter 2 presents the fundamentals of WSNs from the perspective of this thesis, and provides a review of the relevant literature in the area of logical topologies of WSNs. In this chapter, different constraints of WSNs and logical topology evaluation metrics are identified. Several underlying logical topologies are then investigated, and compared in respect to the constraints using the evaluation metrics. This chapter affirms that chain oriented topology is the most effective amongst the other topologies.

Following the results in Chapter 2, Chapter 3 proposes a basic multi-chain oriented logical topology for WSNs. This chapter describes the chain formation, management and communication issues. At the end of this chapter, the scopes to improve the basic logical topology are described.

Chapters 4, 5 and 6 then study various adaptations of the basic logical topology in the areas of node scheduling, chain construction and data collection method. Chapter 4 proposes a node scheduling algorithm for selecting the member nodes of the proposed multi-chain oriented logical topology. It is shown that a significant amount of energy can be saved sacrificing little coverage. Chapter 5 then uses a tessellation method that helps in creating more efficient logical chains. A distributed algorithm is proposed to create the tessellation graph to divide the target field into several smaller regions. Then another algorithm is proposed to create logical chains in each of the smaller regions. Chapter 6 further extends the proposed logical topology, by adding a data collection method to the logical topology. The proposed data collection method employs mobile data collectors, whose routing paths are designed using the tessellation method.

Chapter 7 shows how several application protocols can be adopted on the top of the logical topology. This chapter presents several data oriented application protocols, which use the hierarchical structure and the communication abstraction of the proposed logical topology. Using the experimental results, this chapter concludes that a well-designed logical topology influences the designing of various application protocols for WSNs.

Finally, Chapter 8 draws the conclusion of the thesis.

Chapter 2

Background

2.1 Preamble

In the last chapter, it was argued that topology plays a vital role in minimizing various constraints of WSNs, such as limited energy, latency, network capacity etc. Various researchers have designed numerous protocols/algorithms for WSNs for different purposes, and in those protocols/algorithms they have used various underlying logical topologies. However, to the author's best knowledge, there has not been any existing approach adopted by the researchers to solve constraint minimizing problems of WSNs from the topological point of view. The aims of this chapter are to identify underlying topologies in different protocols/algorithms designed for WSNs, and to compare these topologies using various performance metrics. As the main aim of this thesis is to construct an efficient logical topology for WSNs, evaluating these existing topologies would be very important.

This chapter primarily consists of three parts: preliminaries, descriptions of different topologies, and performance evaluation of the topologies. The first part presents introductory materials that are preparatory for what is described in the rest of the chapter. The architecture of a sensor node is described first. This discussion helps in the understanding of the characteristics of the wireless sensor nodes. A brief description is then provided regarding the state of progress of the current sensor network technology, and the main challenges that the designer/researchers have to face. In conjunction with this discussion, the requirements and constraints of WSNs are identified.

Following the preliminary discussions, this chapter provides detailed descriptions of existing topologies, which are identified from different protocols for WSNs. To design various protocols for WSNs, different underlying logical topologies have been used. Each topology has its own advantages and disadvantages under a specific working environment. Because of this, to compare and evaluate the effectiveness of each topology, a set the performance evaluation metrics is required. In doing so, this chapter also focuses on two major issues, namely i) the system model of the WSN, which would be used throughout the thesis and ii) the list of performance metrics to evaluate existing topologies.

The third and final part of this chapter presents a comparative discussion of performance of different existing topologies of WSNs. In doing so, various performance metrics are first identified, and then detailed comparisons among identified topologies are provided for each of these metrics. The chapter ends with a table, which shows the comparison summary of the performance evaluation of different topologies.

2.2 Characteristics of Sensor Nodes

Wireless sensor networks are built with numerous sensor nodes. A sensor node, also known as a mote, is capable of performing some processing, gathering sensory information, and communicating with other connected nodes in the network. The history of development of sensor nodes dates back to 1998 in the Smartdust project [Pister, 1998]. One of the objectives of this project was to create autonomous sensing and communication in a cubic millimeter. Although this project ended soon after its initiation, it has created an intuition for further research projects; those include Berkeley NEST [Nest, 2001] and CENS [Cens, 2002]. Despite a variety of sensor nodes used since, there are some common aspects in terms of their architecture. These common structural aspects of modern wireless sensor nodes (motes) are described below.

2.2.1 Sensor node architecture

A wireless sensor node is composed of four basic components, which are a sensing unit, a processing unit (microcontroller), a transceiver unit and a power unit [TI, 1999].

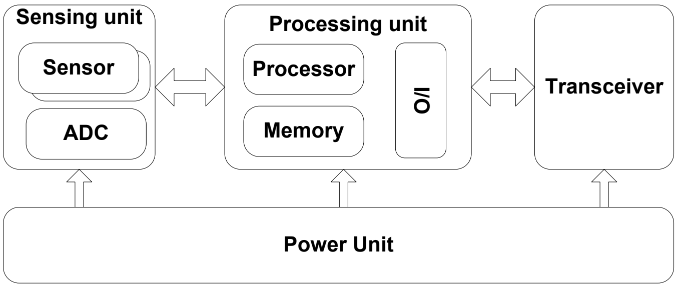


Figure 2.1: Architecture of a typical sensor node.

Figure 2.1 shows the typical construction of a sensor node. In addition to the above units, a wireless sensor node may include a number of application-specific components, for example, a location detection system or mobilizer. For this reason, many commercial sensor node products include expansion slots and support serial wired communication. Descriptions of the basic components are given below.

- ***Sensing Unit***. A sensor is a device that measures some physical quantity and converts it into a signal to be processed by the microcontroller. A wide range of sensor types exist including seismic, thermal, acoustic, visual, infrared and magnetic. Sensors may be passive (sensing without active manipulation of the environment) or active (using active manipulation/probing of the environment to sense data, e.g. radar) and may be directional or omni-directional. A wireless sensor node may include multiple sensors providing complementary data. The sensing of a physical quantity such as those described typically results in the production of a continuous analogue signal. For this reason, a sensing unit is typically composed of a number of sensors and an analogue to digital converter (ADC) which digitizes the signal.

- ***Microcontroller***. A microcontroller provides the processing power for, and coordinates the activity of, a wireless sensor node. Unlike the processing units associated with larger computers, a microcontroller integrates processing with some memory provision and I/O peripherals. Such integration reduces the need for additional hardware, wiring, energy and circuit board space. In addition to the memory provided by the microcontroller, it is not uncommon for a sensor node to include some external memory, for example, in the form of flash memory.

- ***Transceiver***. A transceiver unit allows the transmission and reception of data to other devices connecting a wireless sensor node to a network. Wireless sensor nodes typically communicate using an RF (radio frequency) transceiver and a wireless personal area network technology such as Bluetooth or the 802.15.4 compliant protocols ZigBee [CC2420, 2004] and MiWi [Cody Kenny et al., 2009]. The 802.15.4 standard specifies the physical layer and medium access control for low-rate, low-cost wireless communications, whilst protocols such as ZigBee and MiWi build upon this by developing the upper layers of the OSI Reference Model. The Bluetooth specification crosses all layers of the OSI Reference Model and is also designed for low-rate, low-cost wireless networking.

- ***Power Source***. All wireless sensor nodes must be supported by a power unit which is typically some form of storage (that is, a battery) but may be supported by power scavenging components (for example, solar cells). Energy from power scavenging techniques may only be stored in rechargeable (secondary) batteries and this can be a useful combination in wireless sensor node environments where maintenance operations like battery changing is impractical. To conserve energy, a power unit may additionally support power conservation techniques such as dynamic voltage scaling.

2.2.2 Some commercial sensor nodes

The capacity and capabilities of wireless sensor nodes are increasing day by day. Table 2.1 lists some of very commonly used sensor nodes over the last few years with their incremental growth in terms of CPU, memory and radio technologies. An overview of commonly used sensor network platforms, components, technology and related topics is available in the SNM - Sensor Network Museum [TIK, 2008], as well as in [SenSAR, 2010] and in [Senses, 2005].

These tiny memory and power constrained nodes are spread over a target area to monitor an event or to estimate a value. After deployment, the sensor nodes communicate with each other and construct a WSN. The following section describes the expected qualities of a WSN. These qualities are to be maintained while designing protocols for WSNs. This is termed as requirements of WSNs.

Table 2.1: Some commercial sensor nodes

	Rene (1999)	**Mica-2 (2002)**	**Tmote sky (2005)**	**Imote2 (2007)**
CPU	ATMEL 8535 8-bit, 4 MHz 36 μW sleep 60 μW active	Atmega 128L 8-bit, 8 MHz 36 μW sleep 60 μW active	TI MSP430 16-bit, 8 MHz 15 μW sleep 5.4 μW active	Intel PXA271 32-bit, 13.4 MHz 390 μW sleep 30 μW active
Memory	512B RAM 8KB Flash	4KB RAM 128KB Flash	10 KB RAM 48KB Flash	32 MB RAM 32MB Flash
Radio	RFM TR1000 10 kbps 2 μW sleep 12 μW receive 36 μW xmit 0.5 msec setup	CC1000 76 kbps 100 μW sleep 12 μW receive 75 μW xmit 2 msec setup	CC2420 250 kbps 60 μW sleep 63 μW receive 57 μW xmit 1 msec setup	

2.3 Requirements of WSNs

A well-designed sensor node is not the only requirement of a WSN needed to operate in an optimal way. Although a good sensor node is an important part of the whole wireless sensor network, a sensor node is just a very small part of the wireless sensor network. To build a good wireless sensor network and effective protocols for WSNs, various qualities of WSNs should be considered. Wireless sensor network requirements include the following:

- ***Large number of (mostly stationary) sensors.*** In most surveillance or monitoring cases, sensors are deployed densely in a large target field. Aside from the deployment of sensors on the ocean surface or the use of mobile, unmanned, robotic sensors in military operations, most nodes in a sensor network are stationary. Networks of 1,000 or even 10,000 nodes are envisioned, so scalability is a major issue.

- ***Low energy use.*** With only very limited exceptions, almost all sensor nodes of WSNs are operated by battery power. Since in many applications the sensor nodes are placed in remote/unattended areas, replacement of batteries is not possible. Thus battery power should be very cautiously consumed. In this kind of situations, the lifetime of a node is determined by the battery life, thereby requiring the minimization of energy expenditure.

- ***Network self-organization.*** Given the large number of nodes and their potential placement in hostile locations, it is essential that the network be able to self-organize. Manual configuration is not often feasible. Moreover, nodes may fail (either from lack of energy or from physical destruction), and new nodes may join the network (the user may deploy more sensor nodes in the network to increase the coverage). Therefore, the network must be able to periodically reconfigure itself so that it can continue to function. Individual nodes may become disconnected from the rest of the network, but a high degree of connectivity must be maintained.

- ***Collaborative signal processing.*** Yet another factor that distinguishes WSNs from other networks is that the end goal is detection/estimation of some events of interest, and not just communications. To improve the detection/estimation performance, it is often quite useful to fuse data from multiple sensors [Luo et al., 2006]. This data fusion requires the transmission of data and control messages, and so it may put constraints on the network architecture.

- ***Querying ability.*** A user may want to query an individual node or a group of nodes for information collected in the region. Depending on the amount of data fusion performed, it may not be feasible to transmit a large amount of the data across the network. Instead, various local sink nodes will collect the data from a given area and create summary messages. A query may be directed to the sink node nearest to the desired location.

2.4 Design Challenges and Constraints of WSNs

Considering the characteristics of WSNs described in section 2.2, and the requirements of WSNs described in section 2.3, the following major design challenges and constraints of WSNs can be identified.

- ***Limited energy***. Reducing node energy consumption is vital in WSNs. In fact, because of the reduced size of the sensor nodes, the battery has low capacity, and the available energy is very limited. Despite this scarcity of energy, the network is expected to operate for a relatively long time. Given that replacing/refilling batteries is usually impossible, one of the primary design goals is to use this limited amount of energy as efficiently as possible.

- ***Low-quality communications***. Wireless communications are always less reliable and have less quality compared to wired communication. Besides, sensor networks are often deployed in harsh environments, and sometimes they operate under extreme weather conditions. In these situations, the quality of the radio communication might be extremely poor, and performing the requested collective sensing task might become very difficult.

- ***Resource-constrained computation***. The resources are scarce in WSNs. Protocols for sensor networks must strive to provide the desired quality of service (*QoS*) in spite of the few available resources.

- ***Data processing***. Given the energy constraints, and the relatively poor communication quality, the data collected by the sensor node must be locally compressed and aggregated with similar data generated by the neighbouring nodes. This way, relatively few resources are used to communicate the data to the external observer.

- ***Scalability***. Scalability refers to the ability of the network to grow, in terms of the number of nodes, without excessive overhead. This is an important real-world requirement, where networks must support more than the small handful of nodes typical in a pilot implementation. This is due to the network overhead that comes with the increased size of the network. In ad hoc networks, the network is formed without any predetermined topology or shape. Therefore, any node wishing to communicate with other nodes should generate more packets than its data packets, i.e., control packets or network overhead. As the size of the network grows, more control packets will be needed to find and keep the routing paths. Moreover, as the network size increases, there is higher chance that communication links become broken in communication paths, which will end up creating more control packets. In a small network, the amount of control packets is almost negligible. However, when the network size starts increasing, the overhead increases rapidly. Since the available overall bandwidth is limited, the increase of overhead results in the decrease of usable bandwidth for data transmission. As the network size grows further, there is a very small amount of bandwidth left for application data transmission.

- ***Reliability***. Reliability is the ability of the network to ensure reliable data transmission in a state of continuous change of network structure. Typically, there is an inverse relationship between scalability and reliability in ad hoc wireless networks. As the number of nodes in the network increases, the more difficult it becomes to ensure reliability. This scalability characteristic of ad hoc networks described earlier imposes an interesting question about the reliability of the network. Since an ad hoc network is designed to automatically adapt itself to a changing environment or interference, it will issue more control packets when it faces dynamics. More dynamics in the environment will increase the number of control packets, and at some point, the network cannot sustain the amount of overhead caused by the dynamics, which will result in less reliability of data transmission. This breaking point will show up earlier in a large-sized network compared to a small or medium sized network. Thus, network scalability and reliability are closely coupled, and typically, they act against each other.

- ***Responsiveness***. Responsiveness is the ability of the network to quickly adapt itself to changes in topology. To achieve high responsiveness, an ad hoc network should issue and exchange more control packets, which will naturally result in less scalability and less reliability.

- ***Mobility***. Mobility refers to the ability of the network to handle mobile nodes and changeable data paths. Generally, a wireless sensor network that includes a number of mobile nodes should have high responsiveness to deal with the mobility. Therefore, it is not easy to design a large scale and highly mobile wireless sensor network.

- ***Power Efficiency***. Power efficiency, the ability of the network to operate at extremely low power levels, also plays an important role. A typical method for designing a low-power wireless sensor network is to reduce the duty cycle of each node. The drawback is that as the wireless sensor node stays longer in sleep mode to save power, there is less chance that the node can communicate with its neighbours. This decreases the network responsiveness, and also lower reliability due to the lack of the exchange of control packets and delays in packet delivery. In addition, a more complicated synchronization technique will be necessary to keep more nodes in low duty cycle, which may also affect scalability.

- ***Managing the design tradeoffs***. The complex issue of managing these tradeoffs comes down to how the communication overhead can be minimized while maintaining the network reliability and responsiveness. As explained earlier, there are many conflicting factors involved in the design of wireless sensor networks, and there are always design tradeoffs. When choosing a wireless sensor network for an application, careful consideration of the balance of these factors within the context of the needs of the application is critical.

2.5 System Model and Assumptions

In this section, the basic system model is described. This is a very familiar system model, and almost any application of event detection (ED) category or spatial process estimation (SPE) category (discussed in Chapter 1) can use this model. Moreover, the same model has been used by many researchers in designing various protocols for WSNs [Lindsey et al., 2002a; Muruganathan et al., 2005; Heinzelman et al., 2000; Tsai, 2007; Chen and Hu, 2010; Kim et al., 2010]. This system model has been used throughout this thesis.

Wireless sensor networks are very much application and system model dependent. Algorithms/protocols which are designed based on one system model usually do not produce the same results or show the same effectiveness when they are applied on another system model without modification. Thus it is important to define the system model before presenting any algorithm/protocol/architecture. The assumptions of the basic system model, which are used throughout the thesis, are described below:

- Assume a large-scale WSN. Large-scale WSNs consist of hundreds to thousands of nodes [Estrin et al., 2001; Kahn et al., 2000].

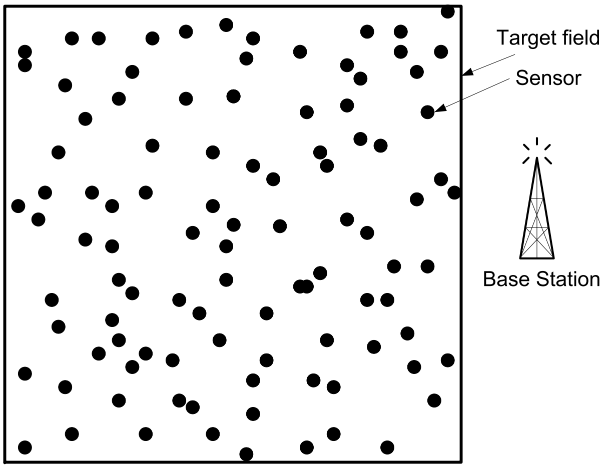


Figure 2.2: A system model of WSNs.

- Assume a target field, where a large number of wireless sensor nodes are deployed randomly. The deployed sensors sense the data and send it to the BS periodically. The BS is located outside the target field (see Figure 2.2).

- Both the BS and the sensors are stationary after deployment.

- All sensor nodes have limited battery power, and recharge of the batteries is impossible. Efficient energy-aware protocols are thus required for energy conservation.

- All the sensors are homogeneous. They have the same initial power and communication and processing capabilities.

- All the sensors have limited sensing range. However, the sensors have the ability to control the transmission range depending on the distance between a sensor node and its next-hop node.

- The radio channel is symmetric such that energy required for transmitting a message from a sensor node A to another sensor node B is the same as the energy required for transmitting a message from the sensor node B to the sensor node A.

- All sensor nodes sense the environment at a fixed rate, and thus always have data to send to the end user.

2.6 A Review of Evaluation Criteria

In order to establish a standard set of evaluation criteria, it is important to describe different metrics, which would be used to evaluate the performance of different topologies. Interestingly, it is found that the set of evaluation criteria and their definitions vary quite considerably. The following six sub-sections describe these metrics.

2.6.1 Evaluation measures related to energy usage

Not surprisingly, given the intrinsic constraints of WSNs, almost all evaluation strategies include some form of energy metric. Different metrics that relate to energy used by the researchers are listed below.

- The most common form of using energy for evaluating the efficiency is the overall energy consumption by the sensor network. This is measured by adding the total energy dissipation by each sensor in the network.

- Another important metric is energy distribution. This is a qualitative metric, rather than a quantitative one. This metric measures how evenly the energy dissipation is distributed. This is important for the sensor network to balance the energy consumption by the participating sensor nodes. If some sensors dissipate energy rapidly compared to other group of sensors, i.e., energy consumption is not evenly distributed, this adversely affects the system lifetime of the network.

- Average dissipated energy is an important metric. This is the ratio of total energy used per node to the number of events detected.

- A useful evaluation metric list includes 'Resource expended per packet delivered' metric. Here 'resource expended' refers to the numbers of connected pairs that are broken down because of nodes being depleted on their power. In other words, it is defined as the ratio between numbers of broken pairs to the total packets delivered.

- The metric 'packets before partition' is measured by the number of data packets sent and successfully delivered before network partition (partition due to node energy depletion).

2.6.2 Evaluation measures related to lifetime of network

The evaluation metric 'lifetime of the network' has a strong dependence on the nodes' battery capacity. As such, the network lifetime has been a critical concern in WSN research. While numerous energy-efficient protocols have been proposed to prolong the network lifetime, various definitions of network lifetime have also been used for the different scenarios and protocols. The lifetime of a sensor network is most commonly defined as the time to the first sensor node failure. However, this definition is seemingly over-pessimistic in many envisaged deployment scenarios, such as habitat monitoring, forest fire detection etc. [Kumar et al., 2005]. While other definitions exist, there has not been any consensus on which quantitative lifetime definition is most useful. The various 'network lifetime' definitions proposed and used in the literature include:

- Time till the first sensor node's failure [Chen and Zhao, 2005; Kang and Poovendran, 2005; Shi et al., 2006; Vass et al., 2005].
- Time till a certain percentage of sensor nodes die, or the number of surviving nodes in the network falls below a given application dependent threshold [Raicu et al., 2005; Cheng et al., 2008a].
- Time till the network becomes disjoint, and network partitions emerge [Pandana, 2005].
- Time till size of the largest connected component drops below a threshold [Blough and Santi, 2002].
- Time till the packet delivery rate falls below a certain threshold [Chen et al., 2002].
- Time till all the sensor nodes dies [Heinzelman et al., 2000].
- Time till the number of errors exceeds a threshold [Verdone et al., 2010].
- Time till the number of packets that can be (successfully/correctly) delivered by the network falls below a threshold [Dong, 2005].

2.6.3 Evaluation measures related to scalability

Scalability of a network means that the protocols running of a network perform well as the network grows larger or as the workload increases [Zhao and Raychaudhuri, 2009]. Scalability is an important factor in designing routing protocols for WSNs. A good routing protocol has to be scalable and adaptive to the changes in the network topology. Routing packets within a large scale WSN without storage overhead and routing table updates is a challenging problem. In turn, to constrain this communication overhead, routing in a sensor network demands efficient protocols for routing packets.

For large-scale WSNs, scalability is an important metric which measures how a protocol performs as node density varies, or the overall size of the network, or the number of data sources and sinks vary. Many researchers have tried to establish mathematical models for scalability. Using these mathematical models they put some numerical values against this qualitative metric to compare different protocols/architectures.

2.6.4 Evaluation measures related to overhead and efficiency

In conjunction with the direct measures of energy, the other metrics which are related to evaluating the performance are overheads and efficiency of the WSN. The following is a list of possible metrics.

- *Routing protocol message cost*. This is one of the most common metrics used for evaluating the efficiency of the protocols. It measures the number of packets generated by a protocol/algorithm for each successful communication.
- *Message Loss*. It measures the percentage of messages not received by any node in the network.
- *Control Overhead*. It measures the ratio between control and data messages being transmitted in the network. Some authors also use the overall packets sent or packets received, while others compare the application packet delivery rate with the routing packet rate.

- *Event Delivery Ratio*. This criterion is the ratio of the number of distinct event messages received by the sink to the number originally sent by the source. Some authors measure a related 'loss to collision' ratio.

- *Transmissions to query ratio*. This is the ratio between the total number of packets to the number of queries injected into the sensor network.

- *Average path or route length*. It measures the number of hops from source node to destination node. Although it is related to energy usage, each path or route can give very different results due to the non linear relationship between transmission power and range.

2.6.5 Temporal evaluation criteria

The primary temporal evaluation criteria used in the existing literature are the latency and the reaction time.

- *Latency*. This is measured by the average delay between transmitting an event message and receiving it at the sink. There have also been other measures used by many researchers to calculate the latency. One way is to calculate the total time elapsed to perform an action by the sensor nodes, for example, disseminate information to a set of nodes, or to complete a number of data collection rounds. Another way to calculate the latency is to calculate the time duration for identifying an event or to reach a consensus for a measured value.

- *Reaction Time*. The definition of this term varies among researchers, but essentially captures the average time it takes for the sink to receive data or particular messages after some change occurs in the network.

2.6.6 Other performance evaluation measures

The other various measures used in the existing literature of WSNs, which are concerned about deployment and design related issues, are mentioned below.

- *Storage requirement*. This is measured by the amount of memory required by an algorithm at each node.

- *Ease of deployment*. This metric is mentioned by some researchers in their technical papers but no specification of this metric has been found.

2.7 Descriptions of Different Topologies of WSNs

This section identifies and studies various types of topologies of WSNs. First of all, different topologies of WSNs are identified. In doing so, different application protocols proposed by various researchers, such as protocols for data gathering/collection, target tracking, routing, data aggregation, data dissemination etc. are studied. These protocols use various types of logical topologies. From each of the protocols, the topology is identified and listed in Table 2.2. The identified topologies are i) flat topology, ii) cluster based topology, iii) chain oriented topology, and iv) tree based topology.

The following sub-sections describe each of the topologies with their advantages and disadvantages in detail.

2.7.1 Flat/Unstructured topology

This is actually no topology, or the absence of any defined topology. In flat topology each sensor plays an equal role in network formation. Different protocols have been proposed based on flat/unstructured topology. For example, this flat topology has been used in data aggregation protocols [Xu et al., 2011], data gathering protocols [Fan et al., 2007], node scheduling protocols [Sausen et al., 2008], and routing protocols [Hussain and Islam, 2007]. Figure 2.3 shows the flat topology architecture where the nodes are the sensors and the edges are available communication links between two sensors.

All the protocols, while using flat topology, attempt to find good-quality routes from source nodes to sink nodes by some form of flooding. Flooding is a technique in which a given node broadcasts received data and control packets to the rest of the nodes in the network. This process repeats until the destination node is reached. Note that this technique does not take into account the energy constraints imposed by the WSNs. As a result, when used for data routing in WSNs, it leads to two problems, namely, implosion and overlap [Heinzelman et al., 1999]. Given that flooding is a blind technique, duplicate packets may keep circulating in the network, and hence

Table 2.2: Different protocols and their corresponding topology.

Protocol for	**References**	**Topology used in the protocol**
Data gathering	[Fan et al., 2007]	Flat
	[Ye et al., 2005]	Cluster based
	[Wu and Tseng, 2009; Weng et al., 2007; Li et al., 2006]	Tree based
	[Yuan et al., 2008; Tabassum et al., 2006]	Chain oriented
Target Tracking	[Olule et al., 2007]	Flat
	[Chen et al., 2004]	Cluster based
	[Tsai et al., 2007; Chen and Wang, 2009; Lin et al., 2006]	Tree based
	[Kang et al., 2007]	Chain oriented
Routing	[Lee et al., 2006; Heinzelman et al., 1999]	Flat
	[Al-Karaki and Kamal, 2004; Heinzelman et al., 2000; Liu et al., 2005; Manjeshwar and Agrawal, 2001]	Cluster based
	[Fariborzi and Moghavvemi, 2009; Karthickraja and Sumathy, 2010; Kim and Han, 2005; Chen et al., 2004]	Tree based
	[Yu and Wei, 2007; Chen et al., 2009]	Chain oriented
Data aggregation	[Xu et al., 2011]	Flat
	[Su and Bougiouklis, 2008; Jung et al., 2009]	Cluster based
	[Messina et al., 2007; Fan et al., 2008]	Tree based
	[Wu et al., 2009]	Chain oriented
Data dissemination	[Khan et al., 2008]	Flat
	[Wang et al., 2005]	Cluster based
	[Xing et al., 2009]	Tree based
	[Huang et al., 2009]	Chain oriented
Synchronization	[Beluch et al., 2010]	Flat
	[Long and Tao, 2006]	Cluster based
	[He, 2008; Li et al., 2006]	Tree based

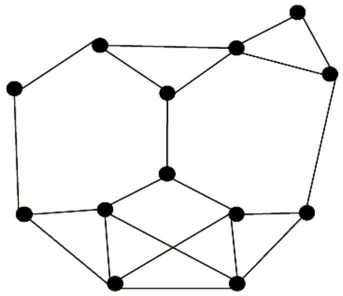


Figure 2.3: Flat topology architecture.

sensors will receive those duplicate packets, causing an implosion problem. In addition, when two sensors sense the same region and broadcast their sensed data at the same time, their neighbours will receive duplicated packets.

In a flat network, data aggregation is accomplished by data-centric routing where the BS usually transmits a query message to the sensor nodes via flooding, and the sensor nodes that have data matching in the query, will send response messages back to the BS. The sensor nodes communicate with the BS via multi-hop routes by using peer nodes as relays. The choice of particular communication protocol depends on the specific application.

Since flooding is a very costly operation in resource starved networks, smart routing algorithms restrict the flooding to localized regions. Some algorithms use probabilistic techniques based on certain heuristics to establish routing paths. Some examples of routing protocol based on flat topology are Sensor Protocols for Information via Negotiation (SPIN) [Heinzelman et al., 1999; Kulik et al., 2002], Directed-Diffusion [Intanagonwiwat, 2002], zonal rumor routing [Banka et al., 2005], Serial Directional Rumor Routing (SDRR) [Shokrzadeh et al., 2010] etc.

SPIN's meta-data negotiation solves the classic problems of flooding to some extent. SPIN is a three-stage protocol as sensor nodes use three types of messages, such as ADV, REQ, and DATA to communicate sensors with each other. ADV is used to advertise new data, REQ to request data, and DATA is the actual message itself. The protocol starts when a SPIN node obtains new data it is willing to share. It does so by broadcasting an ADV message containing metadata. If a neighbour is interested in the data, it sends a REQ message for the DATA and the DATA is sent to this neighbour node. The neighbour sensor node then repeats this process with its neighbours. As a result, the entire sensor area will receive a copy of the data. One of the advantages of SPIN is that topological changes are localized since each node need know only its single-hop neighbours. SPIN provides more energy savings than flooding, and metadata negotiation almost halves the redundant data. However, SPIN's data advertisement mechanism cannot guarantee delivery of data. To see this, consider the application of intrusion detection where data should be reliably reported over periodic intervals, and assume that nodes interested in the data are located 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.

Directed diffusion differs from SPIN in two aspects. First, directed diffusion issues data queries on demand as the BS sends queries to the sensor nodes by flooding some tasks. In SPIN, however, sensors advertise the availability of data, allowing interested nodes to query that data. Second, all communication in directed diffusion is neighbour to neighbour with each node having the capability to perform data aggregation and caching. Unlike SPIN, there is no need to maintain global network topology in directed diffusion. However, directed diffusion may not be applicable to applications (e.g., environmental monitoring) that require continuous data delivery to the BS. This is because the query-driven on-demand data model may not help in this regard. Moreover, matching data to queries might require some extra overhead at the sensor nodes.

Rumor routing performs well only when the number of events is small. For a large number of events, the cost of maintaining agents and event tables in each node becomes unfeasible if there is not enough interest in these events from the BS. Moreover, the overhead associated with rumor routing is controlled by different parameters used in the algorithm such as time to live (TTL) pertaining to queries and agents. Since the nodes become aware of events through the event agents, the heuristic for defining the route of an event agent highly affects the performance of next-hop selection in rumor routing [Messina et al., 2007].

Overall, the advantages of flat based protocols include good quality routes from source to sink and no topology maintenance overhead.

On the other hand, there are a few disadvantages:

- The main way of communication is by using flooding. However, flooding is an expensive operation that is normally avoided by WSNs routing protocols.
- A large number of redundant messages are created and passed. This redundancy consumes processing cycles, as well as bandwidth of the network. Latency is increased by the redundancy, because of the high contention of the wireless communication medium.
- Non-uniform energy distribution occurs in flat/unstructured topology. This is the reason why the lifetime of a sensor network decreases.
- Sensors are not aware of new members or members that die.
- Highly unreliable.
- High delay.

2.7.2 Cluster based topology

Cluster based topologies have widely been used in WSNs for various types of protocols, such as data gathering [Ye et al., 2005], target tracking [Chen et al., 2004], one-to-many, many-to-one, one-to-any, or one-to-all communications, routing [Al-Karaki and Kamal, 2004; Heinzelman et al., 2000; Yuan et al., 2008; Manjeshwar and Agrawal, 2001] etc. Clustering is particularly useful for applications that require scalability to hundreds or thousands of nodes. Scalability in this context implies the need for load balancing, efficient resource utilization, and data aggregation. Many routing protocols also use clustering to create a hierarchical structure and minimize the path cost when communicating with the BS.

Elements in a cluster

In general, when working with clusters it is possible to identify three main different elements in the WSN: sensor nodes (SNs), base station (BS) and cluster heads (CH) (see Figure 2.4). The SNs are the set of sensors present in the network, arranged to sense the environment and collect the data. The main task of a SN in a sensor field

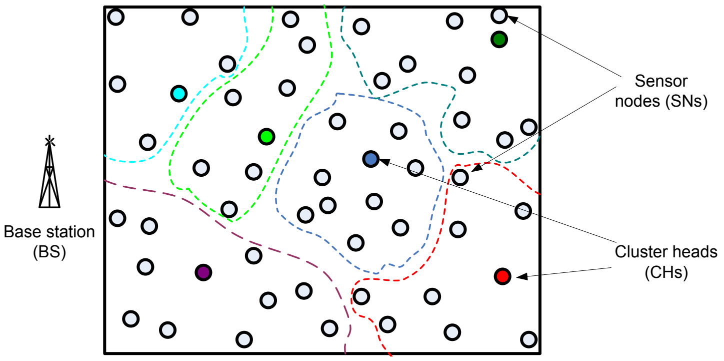


Figure 2.4: Cluster based topology architecture.

is to detect events, perform quick local data processing, and then transmit the data. The BS is the data processing point for the data received from the sensor nodes, and where the data is accessed by the end-user. It is generally considered fixed and at a far distance from the sensor nodes. The CH acts as a gateway between the SNs and the BS. The function of the cluster head is to perform common functions for all the nodes in the cluster, like aggregating the data before sending it to the BS. In some way, the CH is the sink for the cluster nodes, and the BS is the sink for the cluster heads. This structure is formed among the sensor nodes, the sink and the BS and is replicated as many times as it is needed, creating the different layers of the hierarchical WSN. The greatest constraint it has is the power consumption, which usually is caused when the sensor is observing its surroundings, and communicating (sending and receiving) data.

Cluster types

There exist many different ways to classify the clusters. Two of the most common classifications are homogeneous or heterogeneous clusters and static or dynamic clusters. The former classification is based on the characteristics and functionality of the sensors in the cluster, whereas the latter is based on the method used to form the cluster.

In homogeneous networks, all nodes have the same characteristics, hardware and processing capabilities. The role of cluster head is periodically rotated among the nodes to balance the load, to ensure that sensors consume energy more uniformly, and to try to avoid the black hole problem.

In heterogeneous sensor networks, there are generally two types of sensors:

(a) Sensors with higher processing capabilities and complex hardware, used generally to create some sort of backbone inside the WSN. They are designated as the cluster head nodes, and therefore have to serve as data collectors and processing centers for data gathered by other sensor nodes, and

(b) Participating sensors, with lower capabilities than the previous ones, used to actually sense the desired attributes in the field.

Static clusters are usually created when the network is formed of heterogeneous nodes and the network designers want to create the clusters around the more powerful nodes. In this case, the clusters are formed at the time of network deployment. The attributes of each cluster, such as the size of a cluster, the CH, the number of participating sensors and the area it covers, are static. Static clusters are easy to deploy, but their use is only appropriate for limited scenarios where the sensor field is predetermined, the targets to monitor are not in motion and it is easy to perform maintenance tasks (i.e. sensors replacements) in the network. On the other hand, dynamic cluster architectures make better use of the sensors. Sensors do not statically belong to a cluster, and may support different clusters at different times.

Clustering process

During the establishment of the cluster, it is necessary to take the following aspects into account: cluster size and form, selection process of the cluster heads, control mechanism of inter-cluster and intra-cluster collisions, and energy saving issues. The design of the clustering process is one of the important issues for the correct functioning of the WSN, due to the probed efficiency of using a hierarchical scheme for communication among the network elements.

In all the cluster-based protocols, three main phases can be identified during the clustering establishment process, namely (a) cluster head election phase, (b) cluster formation or set up phase, and (c) data transmission phase (steady-state phase). Different approaches exist to implement each one of these stages. For example, it is possible to use a fixed distribution of the SN and the CH, or to use a dynamic algorithm for the location of the sensors and the CH election. Clusters may be formed in any one of the following ways:

- *Probabilistic method*. The protocol LEACH [Heinzelman et al., 2000] uses this method where each sensor randomly picks a real number from 0 to 1. If the number is greater than a threshold value, the sensor declares it as a cluster leader and broadcasts invitation messages to all other sensors. A sensor, not picking a real number greater than threshold, joins any one of the leaders. Thus clusters are formed.
- *By election phase*. In this method all the sensors broadcast their information to all other sensors and form a knowledge base. Based on the local knowledge they form a cluster, and then select a leader.
- *Assigned by the BS*. In this method, clusters are formed by the BS. After deployment of the sensors, each node communicates with the BS and based on the information in the BS, it tries to form optimal nodes. Although this method can form optimal clusters, this method is rarely used because of the cost incurred for the communications of all sensors to the BS. An example protocol of this is the Base-station Controlled Dynamic Clustering Protocol (BCDCP) [Muruganathan et al., 2005].

Advantages and disadvantages

Cluster based routing protocols greatly increase the scalability of a sensor network. The overall energy consumption of the nodes compared to the flat topology protocols is reduced, leading to a prolonged network lifetime. The organization of the network into clusters lends itself to efficient data aggregation, which in turn results in better utilization of the channel bandwidth. Cluster-based routing holds good promise for many-to-one and one-to-many communication paradigms that are prevalent in sensor networks.

However, non-uniform clustering is the main problem for this topology. Considering the LEACH protocol, there is a fair chance that most of the cluster heads are situated in the same side of the network, whereas few cluster heads are on the other side, or even worsen, there is no cluster head in a specific area. Thus non-uniform clustering happens. Non-uniform clustering causes the following problems:

- The energy dissipation rate is highly different from one sensor to another sensor, even if they are in the same cluster. Thus energy distribution is not even.
- Total energy dissipation increases due to the long way communication between a cluster member and cluster head.
- Because of very long-way communications, some sensors consume energy more rapidly than other nodes, and die soon. As a result, the network lifetime decreases.
- Network connectedness may not be guaranteed.

2.7.3 Chain oriented topology

In this topology the protocols construct transmission chain(s) connecting the deployed sensor nodes to save energy dissipation of data transmission. A leader is selected in a chain which acts as the sink. All sensor nodes communicate with each other along the chain. A node sends data to the next node, which is called the successor node of the former node, towards the leader node. A successor node, receiving data from the predecessor node, forwards the data to its successor node towards the leader. In this fashion, all sensor nodes send their sensed data to the leader node(s). This method of communication facilitates the data aggregation [Mamun et al., 2010b].

PEGASIS (Power-Efficient Gathering in Sensor Information Systems) [Lindsey and Raghavendra, 2002] is an example protocol based on chain topology. In PEGASIS, every node in the chain senses the data, receives data from its predecessor, fuses with the received predecessor's data, and transmits to next node in the chain. Data aggregation performs in-network fusion of data packets, coming from different sensors en-route to the BS, in an attempt to minimize the number and size of data transmissions, and thus save sensor energies.

The basic idea of the PEGASIS protocol is that in order to extend network lifetime, nodes need only communicate with their closest neighbours, and they take turns in communicating with the BS. When the round of all nodes communicating with the BS

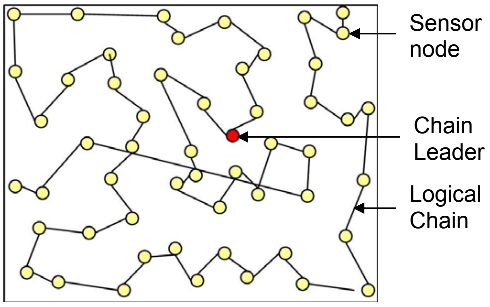


Figure 2.5: Architecture of chain oriented topology (used in PEGASIS).

ends, a new round starts, and so on. This reduces the power required to transmit data per round as the power draining is spread uniformly over all nodes. Hence, PEGASIS has two main objectives. First, to increase the lifetime of each node by using collaborative techniques. Second, to allow only local coordination between nodes that are close together so that the bandwidth consumed in communication is reduced. The chain construction is performed in a greedy fashion. Simulation results showed that PEGASIS is able to increase the lifetime of the network to twice that under the LEACH protocol. Such performance gain is achieved through the elimination of the overhead caused by dynamic cluster formation in LEACH, and decreasing the number of transmissions and reception by using data aggregation.

Although the clustering overhead is avoided, the protocol PEGASIS still requires dynamic topology adjustment. This is because, a sensor node needs to know about the energy status of its neighbours in order to know where to route its data. Such topology adjustment can introduce significant overheads, especially for highly utilized networks. Moreover, PEGASIS introduces excessive delay for distant nodes on the chain. In addition, the single leader can become a bottleneck. Finally, although in most scenarios sensors will be fixed or immobile as assumed in PEGASIS, some sensors may be allowed to move and hence affect the protocol functionality.

Figure 2.5 shows the chain oriented topology used in PEGASIS. In this figure, the circles represent sensor nodes whereas a bidirectional line between two nodes represents a successor-predecessor relationship.

Besides PEGASIS, there are also other protocols, such as COSEN and CHIRON, which use chain oriented topologies. COSEN is the first chain oriented topology which used multiple chains instead of a single chain. The advantages and disadvantages of chain oriented topologies are summarized below.

Chain oriented topologies save more energy than cluster based topologies do. For example, PEGASIS saves 50% more energy compared to LEACH. In addition, energy distribution in a chain oriented topology is more even than that of other topologies. Furthermore, because of better energy conservation, chain oriented topologies offer longer lifetime for WSNs.

On the ther hand, for a single-chain oriented topology takes too much time for data collection. Additionally, topology management overheads are high for a single-chain topology.

2.7.4 Tree based topology

In this topology all the deployed sensors construct a logical tree. Data are passed from a leaf node to its parent nodes. In turn, a receiver node the receiving data from the child node sends data to receiver's parent node after aggregating data with its own data. In this fashion, data flow from leaf nodes to the root node, which generally acts as the sink. The idea behind constructing a logical tree is that it avoids flooding and data can be sent using unicast instead of broadcast. In this way the topology can save energy. Figure 2.6 shows a typical formation of a logical tree. The arrows show the data flow from a leaf node to the root node/sink.

Tree topology is used to design various protocols for WSNs, such data collection scheme (TBDCS [Li et al., 2006]), routing protocols ([Woo et al., 2003; Park and Jung, 2007]), data dissemination protocols ([Messina et al., 2007; Fan et al., 2008]) etc.

One advantage of this topology is that it consumes less power than flat topology as flooding is not necessary for data communication. Further it can save bit more energy than some protocols based on cluster topology. Zhang and Yu [2010] prove that for data acquisition, tree based topology saves more energy than cluster based topology.

The disadvantages of tree based topology are as follows: i) the formation of tree is time consuming and costly, ii) the topology is not resilient to node failures, because if a parent node fails, then its entire sub-tree is cut off from the BS during the current epoch. iii) Uneven power consumptions occur across the network nodes. The nodes

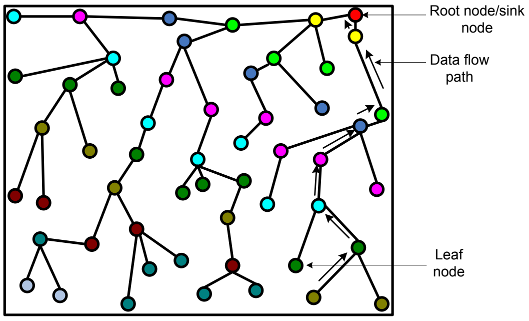


Figure 2.6: Tree based topology architecture.

nearer to the BS consume a lot of power in forwarding packets from all the nodes in their sub-tree, whereas the leaf nodes in the spanning tree do not have to perform any forwarding at all, and thus consume the least power, iv) latency is high for sending data from leaf to the root node, and v) tree maintenance overhead is very high.

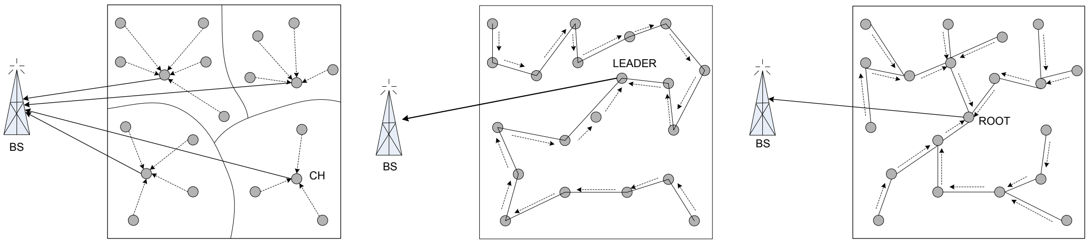
2.8 Comparison of Different Topologies

This section compares the different topologies introduced above, namely flat, cluster based, chain oriented and tree based topologies. They will be compared using the performance metrics that is described in section 2.6.

2.8.1 Topology comparison based on energy efficiency

Energy efficiency is the most important constraint and performance metric for WSNs due to the limited energy resources of the sensor nodes, and their operations in unattended and inaccessible environments where replacement of energy resources might be impossible. Therefore, while traditional networks aim at achieving high Quality-of-Service (*QoS*)provisions, WSNs focus primarily on energy awareness in every aspect of hardware and software design and operations to prolong the useful lifetime of each sensor node and, more importantly, of the entire WSN.

Communication is the most energy intensive activity performed by the sensor nodes, and hence the WSN topology and communication protocols can play a significant role in the energy efficiency and lifetime of the WSN. Figure 2.7 depicts the communication patterns of basic cluster, chain and tree topologies. The energy required



(a) Cluster topology (b) Chain topology (c) Tree topology

Figure 2.7: Communication patterns for different topologies of WSNs.

for communication scales with distance between two nodes (d) from d^2 to d^4. Since the radio signal attenuation scales with distance in a greater-than-linear fashion, the multi-hop communication in chain topology consumes less power than the single-hop long distance radio communication in cluster topology [Chandrakasan et al., 2002]. In chain oriented topologies, chains are usually formed considering the minimum distance from a node in the chain to its successor. On the other hand, while configuring clusters, distance has never been used as a selection criterion. Simulation results show that summation of d^2 values is the minimum for a chain oriented topology compared to cluster or tree based topologies. As the total energy consumption is directly proportional to the d^2 / d^4, total energy consumption for chain oriented topology is always lower than other topologies. For example, PEGASIS spends only 70% of total energy spent by LEACH for 300 rounds of data collection [Lindsey and Raghavendra, 2002].

Energy consumption of the sensor nodes of a WSN should be evenly distributed. If some nodes spend too much energy to perform a task, by repetition of that task, those nodes would lose their energy rapidly and die soon. It is apparent from Figure 2.7 that, in cluster and tree topologies, cluster heads (cluster topology) and parent nodes (tree topology) handle more traffic than the leader node(s) of chain topology. As a result, those nodes in cluster and tree topologies deplete their energy faster than other nodes, and thereby disconnecting the BS from the whole WSN, which might still have adequate resources and infrastructure. This is the well-known self-induced black hole effect. Simulations show that nodes closest to the BS are the ones to die out first for flat mesh routing, whereas nodes farthest from the BS are the ones to die out first for direct transmission [Heinzelman et al., 2000].

In-network processing is one of the key mechanisms to improve the energy efficiency of WSNs. Simulations have shown that it typically requires around 100 to 1000 times more energy to transmit a bit than to execute an instruction [Schurgers et al., 2002]. Possibilities for in-network processing in WSN include aggregation and compression, which exploit spatial and temporal correlation in the sensed data, for performing local compression to reduce global communication to BS by reducing the overhead of packet headers, by compressing the payload, and by reducing the probability of packet collisions. For example, the predominant communication in WSN is converge-cast, i.e., collection of sensed data from multiple sensors at the BS: a kind of reverse multicast. In addition, coverage-cast aligns closely with the need and capacity of WSN to perform in-network processing.

In a flat topology, routing paths are not fixed a priori, and hence the opportunity for the in-network processing is very much limited. In a cluster based topology, where the cluster members reach the cluster head in a single hop, only the cluster heads can be used for data aggregation/pre-processing. For a tree based topology, sensor nodes get more opportunity to aggregate/pre-process data. For example, a parent node can process the data received from its child node(s). Finally, chain oriented architecture is inherently amenable to in-network processing. In this topology each single node can be used for in-network processing of its predecessor's data. It decreases communication traffic and communication frequency via data aggregation progressively at each leader in the chain by processing and filtering the possibly redundant data received from other chain members. Unlike the cluster topology, the leader of a chain is not responsible for all data aggregation. Every member of a chain participates in the process of data aggregation. This actually balances the load among the nodes of a chain and thus energy consumption is evenly distributed. This is one of the important reasons why chain oriented topology offers longer lifetime of WSNs.

In summary, chain oriented topology performs best with regard to energy efficiency. On the other hand, flat topology is the least energy efficient topology. Cluster based topology is ahead of tree based topology in respect of energy efficiency. However, the energy efficiency of cluster based topology primarily depends on the cluster formation algorithm.

2.8.2 Topology comparison based on reliability

Reliability analysis is an important task for the successful operation of WSNs. IEEE P1451.5 web survey [WSWG, 2002] identified data reliability as one of the most important parameters in the design of WSNs. Reliability is generally defined as the probability that the system will perform its intended function under stated conditions for a specified period of time [Rausand and Hoyland, 2004].

The WSN reliability can be studied for three different scopes of data delivery [Abo-El-Fotoh et al., 2006], collectively known as the infrastructure communication: a) users send their interest to a single sensor node, b) users send their interest to a subset of nodes in a sub-area and the message needs to be delivered to all sensors in the particular group, and c) users send their interest to the entire sensor network and the message needs to be delivered to all sensors in the network. There exists another scope of message delivery, known as application communication, in which it is sufficient that the message from sink is reliably delivered to only a group of sensor nodes that together cover the entire sensor field or the intended area of observation [Tilak et al., 2002]. This is different from the delivery to all sensors in the infrastructure communication due to the typical redundant deployment of sensors.

Given a single node failure, flat topology reduces the chance of the entire network failure, because the failure of any node results only in the localized failure, leaving the rest of the system unaffected. However, when a node becomes obstructed, there is no alternate path from the associated node to the BS. A flat topology is highly fault tolerant as it offers multiple redundant paths throughout the network. If a routing node fails, or the link between nodes becomes unavailable, the network can automatically reconfigure itself around the failed component. In a WSN, the degree of redundancy, and in general the reliability of the network, is essentially a function of node density. A WSN with flat topology can be deliberately over-provisioned for reliability simply by adding extra nodes. The addition of redundant nodes also improves the reachability of WSN by providing multi-hop routes to inaccessible or hidden nodes. Also if certain environmental or architectural conditions result in poor reliability, it is difficult or impossible to adapt a point-to-point network like the network with a cluster topology to increase reliability. In contrast, the WSN reliability can be improved by redeploying redundant nodes in the affected area.

Tree topology has the lowest reliability due to the use of only a single direct link between nodes at successive levels in the hierarchy. For the same hierarchy level, chain oriented topology offers better reliability than tree based topologies. Clustered hierarchical topology is a compromise between the two extremes. It is better than tree/chain oriented topology, as it maintains multi-hop paths, while it has lower reliability than flat topology because each communication between nodes at different clusters must route through affiliated cluster heads. This is to note that, in WSNs, the residual energy of a node affects the reliability in an indirect way. For example, if the energy of the cluster head goes down, then the reliability decreases in an exponential manner. Moreover, energy simultaneously affects the number of normal and critical faults. As the energy decreases, the number of faults increases [Moraes et al., 2009]. As energy efficiency of chain oriented topology is higher than that of any other topology, it is possible that energy efficiency of chain oriented topology compensates the relatively weak reliabilities.

2.8.3 Topology comparison based on scalability and self-organization

WSNs should be scalable to varying sensor density, and should maintain performance that is independent of the number of nodes or gracefully degrade the performance depending on the number of surviving nodes. Also WSNs will presumably be required to self-configure into connected networks, and will require different or at least adaptive protocols. For example, by allowing the algorithms and protocols to trade off accuracy and latency with energy dissipation, WSN can be scalable and flexible to the application requirements that might change over the WSN lifetime.

Self organization helps in maximizing the network lifetime. Nevertheless, self-organization should be kept in perspective with energy cost and speed. Sometimes letting a WSN kill its nodes may be more energy efficient than trying to revive it [Lai et al., 2009]. Self-configuration time involves fault identification, fault localization and fault recovery phases. Self-configuration time for tree topology can be quite high, while that for cluster and chain oriented topology take the less time compared to the time required for tree based topology. In the chain oriented and clustered hierarchical approaches, the chain leader and the CH respectively can initiate the localized reconfiguration of the chains and clusters.

In flat topology, a high number of sensor nodes increases load on the BS, which results in increased power consumption and complexity. Also, as node density increases, the increase in collisions greatly degrades performance. Further, not all nodes have enough transmission range or the line-of-sight communication with the BS. It is difficult to scale flat WSNs to more than a few nodes.

For tree based topologies, self-configuration and scalability are limited up to a certain number of depths of the tree. After that, WSNs designers should carefully plan the transmission and duty cycle scheduling of the sensor nodes to avoid the aforementioned negative effects of dense deployment. Therefore, in practice, tree based topology works well for medium sized networks, but has scalability limitations that degrade performance for larger or densely deployed WSNs.

The scalability and self-organization issues of chain oriented topology primarily depend on the number of chains in the network. Chain oriented topology with a single chain (PEGASIS) has the same limitations that tree based topologies have. Multiple chain oriented topology and clustered hierarchical topologies improve the scalability of the flat networks by assigning leaders / cluster heads to manage the local neighbourhood of sensor nodes. For example, LEACH and COSEN use localized coordinations to enable scalability and robustness for dynamic networks [Messina et al., 2007; Fan et al., 2008]. Furthermore, the adaptive self-organizing capabilities of multi chain and clustered hierarchical WSNs allow the periodic reformation of hierarchical chain / clusters of sensor nodes in the event of environmental or topological changes as sensor nodes fail or new sensor nodes are added to improve connectivity and coverage.

In addition, for scalability, the addressing structures of WSNs are likely to be quite different; for example, geographic, data-centric, or address-free structures. Distributed and/or probabilistic assignments of addresses are only unique in a two hop neighbourhood. For example, address-free architecture [Elson and Estrin, 2001], which leverages the spatial and temporal locality of WSNs to assign probabilistically unique identifiers for each new transaction, must only scale with the transaction density of WSNs, while a statistically assigned global address space must scale with the total number of nodes in the WSNs. Hierarchical tree and hierarchical clustered architectures are inherently amenable to a scalable addressing structure where the nodes are addressed based on their position in the hierarchy. For example, a node z that is a member of level-1 cluster y and level-2 cluster x could have an address x.y.z [Belding Royer, 2003]. Additionally, this scheme allows simple routing protocols with small foot-prints that are scalable and occupy small memory space.

2.8.4 Topology comparison based on data latency

The WSNs traffic, which is characterized by the interaction with the environment or generated in response to certain events, is likely to be very different from human-driven forms of networks. A typical consequence is that WSNs are expected to exhibit very low data rates over a large time scale, but can have bursts of traffic on the occurrence of certain events.

A single-hop-to-sink structure has the least data latency, because there is no delay due to buffering at routers along the path. However, this structure is not scalable, and there may be more loss due to collisions as the network density increases. Flat topology networks have higher data latency than the single-hop-to-sink structure but lower data loss, because keeping the transmission power lower reduces the packet collision rates. Depending on the number of nodes and the distance between them, a flat topology network may endure increased latency as a message moves along a multi-hop route to the BS. In addition, a flat topology network can cause the nodes which are closer to the BS to overload with the increased node density. Such overload causes a high latency in communication, and in the worst case, creates a black hole of overloaded (or dead) nodes around the BS.

In hierarchical tree topology, as the data moves from the lower level to a higher level, it moves a greater distance, thus reducing the travel time and data latency. However, as the distance betwen cluster levels increases, the energy dissipation, which is proportional to the square of distance, increases. Lindsey and Raghavendra [2002] propose a metric by (energy×delay), and present a chain oriented scheme that attempts to balance the energy and delay cost for data gathering from WSNs.

Clustering is a design approach to minimize energy consumption, and to minimize data latency. In clustered hierarchical topology, only CH (along the hierarchy) performs aggregation, whereas in chain topology, each intermediate node performs data aggregation. As a result, clustered hierarchical architecture has lower latency than chain topology. Nevertheless, individual packet latency may not be an important criterion due to the inherent redundancy (caused by spatial and temporal correlation in the sensed data) in the transmitted packets.

2.8.5 Topology comparison based on overhead and efficiency

Flat topology produces the maximum number of packets for routing. In a flat topology, because of flooding, a node can receive multiple copies of the same data from different nodes. On the other hand, in cluster based topology, the cluster heads receive data from all members of the cluster. In tree topology, a parent node receives data from its children node(s). But in a chain oriented topology, a node in a chain receives data from only one node. If a sensor node always receives data from a single node instead of multiple nodes, the sensor node's communication overhead is reduced by-

i) reduced information flow

ii) reduced processing time for example, decoding the sources

iii) reduced queuing time and space requirements for buffering

Typically, communication overhead is defined as the ratio between the number of control bits and number of bits in a data packet. The control bits convey different information, such as where the information was originated and where it is being sent to, or any other information that is not actually the payload. If a node always receives data from a single node and sends its data to another node, less number of control bits are required for these control bits (for example, addressing a large group requires higher number of bits while addressing a small group requires lower number of address bits). Thus, a single sender/receiver reduces communication overhead.Thus, in terms of communication overhead, chain oriented topology performs better than any other topologies. Tree based topology performs better than cluster based topology, which in turns, performs better than flat topology.

In terms of topology management overhead, flat topology is the best, because flat topology does not need to maintain any structure. Thus this topology does not need to disseminate topology control messages. Tree based topology, on the other hand, has the largest number of control message overhead to maintain the tree structure. Cluster based and chain oriented topologies also have control message overheads, but much less than that of tree based topology.

2.8.6 Topology comparisons at a glance

Table 2.3 summarizes the comparative analysis of the four topologies. The topologies are compared using ten performance metrics. Each topology is marked out of 4 for each evaluation metric according to their performance in regard of the corresponding metric. Finally, the points for each evaluation metrics of each topology are added. The totals, which indicate the overall performance, for each topology are shown at the bottom row of the table.

In the point system used in Table 2.3, 4 means excellent, 3 means good, 2 means fair, and 1 means poor. Depending on the design structure of a topology, some fields of the table may have a point presented as x to y, where x is the minimum point, and y is the maximum point. For example, with regard to latency, single chain oriented topology shows fair results (thus receiving 2 out of 4), whereas multi-chain topology shows excellent results (thus receiving 4 out of 4). These points used here are entirely relative. For example, with respect to energy consumption, cluster based topology is better than flat topology, but not as efficient as chain oriented topology. Note that this point system is used for the purpose of easy understanding; this does not provide a standardized measure for comparison.

Table 2.3 shows that chain oriented topology scores the highest among four topologies, whereas flat topology scores the lowest. Cluster based topology performs better than tree based topology. However, this topology is not as efficient compared to chain oriented topology. Nevertheless, there are some areas in chain oriented topologies, such as energy distribution, scalability, latency etc. where special attention should be paid by the designers to make the topology more efficient. Furthermore, chain construction can be made more energy-efficient.

From the above discussions, it is found that chain oriented topology performs better than any other topology, and still there are many scopes to make this topology even better. For this reason, this thesis chooses chain oriented topology, and aims to construct an efficient chain oriented logical topology.

Table 2.3: Comparison of different topologies

Evaluation metric	**Flat**	**Cluster based**	**Tree based**	**Chain oriented**
Total energy consumption	1	3	2	4
Energy Distribution	1	2	2	3
Load distribution	3	3	3	4
Redundant Communication	1	4	4	4
Data reliability	4	3	3	2 to 3
Scalability	2	4	3	2 to 4
Latency	4	3	3	2 to 4
Network Connectedness	1	3	3	3
Lifetime	2	3	3	4
Topology management overhead	4	3	2	4
Overall Scores (Out of 40)	**23**	**31**	**28**	**32 to 37**

2.9 Summary

In this chapter, different topologies, which are used for designing different protocols by the researchers, are identified. The topologies, namely flat, cluster based, chain oriented, and tree based topologies, are discussed in detail. This chapter also discusses different performance metrics of WSN topologies. Defining a system model, all topologies are compared against each other using these performance evaluation metrics. From the discussion of this chapter, this thesis argues that chain oriented topology is the most promising topology among all topologies described in this chapter. Moreover, there are some provisions to make the chain oriented topology perform even better. As a result, chain oriented topology is chosen as the target topology for this thesis. In the next chapter, a model of chain oriented topology is proposed.

Chapter 3

Multi-Chain Oriented Logical Topology

3.1 Preamble

In Chapter 1, it is argued that the constraint minimizing problems of WSNs should be addressed from the topological point of view. Additionally, Chapter 2 discusses the potentiality of the chain oriented topology as a candidate topology in this regard. In this progression, this chapter proposes a variant of chain oriented logical topology. The main aim of this study is to design a logical topology, so that the proposed topology retains the advantages of the chain oriented topologies, and at the same time, overcomes the problems of the chain oriented topology.

Chain oriented topology facilitates the minimizing of different constraints of WSNs in many ways. For example, energy consumptions by the sensor nodes can be greatly reduced by the chain oriented topology [Pham et al., 2004; Shin and Suh, 2008; Satapathy and Sarma, 2006]. For data fusion/aggregation, chain oriented topology offers substantial advantages by the logical structure of the sensor nodes [Luo et al., 2011; Wu et al., 2009]. It is also possible to obtain collision-free transmissions using a chain-oriented topology [Yoo and Kim, 2007]. Other WSNs requirements, such as connectivity, robustness, scalability, responsiveness, and reliability can also be enhanced by the chain oriented topology.

To achieve the above mentioned facilities, careful designing of chain oriented topology is essential. Designing a logical topology for WSN needs to be considered from different perspectives, namely i) Resource oriented considerations, such as energy consumption and time requirement, ii) Networking related considerations, such as connectivity, robustness, and reliability, iii) Data centric considerations, such as data collection strategy, data aggregation facility, iv) Architecture oriented considerations, such as scalability, task orientation, and light weighting, and v) Network management considerations, such as fault detection, performance management etc. All these aspects are taken care of in designing the proposed logical topology.

The rest of the chapter is organized as follows. Section 3.2 discusses the considerations for topology design. Section 3.3 then describes the existing chain oriented topologies, and discusses the observations of different chain oriented topologies. Section 3.4 presents the detailed description of the proposed topology. This section also describes different terminologies, and their definitions. Furthermore, this section provides discussion of different topology designing issues, the workflow and the communication abstraction of the proposed topology. Section 3.5 presents the discussion regarding the network management issue for the proposed topology. Section 3.6 evaluates the performance of the proposed topology. Finally, the summary of this chapter is provided in Section 3.7.

3.2 Considerations for Topology Design

This section provides detailed descriptions of different design aspects, which are considered in constructing the proposed logical topology.

3.2.1 Hierarchical structure

The first issue to consider in designing the proposed logical topology is the structure of the topology, i.e., whether the topology should be hierarchical-structured or not. A hierarchical structure has many advantages over a non-hierarchical structure. For example, a hierarchical network structure can reduce the length of time for transmitting messages between two very far nodes in a sensor network. The structure requires that some capable sensors act as local leaders/cluster heads to interface with the outside world. Additionally, the grouping/clustering of sensors can also aggregate and process data locally to reduce communication load in the network. However, this solution may not be energy efficient. It is well known that, given two nodes, the radio transmission power required at the transmitter end is exponentially proportional to the distance from the receiver [Rappaport, 2002]. For hierarchical structures, leaders/cluster heads need to use exponentially more power to relay messages because they decrease the number of intermediate nodes, and consequently have to deal with longer distances. Although a hierarchical structure is not energy efficient theoretically, it is an advantageous choice for a large-scaled dense sensor networks for several reasons. The exponential effect is not significant as the distances within a dense environment are limited. Furthermore, by careful rotation of leaders/cluster heads, a balanced energy dissipation state can be achieved, where some sensors can afford to consume more energy. In contrast, multi-hop communication without a hierarchical structure consumes energy among all participant sensors in an unplanned way, which results in faster energy exhaustion of sensors with lower energy capacity. With local leaders/cluster heads taking more responsibility, energy can be saved for energy-constrained sensors, which extends the lifetime of the overall network. For this reason, hierarchical structure is chosen for the proposed topology.

3.2.2 Resource oriented considerations

Designing topology for resource-constrained sensor network requires careful consideration about the consumption of resources, which include energy, time, processing capability, memory requirements etc. Obviously, the first consideration should be the energy. It is shown in Chapter 2 that chain oriented topology greatly reduces energy consumption. This section, therefore, does not repeat comparing different topologies with respect to the consumption of energy or other resources. However, this section discusses and identifies various scopes, using which resource utilization of chain oriented topology can further be improved.

The aim of the resource oriented consideration is not only to save energy, but also to ensure that energy dissipation is evenly distributed. As the chain leaders undertake more tasks and long distant communications, they deplete energy more rapidly compared to other nodes in the network. Thus, it is important to change the role of the leader often, so that the load of leader is distributed among many nodes. On the other hand, very frequent changing of the role of chain leaders actually diminishes the performance of chain oriented sensor networks. At the same time, energy consumption also increases, because of the increment of control message passing regarding the new leaders, and their selection procedures. Thus, determining the time to change the role of leader is very crucial.

Another important resource related issue is the *time* required by the network to perform an operational round. If the sensors of the network construct only one chain, latency becomes very high. This is because each sensor node needs to wait for the data from its predecessor node. This latency can be reduced by constructing multiple chains using the sensor nodes. Multiple chains in the network introduce parallelism to a certain extent. This phenomenon directly recommends the use of multiple chains, instead of a single chain. Multiple chains are advantageous compared to a single chain not only for decreasing the latency, but also for receiving other facilities, such as scalability, flexibility, and ease for management. For these reasons, the proposed topology uses multiple chains instead of a single chain.

3.2.3 Networking related considerations

Connectivity, robustness, and reliability are the most import issues from networking related considerations. In sensor networks, one of the main concerns is that sensor nodes can die anytime, and because of wirelessness, the probability of missing a message is high. The topology should take care of the communication model whenever a sensor node dies or is not responding.

Connectivity, robustness, and reliability are directly related to the distance between two nodes which communicate wirelessly. For a pair of nodes with a short distance between them, higher values of connectivity, robustness, and reliability are achievable than in a pair of nodes having a longer distance. This motivates to select the closest node as a neighbouring node along the chain. However, adopting the greedy method of choosing the nearest neighbour always results in producing few longer links at the end of the chain formation phase. On the other hand, brute-force search for searching neighbour nodes are not suitable for WSNs, because of the scarcity of processing power and the memory. Thus, in designing the chains for the proposed topology, the emphasis is given to keeping the chains shorter, as well as maintaining lower time complexity and memory complexity of the algorithm.

3.2.4 Data oriented considerations

WSNs are very much data oriented. Usually WSNs are deployed to collect environmental/monitoring data. Thus, data related considerations during the designing of the topology are very crucial. The two most important issues of data related considerations are data collection, and data aggregation. These issues are considered in designing the proposed topology, and the discussion is provided below.

Data collection. According to the system model, based on which the multi-chain oriented logical topology is proposed, sensed data are continuously/periodically collected at all of the sensor nodes, and forwarded through wireless communications to a central BS for further processing. Sensor data collection requires that all sensing data are correctly and accurately collected and forwarded to the BS. This is because, sometimes, the processing of the data needs the global knowledge, and is much more complex. This feature thus prevents using data aggregation/fusion techniques which are usually used to enhance the network performance. As a result, the major traffic in sensor data collection is the reported data from each sensor to the BS. Such a "many-to-one" traffic pattern, if not carefully handled, causes high unbalanced and inefficient energy consumption in the whole network. For example, the energy hole problem is reported and discussed in [Stallings, 1999], where sensor nodes close to the BS are depleted quickly due to traffic relays and create a hole shape area that leaves the remaining network disconnected from the BS.

Figure 3.1 depicts an example of such a scenario. One possible solution to alleviate the issue of uneven energy dissipation is avoiding construction of complex chains, where two or more nodes send their data to a single node. Another way is to exclude the set of sensor nodes from doing the same task repeatedly. In the proposed logical topology, these matters are taken into consideration.

Data aggregation. Besides considering the data collecting technique, another important issue to consider is data aggregation. The information gathered in a sensor network is highly correlated, due to a spatial and temporal correlation between successive measurements. Exploiting the data-centricity and the spatial-temporal correlation characteristics allows the application of effective in-network data aggrega-

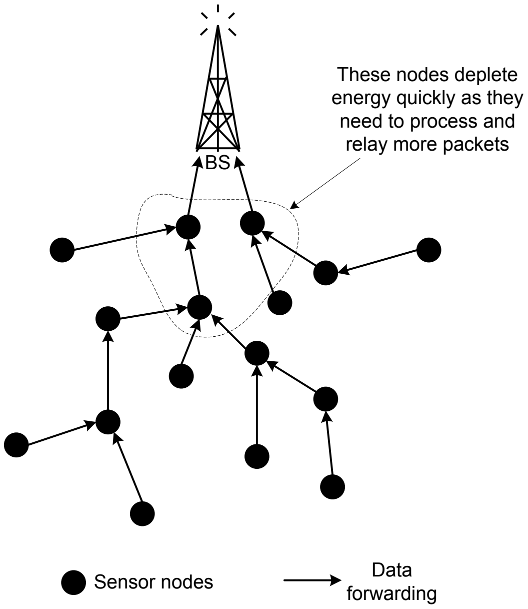

These nodes deplete energy quickly as they need to process and relay more packets
BS
Sensor nodes
Data forwarding

Figure 3.1: Uneven energy dissipation by sensor nodes.

tion techniques, which further improve the energy-efficiency of the communication in WSNs [Xibei et al., 2010]. Aggregation can eliminate the inherent redundancy of the raw data collected and, additionally, it diminishes the traffic in the network thereby reducing congestion and induced collisions [Macedo, 2009]. Thus, data aggregation policies are adopted in WSNs to increase the lifetime of the network. However, designing aggregation points for data aggregation needs careful attention. Data aggregating points consume more energy in processing the aggregation method, and an unplanned, non-distributed aggregation points can drastically affect the lifetime of the network [Chen et al., 2006]. In designing the logical topology and its communication model, these data aggregation related issues are considered.

3.2.5 Architecture oriented considerations

The architecture of wireless sensor networks needs to accommodate the following three characteristics:

Scalability. Large-scale wireless sensor networks rely on thousands of tiny sensors to observe and influence the real world [Akyildiz et al., 2002]. These sensors do not necessarily need to be active at all times, so sensors can be dynamically added to or removed from the network [Tian and Georganas, 2002]. A durable and scalable architecture would allow responses to changes in the topology with a minimum of update messages being transmitted. Another important feature of chain oriented WSNs that affects scalability is the number of nodes in a chain. If there is a single chain in the whole network, the topology is subject to poor scalability. On the other hand, multiple chains in the network can solve the scalability problem. However, all the chains in the network should be of similar length. Therefore, in designing the proposed topology, the lengths of multiple chains are kept similar.

Task Orientation. The sensor networks are always correlative with tasks at the current stage. The tasks of wireless sensor networks range from the simplest data capturing and static-nodes to the most difficult data collecting, mobile-node sensor network [Chong and Kumar, 2003; Akyildiz et al., 2002]. The sensor networks for different tasks behave totally differently sometimes. The software structure should be reasonably optimized and tailored, according to a predefined task-set of each node, to be adapted to this distinction. Thus, the proposed topology divides all the deployed nodes in the hierarchical structure, assigns specific tasks to each node, and gives a communication abstraction, through the use of which other protocols can be designed.

Light Weighting. The computing and storage capabilities of sensor nodes are very limited. Lightweight operations, such as data aggregation, reduced message size, and a piggyback acknowledgment mechanism, must be applied to the architecture. In designing the communication abstraction of the proposed logical topology, this notion is considered.

3.2.6 Network management considerations

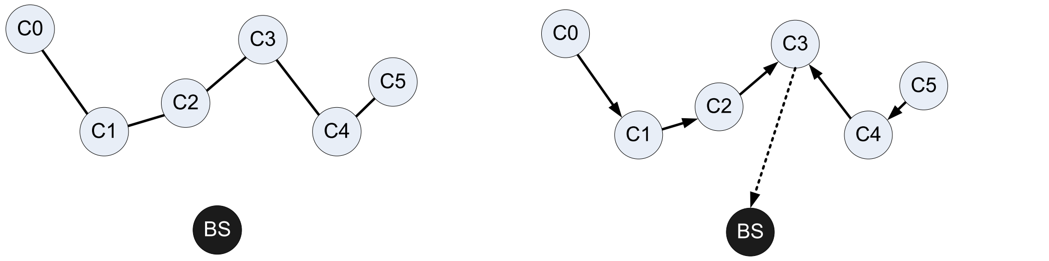
Large-scale wireless sensor networks are composed of hundreds or thousands of sensor nodes. For this reason, effective management of WSNs is a big challenge. Network management includes fault management, configuration management, security management, performance management, and accounting management [Stallings, 1999]. In particular, most wireless sensor nodes are powered by battery rather than external power, so that energy conservation is a key issue for the design and implementation of wireless sensor networks. Consequently, energy management becomes a special and important aspect of wireless sensor network management.

Effective management requires a practical architecture that is optimized to the features of wireless sensor networks and satisfies the requirements of wireless network management protocol. Therefore, the logical topology is built in such a way that it can be used as the underlying architecture by the network management scheme. Once the architecture of the network management scheme is constructed, various issues of network management scheme, such as primitives, functionalities, Management Information Base (MIB) etc. can be designed easily. Since both the logical topology and the network management scheme use the same architecture, this phenomena can assist to assess a system's resource requirement, response time, and performance patterns and anti-patterns with the help of a performance model [Smith and Williams, 2003].

3.3 Different Existing Chain Oriented Topologies

Chain oriented topologies have been used by the researchers in designing various protocols, among which data broadcasting protocols, data collection/gathering protocols and routing protocols are the major instances. Chain topologies are mainly used in these protocols to reduce the total energy consumption, and thus to increase the lifetime of the network. This section discusses different protocols, which use chain oriented topologies.

Lindsey and Raghavendra present several chain oriented data broadcasting and data collection/gathering protocols for sensor networks [Lindsey et al., 2001; Lindsey and Raghavendra, 2002]. They investigate broadcast problems in sensor networks and adopte a chain oriented approach for situation awareness systems, where networked sensors track critical events via coordination. They propose a linear-chain scheme for all-to-all broadcasting and data gathering. They also propose a binary-combining scheme for data gathering which divides each communication round into levels in order to balance the energy dissipation in sensor networks. For broadcasting, the linear-chain scheme starts data transmission with a packet at the beginning of a chain. Each node along the chain attaches its own data to this packet. Eventually, information of the whole network reaches the end of the chain. The same procedure runs in the reverse direction to complete all-to-all broadcasting. The linear-chain scheme can also be applied to gather data in sensor networks. To gather data, each node senses and transfers information along the chain to reach one particular node which will send



(a) chain formation using greedy method

(b) Data fusion at the leader node, and transmitting it to BS

Figure 3.2: PEGASIS protocol chain.

data to a remote BS. Such a scheme is named as PEGASIS [Lindsey and Raghavendra, 2002].

PEGASIS is the first protocol which uses chain oriented topology for periodic data collection from the target field. PEGASIS forms a chain of the sensor nodes and uses this chain as the basis for data aggregation. In PEGASIS, the chain is formed using a greedy approach, starting from the node farthest to the sink. The nearest node to this is added as the next node in the chain. This procedure is continued until all the nodes are included in the chain. A node can be in the chain at only one position. Figure 3.2(a) shows the chain creation method. In this figure, the node C0 lies furthest from the BS, chain construction starts from the node C0, which connects to the node C1, because C1 is the closest node to C0. Further, the node C1 connects to its closest node C2; the node C2 connects to the closest node C3, and so on. In this fashion a chain C0-C1-C2-C3-C4-C5 is created. Figure 3.2(b) shows the data collection strategy adopted by PEGASIS. In the constructed chain, a leader node for each round is selected randomly. The authors argue that randomly selecting head node provides benefit, as it is more likely for nodes to die at random locations thus providing robust network. All nodes send their data to the leader node, and then, the leader node sends the data to the BS. For example, in Figure 3.2(b), C3 is selected as the leader node. The node C5 passes its data to the leader node C3 via the node C4.

PEGASIS suffers from several problems. First, in this protocol the role of the leader node changes in every round of data collection. This causes extra overhead. Moreover, when a node is selected as the leader, the protocol considers neither the distance of the node from the BS, nor its energy level. Additionally, the chain in PEGASIS is constructed by a greedy algorithm. Using this chain causes some problems, such as unexpected long transmission time, and non-directional transmission to the BS. These problems affect the energy efficiency adversely. All nodes in sensor networks transmit their data in order. Therefore, the delay increases linearly as the number of nodes increases. Thus, PEGASIS is not scalable for large-scale WSNs. PEGASIS also causes redundant transmission of data, because in PEGASIS there is a single leader.

To resolve the delay problem of PEGASIS, a 3-level PEGASIS [Lindsey et al., 2002b] is proposed. In 3-level PEGASIS, the chain is cut into several chains. Each chain has a leader which gathers data from its neighbours and sends aggregated data to the upper level leader. The delay may decrease with 3-level PEGASIS. However, 3-level PEGASIS raises wireless interference problem because it does not consider the relative location of nodes. Another problem is that unexpected long transmission may occur because the leader of a chain sends a packet to the upper leader or the sink node by one hop transmission.

Du et al. [2003] provide an algorithm for constructing the energy efficient chain called minimum total energy (MTE) chain. These chain construction algorithms use centralized approaches for constructing the chain and elect the leader node for transmitting data back to the sink by taking turns. However, if the remaining energy of each node is not taken into account in the leader election, the nodes with low remaining energy will easily run out of energy, leaving just a small number of survival nodes performing the sensing task. From the viewpoint of network lifetime, this is not ideal.

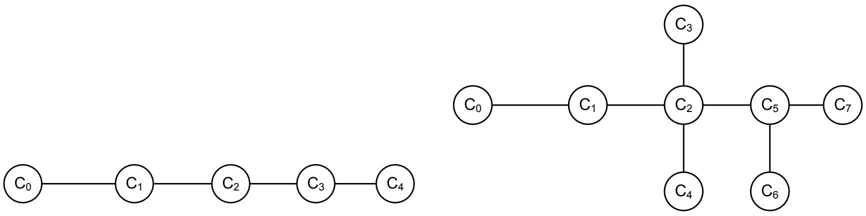
Both PEGASIS and MTE approaches use centralized chain construction. Firstly, their transmission cost calculation based on distance may not reflect the exact cost in different practical environments due to radio irregularity as indicated in [Liu et al., 2008]. Secondly, these centralized approaches may not scale well for large network or large number of nodes. Moreover, after some time, nodes far away from the sink easily run out of battery since they consume more energy to transmit to the sink as a leader.

PAC [Pham et al., 2004] addresses these issues by constructing the chain using a distributed algorithm. PAC is a chain oriented routing scheme, and here the distributed algorithm presented for constructing the routing chain is based on the minimum cost tree. In this protocol, the transmission cost is calculated based on the received signal strength between nodes. Therefore, it does not require global knowledge of nodes' location information and provides more accurate communication cost calculation among nodes under different practical deployment environments. The proposed power aware mechanism for leader node election in the chain ensures more uniform energy consumption among nodes. Thus, in PAC, all nodes die approximately at the same time, which provides better active network operation time than the case where there are only a few nodes still functioning while almost other nodes have died. However, the problem of PAC is that it constructs a single chain, which causes delay in gathering data from all the sensor nodes of the network. This protocol also requires very high processing complexity for a large network, and thus is not applicable for a large-scale WSN.

In this chapter, a multiple-chain oriented topology is proposed. That means, multiple chains are constructed using the deployed sensor nodes in the target field. The chains are constructed in a way to solve the above-mentioned problems of different chain oriented protocols. Furthermore, a network management protocol is associated with the proposed logical topology, so that the network can be managed in such a way as to contend with the resource constraints of WSNs

3.4 Description of the Proposed Topology Construction

This section describes the proposed multi-chain oriented logical topology in detail. The section is divided into several subsections. First, 3.4.1 describes the basic structure of the proposed topology. Section 3.4.2 describes different phases in the proposed logical topology construction. Chain construction algorithm for the proposed topology is discussed in 3.4.3. Section 3.4.4 discusses the leader selection principles. In designing the proposed topology, various issues arise, such as the number of hierarchical layers, number of chains in the system, number of nodes in a chain, time to change the leaders, etc. These issues are discussed in Section 3.4.5. Finally, the communication abstraction of the proposed topology is discussed in 3.4.6.



(a) Simple chain (b) Complex chain

Figure 3.3: Types of chains - simple chain and complex chain.

3.4.1 Basic structure of the proposed logical topology

The features of the basic structure of the proposed logical topology are listed below.

i) All the deployed sensor nodes in the target field take part in the logical topology construction process.

ii) The proposed logical topology consists of multiple chains. Hence, the topology is called multi-chain oriented topology. These chains are called lower-level chains.

iii) All the chains of the proposed topology are simple chains, rather than complex chains. A simple chain is defined as a chain where each member node of the chain has, at the most, two neighbouring nodes. On the other hand, a member node may have more than two neighbouring nodes in a complex chain. Figure 3.3 shows an example of both simple chain and complex chain. Note that, in Figure 3.3 the member node C_2 has four neighbouring nodes - C_1, C_3, C_4, and C_5.

iv) In a lower-level chain, the distances between any two successive nodes are called links. Thus, a chain that consists of n number of sensor nodes has $(n-1)$ links. The sum of these $(n-1)$ links is called the length of that chain.

v) The length of each chain of the proposed topology is similar. As it is assumed that the sensor nodes are deployed randomly in the target field, constructing multiple chains having exactly the same length may not always be possible. However, the proposed logical topology creates similar lengths of chains to avoid uneven energy consumptions by the chains of dissimilar lengths.

vi) For each chain, a member node of the chain is elected as a leader of the chain. These leaders are called lower-level leaders.

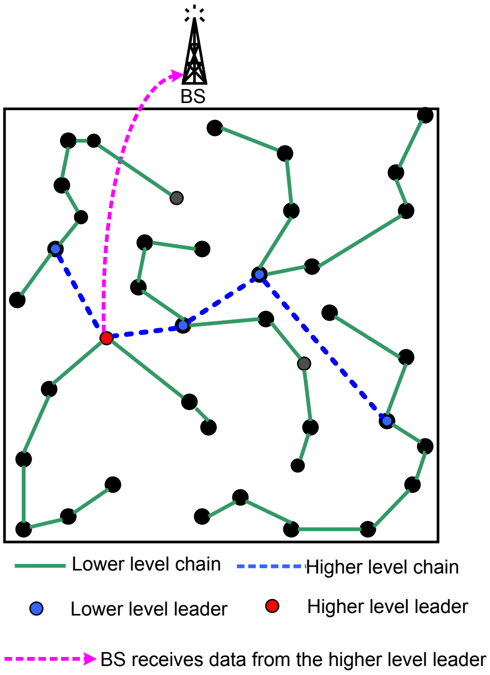


Figure 3.4: A sample model of the proposed topology.

vii) The lower-level leaders construct a higher-level chain. Similarly, a member node of the higher-level chain is elected as the leader of the chain. The leader is called the higher-level leader.

A sample architecture model of the proposed logical topology is depicted in Figure 3.4. This figure shows the logical topology using two hierarchical layers.

3.4.2 Different phases of the proposed topology

The proposed logical topology can be described using three phases, namely i) topology formation phase, ii) steady state phase, and iii) topology update phase. Figure 3.5 demonstrates these phases with respect to a timeline. Additionally, Figure 3.6 demonstrates the transitions among different phases.

At the initial stage of the sensor deployment in the target field, the topology formation phase starts. This phase takes place only once. After that, the steady state phase and the topology update phase come in turns. At the beginning of the topology formation phase no sensor nodes knows about any other sensor node in the target field. Each of the deployed sensor nodes then reports its individual characteristics to all

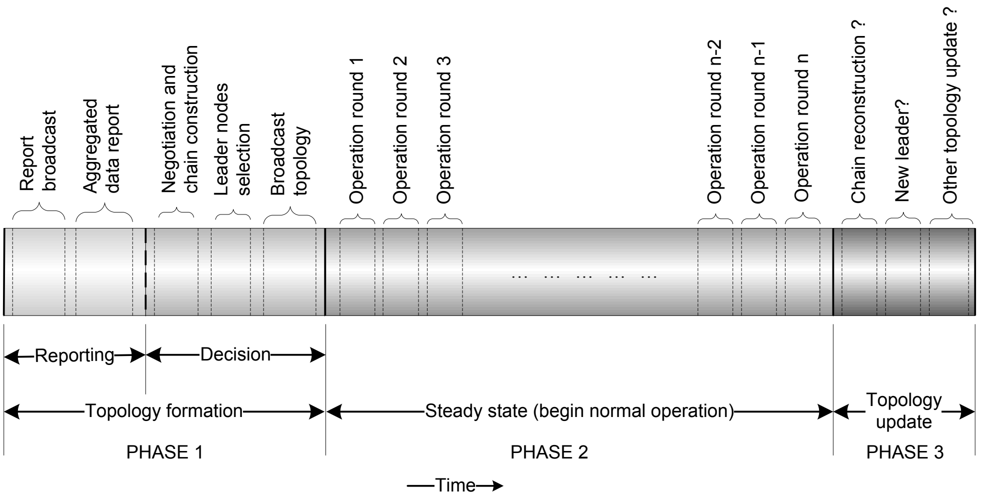


Figure 3.5: Timeline of the proposed topology.

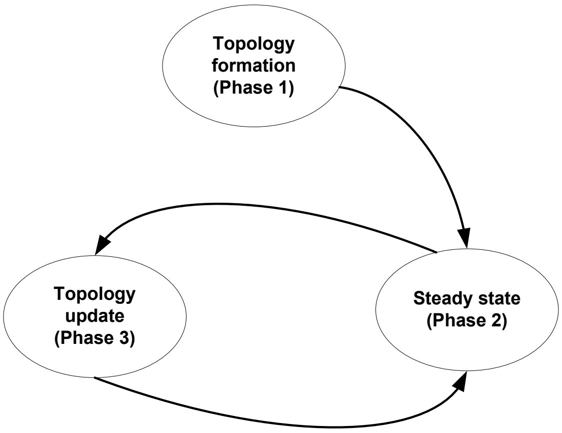


Figure 3.6: Transitions of different phases of the proposed topology.

of its neighbouring sensor nodes using broadcasting. A sensor node, receiving broadcasted messages by its neighbouring nodes, calculates the distances between itself and the neighbouring nodes. Additionally, each sensor node aggregates the reports it collects from its neighbouring nodes. After reporting, all the sensor nodes negotiate with their neighbours and construct several chains. When the chain constructions finish, lower-level leaders are elected for each chain. Each lower-level chain then broadcasts the topology, describing the member nodes, successor-predecessor lists, and TDMA allocations. At this point, the topology formation phase is ended, and the deployed sensors are ready for their normal operation.

At the end of the topology formation phase, the steady state phase begins. In this phase, the sensor nodes start their normal operation. Without the loss of generality, it can be assumed that the sensors are deployed in the target field to collect some data. The steady state consists of several number of rounds. A round begins whenever the sensor nodes start their sensing. A round finishes when the higher-level leader collects all sensed data via the lower-level leaders, and then sends the data to the BS.

After the end of a fixed number of rounds in the steady state, the topology update phase takes place. The tasks of this phase are to maintain the topology, such as selection of new lower-level leaders, construction of a higher-level chain, selection of a higher-level leader, and reconstruction of chains, if necessary.

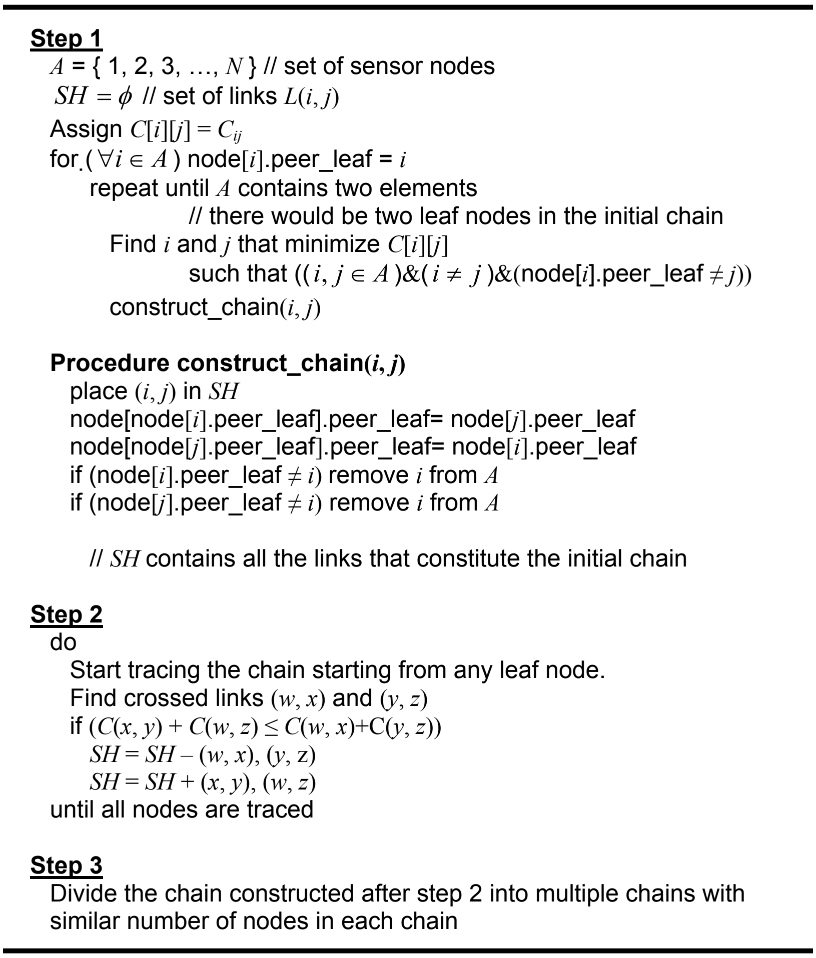
3.4.3 Chain construction algorithm for the proposed topology

The proposed chain construction algorithm consists of three steps, namely i) generating the shortest-path chain, ii) link exchange, and iii) pruning. Step one generates an initial single chain which is derived using the Kruskal minimum spanning tree algorithm. This initial chain may not be optimized, because of the existence of some cross links. At steps two and three, these cross links are removed, the chain is reconstructed and pruned to multiple chains. The chain construction algorithm is depicted in Figure 3.7.[1] Detailed descriptions of the steps are provided below.

Step 1. Configuring the initial chain. This step generates an initial chain, which is derived from the Kruskal minimum spanning tree algorithm by giving an additional constraint of a maximum degree of 2. This algorithm selects a link, one by one, through a specified routine. Since links are selected as long as a loop does not occur, several complex chains (see Figure 3.3(b)) can be generated during generating the chain. When some links are formed, the next link is the shortest link among links that connect those nodes whose degree is under 2. However, the two end nodes are not included in a same sub-chain.

Step 2. Link Exchange. For large number of nodes, there is a very possibility

[1] The chain construction algorithm, depicted in Figure 3.7, employs rigorous computations. It is assumed that the computations are performed at the base station, since the computational complexity may not be feasible for resource-constrained sensor nodes. However, the sensor nodes take part in topology construction by gathering neighbours information and aggregating them. When the base station completes the calculation it broadcast the topology to all sensor nodes.



```
Step 1
  $A$ = { 1, 2, 3, …, $N$ } // set of sensor nodes
  $SH = \phi$ // set of links $L(i, j)$
  Assign $C[i][j] = C_{ij}$
  for.( $\forall i \in A$ ) node[$i$].peer_leaf = $i$
    repeat until $A$ contains two elements
          // there would be two leaf nodes in the initial chain
      Find $i$ and $j$ that minimize $C[i][j]$
          such that (($i, j \in A$)&($i \neq j$)&(node[$i$].peer_leaf $\neq j$))
      construct_chain($i, j$)

  Procedure construct_chain($i, j$)
    place ($i, j$) in $SH$
    node[node[$i$].peer_leaf].peer_leaf= node[$j$].peer_leaf
    node[node[$j$].peer_leaf].peer_leaf= node[$i$].peer_leaf
    if (node[$i$].peer_leaf $\neq i$) remove $i$ from $A$
    if (node[$j$].peer_leaf $\neq i$) remove $i$ from $A$

      // $SH$ contains all the links that constitute the initial chain

Step 2
  do
    Start tracing the chain starting from any leaf node.
    Find crossed links ($w, x$) and ($y, z$)
    if ($C(x, y) + C(w, z) \leq C(w, x)$+C$(y, z)$)
      $SH = SH -$ ($w, x$), ($y$, z)
      $SH = SH +$ ($x, y$), ($w, z$)
  until all nodes are traced

Step 3
  Divide the chain constructed after step 2 into multiple chains with
  similar number of nodes in each chain
```

Figure 3.7: Chain construction algorithm.

that the initial chain generated after step 1, includes some cross links (see Figure 3.8). In this step, cross links are removed, and the chain is pruned to multiple chains. Cross link removal process takes place when there is available links, whose lengths are shorter than that of cross links. This process is called link exchange.

In Figure 3.8, the nodes are numbered from 1 to n. In this figure, dotted lines represent sub-chains that are consisted of several links. The solid lines in this figure represent a single link. When the process of link exchange occurs, the order sub-chain from $i+1$ to j is reversed. To exchange two links of the chain as from $(i, i+1)$ and $(j, j+1)$ to (i, j) and $(i+1, j+1)$, the following condition should be satisfied: $C(i, i+1) + C(j, j+1) \geq C(i, j) + C(i+1, j+1)$ where $C(i, j)$ denotes the length of the link (i, j).

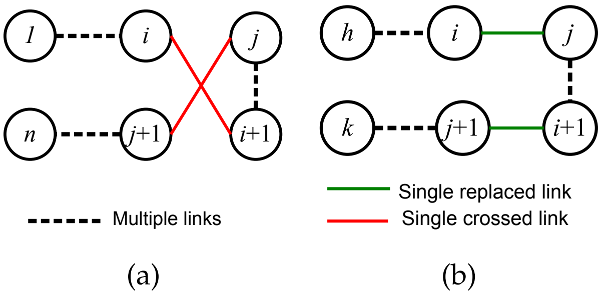


Figure 3.8: Link exchange. Crossed links are replaced by new links.

Step 3. Pruning. At the end of the link exchange, an optimal chain is generated. To create multiple chains from this optimal chain, each node of this chain is traced, starting from the farther end of the chain from the BS. Tracing process takes place from one node to its neighbouring node until the number of nodes traced is equal to C_N. Here, C_N is the optimal number of node in a chain. (The discussion about C_N value is be presented shortly.) At this point all the nodes are which are already been traced are pruned from the initial chain. This pruning process continues until all the nodes of the initial chain are traced.

3.4.4 Leader nodes selection

Given a chain structure, leader scheduling is to determine which nodes play the role of leaders in operational rounds. The goal is to prolong network lifetime, i.e., to maximize the number of operational rounds. The following discusses the general issues about selecting leader node(s) in respect to power consumption.

Suppose any node in a chain can be elected as a leader, and the leader is responsible to send the aggregated data to the BS. The maximum number of operational rounds that can be achieved before any node exhausts its power is analysed first. Without loss of generality, it can be assumed that nodes in the chain are numbered sequentially as $1, 2, \ldots, n$. Let e_i be the energy consumed by the node i in transmitting a data message to the BS. Let $\rho_{i,j} = kE_{elec} + k\varepsilon_{amp}\left(d(i,j)\right)^{\alpha}$ be the energy consumed by the node i, and $e_r = kE_{elec}$ be the energy consumed by the node j when the node i transmits a k-bit message to the node j. When some node i is selected to be the leader, every node numbered $j < i$ (if any) expends $\rho_{j,j+1}$ energy in sending data to the node

$j+1$, at which energy e_r is consumed to receive the data. Likewise, every node numbered $k > i$ (if any) expends $\rho_{k,k-1}$ to send data to the node $k-1$, where energy e_r is expended in receiving the data. The leader transmits the collected data to the BS, consuming energy e_i. Suppose that, every node i is scheduled to be the leader x_i times. Table 3.1 shows the energy expense of every sensor node in this case.

Table 3.1: Energy consumption by different nodes while acting as a leader

Node ID	**Energy spent to send message to the BS**	**Energy spent to send message to neighbours**	**Energy spent to receive neighbour's message**
1	e_1x_1	$\rho_{1,2}\sum_{j=2}^{n}x_j$	e_rx_1
$i \in \{2,3,\ldots,n-1\}$	e_ix_i	$\rho_{i,i-1}\sum_{j=1}^{i-1}x_j \quad + \rho_{i,i+1}\sum_{j=i+1}^{n}x_j$	$e_r\left(\sum_{j=1}^{i-1}x_j+2x_i+\sum_{j=i+1}^{n}x_j\right)$
N	e_nx_n	$\rho_{n,n-1}\sum_{j=1}^{n-1}x_j$	e_rx_n

x_i : the number of times node i is selected to be the leader.

e_i : the amount of energy consumed in transmitting message from node i to BS.

$\rho_{i,j}$: the energy consumed by i in transmitting a message to j.

e_r : the energy consumed by any node in receiving a message.

The optimal leader scheduling problem is to find a positive integer values of x_i's such as to maximize $\sum_i x_i$ subject to the following constraints:

$$E_1 \geq (e_1+e_r)x_1+\rho_{1,2}x_2+\rho_{1,2}x_3+\ldots+\rho_{1,2}x_n$$

$$\vdots$$

$$E_i \geq (\rho_{i,i-1}+e_r)x_1+\ldots+(\rho_{i,i-1}+e_r)x_{i-1}+(e_i+2e_r)x_i+(\rho_{i,i+1}+e_r)x_{i+1}+\ldots+(\rho_{i,i+1}+e_r)x_n$$

$$\vdots$$

$$E_n \geq \rho_{n,n-1}x_1+\rho_{n,n-1}x_2+\ldots+(e_n+e_r)x_n$$

where E_i denotes the amount of energy that node i initially has.

These constraints can be formulated as

$$A \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} \leq \begin{pmatrix} E_1 \\ E_2 \\ E_3 \\ \vdots \\ E_n \end{pmatrix}$$

where

$$A = \begin{pmatrix} e_1 + e_r & \rho_{1,2} & \cdots & \rho_{1,2} \\ \rho_{2,1} + e_r & e_2 + 2e_r & \cdots & \rho_{2,3} + e_r \\ \rho_{3,2} + e_r & \rho_{3,2} + e_r & \cdots & \rho_{3,4} + e_r \\ \vdots & \vdots & \cdots & \vdots \\ \rho_{n,n-1} & \rho_{n,n-1} & \cdots & e_n + e_r \end{pmatrix}$$

Thus, the problem turns out to be a linear programming problem. Round robin leader scheduling equalizes the values of x_i's, which is generally far from optimal. The authors of PEGASIS also proposed an improvement on round robin scheduling in [Lindsey et al., 2002a]. This approach sets up a threshold of distance, and nodes are not allowed to be leaders if their distances to their neighbours along the chain are beyond the threshold.

From the above discussion, it is found that achieving optimal results in leader selection is a computationally rigorous task. Thus, instead of finding an optimal solution, the proposed topology uses a simple rule called Maximum Residual Energy First (MREF) for leader selection. This simple algorithm gives near optimal results for a lower number of nodes.[2] As in the proposed topology, there are only a few number of lower-level leaders; this algorithm perfectly suits for selecting a higher-level leader. As the name suggests, MREF selects the node that has the maximum residual energy to be the leader for network operations. Residual energy information can be piggybacked with data messages as a part of the aggregated data. If every lower-level leader attaches its own energy level to data message and lets the BS find the maximum value, it will incur an additional $O(n)$ overhead on every message. A better approach

[2] Simulation results show that for a 100 sensor node network using MREF algorithm spends around 0.18% more energy after 500 operational rounds than a network which uses linear programming to select the leader. However, the rigourous calculation may not be suitable for resource-constraint sensor nodes.

is to let every lower-level leader compares its energy level with that attached with incoming data message (if any) and send only the large one. The message overhead in this process is only $O(1)$.

For the lower-level leaders, the same selection procedures can be followed. However, since the communications of the lower-level leaders are not as energy intensive as for higher-level leader, it is proposed not to change lower-level leaders as frequently as higher-level leader. The benefits of using a slightly larger duration for selecting lower-level leaders are i) less communication overhead, ii) reduced required time for leader selection at every round, and iii) maximum utilization of higher-level chain.

3.4.5 Design issues of the proposed logical topology

While designing the proposed logical topology, different issues need to be discussed. Some of them are design issues, such as the number of chains in the system, the number of nodes in a chain, time when the leaders should be changed or the chains should be reconstructed/updated. Other issues are regarding the network management, such as arrival of a new node, or dead / aberrant nodes etc. These issues are discussed below.

A. Total number of chains in the system

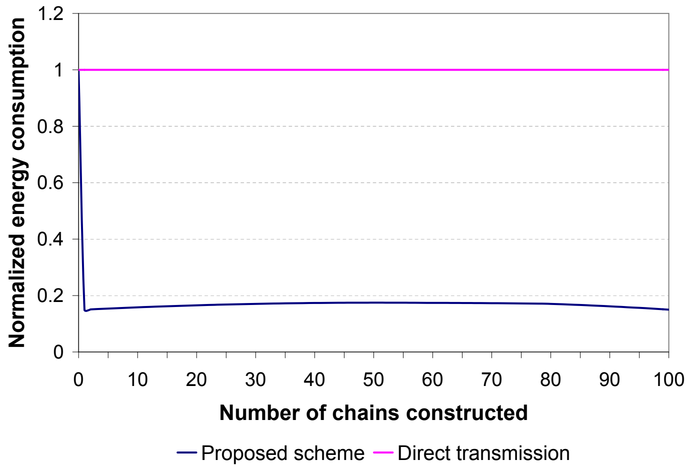
The system can determine, a priori, the optimal number of chains (lower-level) to have in the system. This depends on several parameters, such as the positions of the sensor nodes, and the relative costs of computation versus communication. The proposed topology was simulated for a data collection application using a network where 100 sensor nodes were randomly deployed. The value of the radio parameters of transmitter and receiver electronic that were used in the simulation are $E_{tx-elec} = E_{Rx-elec} = E_{elec} = 50$ nJ/bit. The transmit amplifier was assumed 100 pJ/bit/m^2. A computation cost of 5 nJ/bit/message to fuse 2000-bit messages were further assumed. In the experiment, the number of chains in the system was varied gradually to observe its impact on energy consumption, and delay. Figure 3.9 shows how the energy dissipation in the system varies as the number of chains in the system are changed. Note that, zero chain means no lower-level chain is constructed. Thus there would be no higher-level chain as well. In this situation, each sensor node directly transmit its sensed data to the BS. Also note that, 1 chain means there would be no higher-level chain, and 100 chains means there is actually no lower-level chain (because of only one member in each chain), thus a single higher-level chain. Therefore, both 1 chain and 100 chains refer to the same system as PEGASIS does. Figure 3.9 suggests that energy consumption would be lower if the number of chains can be kept below 10 or above 80. However, a large number of chains would cause more overhead . Thus, for the proposed topology, the number of chain is maintained at 6%-8% of the sensor nodes. That means, for a target field of 200 sensor nodes deployed, 12 to 16 chains would be constructed.

B. Optimal number of nodes in a chain

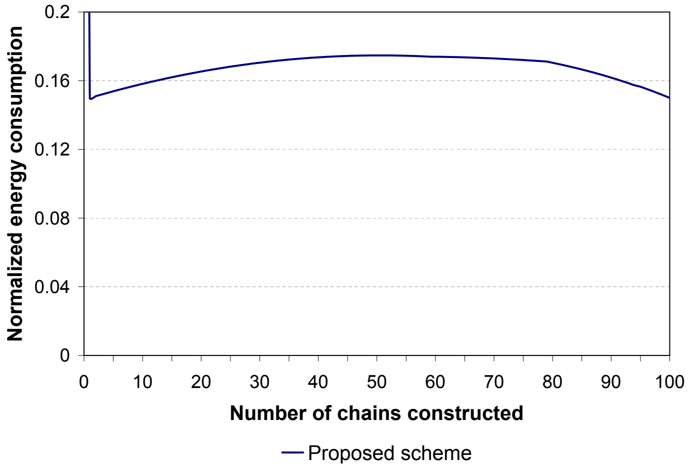
The optimal number of sensor nodes in a chain, denoted as C_N, is the number of nodes that should be included in each chain during the chain construction phase. It can be argued that, if the number of nodes in a chain is fewer than C_N, both the required time and energy dissipation increase in the network. On the other hand, if the number of nodes is more than C_N, energy dissipation may decrease slightly, however the time requirement increases. Additionally, for the sake of even energy dissipation distribution, the lengths of the chains should be similar. Thus, in the proposed scheme, a similar number of sensor nodes are included for each chain. Since it is assumed that sensors are deployed randomly in the target field, creating chains of exactly the same number of sensor nodes may not be possible. However, the proposed scheme maintains a similar number of nodes in each chain. Thus for a target field of 100 nodes, the number of sensor nodes in each chain $C_N = 12$ to 17.

C. Chain Reconstruction

It is important to reconstruct the chains whenever a significant number of sensor nodes in a chain expire. Otherwise, there may be possibilities that one chain contains a higher number of sensor nodes, while others contain lower number of sensors. This affects the performance of the topology, because of uneven energy dissipation by the chains. It is vital to maintain uniformity in the number of sensor nodes in all chains as only one sensor node (i.e. the higher-level chain leader) is responsible to send the aggregated data to the BS, and it has to wait for aggregated data from dif-



(a)



(b)

Figure 3.9: Normalized total system energy dissipated versus the percent of nodes that are chain leaders.

ferent lower-level leaders. Thus, the uniformity of number of sensors in chains affects network lifetime. If a chain consists of a lower number of sensors, the probability of a sensor in that chain to be selected as local leader will be higher. Thus, a chain of short length is likely to lose sensors more often. It is obvious that if chains are reconstructed frequently, e.g. whenever only 4%-5% sensors of the chain die, it causes extra overhead. On the other hand, if the chain is reconstructed whenever 40%-50% sensors of the chain die, the uniformity among the chains is destroyed. To answer the question of when a chain should be constructed, simulation experiments were performed. To

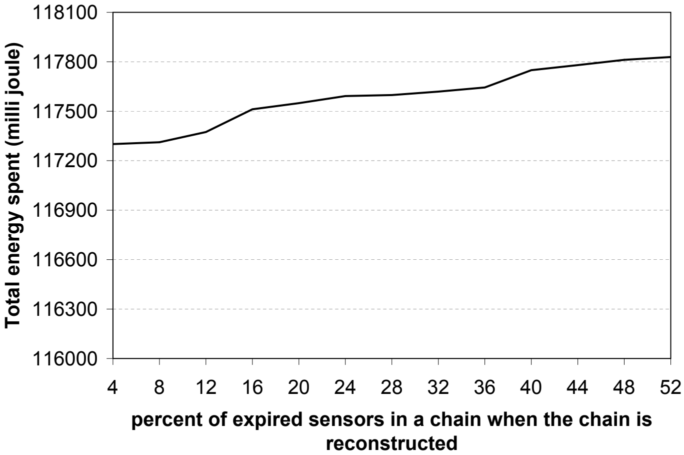


Figure 3.10: Total energy spent vs. percent of expired sensor nodes in a chain when the chain is reconstructed.

find the optimal value, chains were reconstructed varying the percentage of sensors' death in the chains, and its effects were observed against total energy spent, lifetime of the network, and time required to complete 100 rounds. Figure 3.10, 3.11, and 3.12 show the simulation results. Figure 3.10 shows that although the energy consumption increases whenever chains are reconstructed less frequently, the amount of energy difference is not extreme. Figure 3.11 shows that the lifetime increases from 590 to 610 between 4%-52% of sensors death. Thus, lifetime increase rate is slightly more than 3.5%, which is quite small. This figure shows that the lifetime (when around 5% of the deployed sensor nodes die) remains almost steady[3] with a little peak around 20% sensors of chains death. Figure 3.12 shows that time requirements[4] decrease whenever chains are reconstructed less frequently. Time requirement sharply falls between 4%-20% of sensors' death and then decreases slowly afterwards. Thus, it is concluded to reconstruct chains whenever around 20% of the sensors of a chain are expired.

To track how many sensor nodes are expired in a chain, the following method can be used. When data are fused in every sensor of a chain, each sensor adds its tag with the data packet. For example, let node n_1 sends data to n_2, and n_2 fuses n_1's data and send it to n_3. However, if n_2 is dead, n_1 sends data directly to n_3, and thus the

[3] Reproducing the Figure 3.11 using absolute scale will give the a better impression that lifetime remains almost steady.

[4] Time requirements refers to the amount of time spent (in seconds) to perform 100 operational rounds. Refer to Figure 3.5. Time count starts at the beginning of Phase 1 and ends when 100 operational rounds are completed.

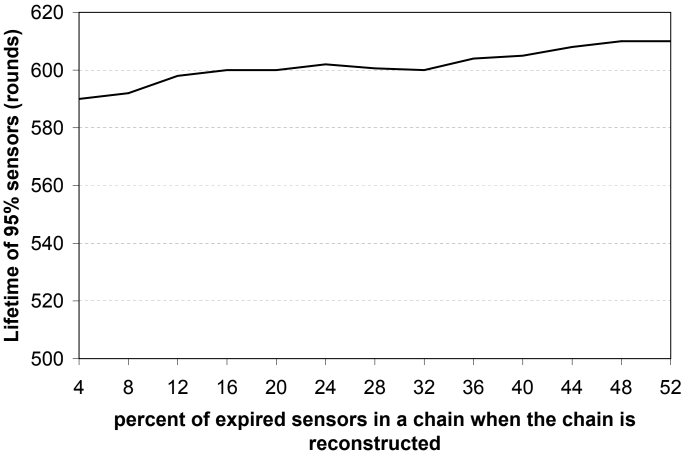


Figure 3.11: Network lifetime vs. percent of expired sensor nodes in a chain when the chain is reconstructed.

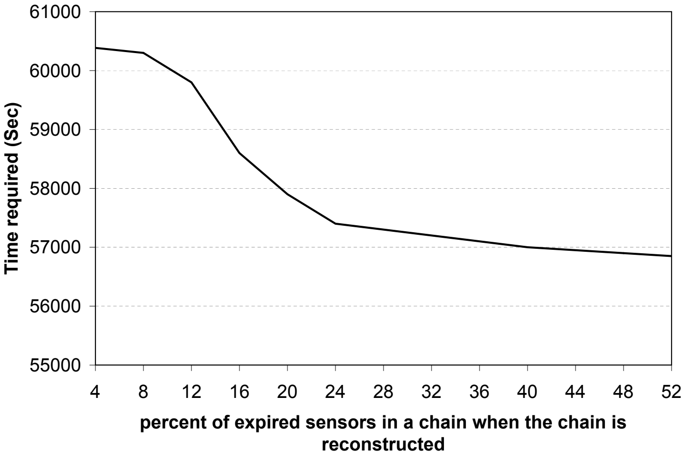


Figure 3.12: Time required vs. percent of expired sensor nodes in a chain when the chain is reconstructed.

node n_3 knows that n_2 is dead. In this way every lower-level leader come to know how many of its members are dead. In a similar fashion, when the higher-level leader collects data from all lower-level leaders, it knows how many sensors are dead in the network. After that, the higher-level leader sends instruction accordingly to all sensor nodes.

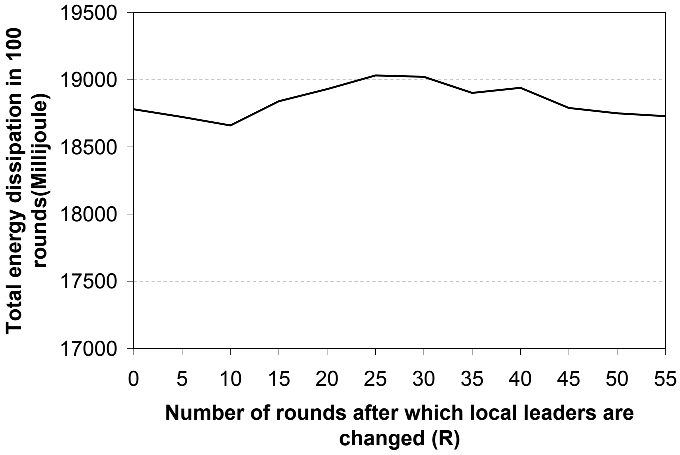


Figure 3.13: Leader selection time vs. total energy dissipation.

D. Changing lower-level leaders

The lower-level leaders should be changed periodically to distribute the energy load. PEGASIS suggests changing the leader node in each round. However, for the proposed topology, if the lower-level leaders are changed at every round, it causes extra energy expenditure for negotiations to select leaders, as well as causes delay. In addition, the higher-level chain would be utilized fully if the lower-level leaders are changed after a number of rounds. On the contrary, if the lower-level leaders are not swapped with other member nodes for long time, they will quickly drain out energy because of excessively long transmissions. Therefore, in the proposed logical topology, lower-level leaders are changed after R rounds, where the value R depends on some criteria, such as i) total energy dissipation in the network, ii) maximum number of round when the first sensor node dies, and iii) delay introduced in the network against the different values of number of rounds.

Recall that, a new leader is selected in the Phase-3 of Figure 3.5. In this phase, each member node of a chain sends a token containing information about its residual energy to the leader node of the chain. The leader node, using the MREF algorithm (Section 3.4.4), decides which would be the next leader, and then disseminates this information to all members of the chain (see Figure 3.19). Thus, there are at most $2n$ number of token passing take place (where n is the number of nodes in a chain). The average chain size for a 100 node network is typically between 12 and 18. Also

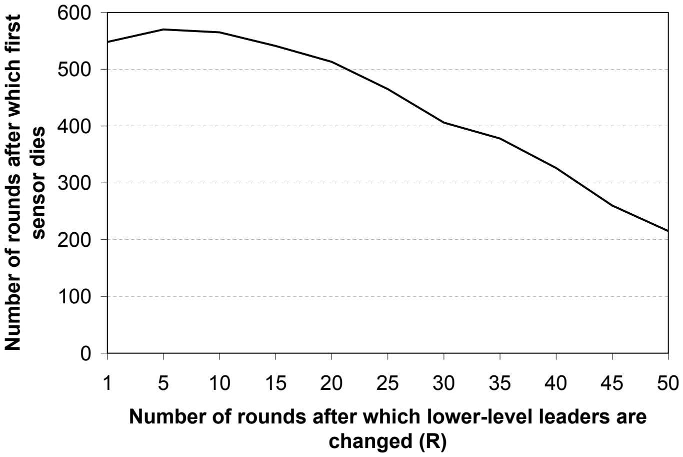


Figure 3.14: Leader selection time vs. network lifetime.

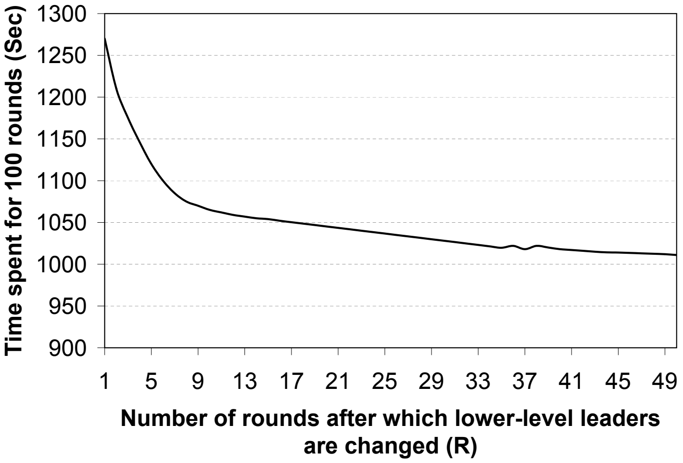


Figure 3.15: Leader selection time vs. time required.

the size of the token (few bytes) is very small compared to data packets (2000 bytes). Therefore, the cost associated with the token passing is negligible [Tabassum et al., 2006]. Hence, the overhead of the leader selection is negligible.

Figures 3.13, 3.14, and 3.15 show the simulation results (average of 100 simulation runs), which are used to determine when the lower-level leaders should be changed. Figure 3.13 shows the relationship between R and the total energy spent in the network. The figure shows that total energy consumption actually does not follow a relationship with R.[5] Figure 3.14 shows that the value R greatly affects the network

[5] Note that, Each time sensors were assumed to be deployed randomly in the target field. This is why

lifetime. As the value of R increases, the network lifetime decreases. This is because, when the same sensor nodes are working as leaders for long periods, they deplete energy quickly compared to other sensor nodes. Figure 3.15 shows the relation between the time spent in the system[6] and R. It is obvious that, if leaders are changed in each round, the delay would be highest. The amount of time required decreases as the value of R increases. From the experimental result, it is proposed that the lower-level leaders are changed in around $C_N/2$ number of rounds.

E. Inserting additional nodes into the network

Additional nodes may be inserted into the network at any time. Before a node is inserted, the BS records and stores its unique ID and will insert the node into a nearby chain with the least number of nodes. This will help minimise the event of a chain monopolising bandwidth if it contains a greater number of nodes than other chains which are communicating. The node will then organize itself within its chain.

F. Identifying and isolating aberrant nodes

Sensor nodes that do not function as specified must be identified and isolated in order to continue the desired operation of the sensor network. An aberrant node may be the result of an attack or may act maliciously due to unexpected network behaviour. According to Fei et al. [2005], an aberrant node is one that is not functioning as specified, and may cease to function as expected for the following reasons:

- It has exhausted its power source.
- It is damaged by an attacker.
- It is dependant upon an intermediate node and is being deliberately blocked because the intermediate node has been compromised.
- An intermediate node has been compromised and is corrupting the communication by modifying data before forwarding it.

constant results were not found. However, energy dissipation deviation is only 2.04%.

[6] Time requirements refers to the amount of time spent (in seconds) to perform 100 operational rounds. Refer to Figure 3.5. Time count starts at the beginning of Phase 1 and ends when 100 operational rounds are completed.

- A node has been compromised and communicates fictitious information to the BS.

Therefore, the WSN should be maintained by identifying an aberrant node quickly and isolating it from the sensor network. The protocol named SecCOSEN [Mamun and Ramakrishnan, 2008] can be used for the authentication purposes. This protocol perfectly suits the logical topology, as it was designed for a multi-chain oriented logical topology. Using this protocol, a node would authenticate the node from where it receives data/messages. If a node is not able to authenticate another node in the chain, the former node reports about the incident to the chain leader. In addition, a node also maintains a timer for identifying any dead node with the help of timeouts. The identifier node then reports the incident to the leader node.

G. Number of Layers

Although in this chapter the architecture of the proposed multi-chain oriented logical topology is described using a two-layer model, the number of layers can be extended based on the number of sensor nodes in the target field. In this case, member nodes of a layer-1 (the lowest level) chain send their data to the layer-1 leaders. All layer-1 leaders construct several layer-2 chains. In each layer-2 chain, a node is elected as a layer-2 leader. Layer-1 leaders send the data to the layer-2 leaders via layer-2 chains, and so on. In this way, the highest level leader collects all data, and then sends it to the BS. Fox example, Figure 3.16 depicts a model of multi-chain oriented logical topology with three hierarchical layers. In this figure, the black nodes are the member nodes of layer-1 green-coloured chains. In each layer-1 chain, a node is elected as a leader, and marked as green. All green-coloured layer-1 leaders construct several layer-2 blue coloured chains. Similarly, in each layer-2 chain, a node is elected as a leader. They are marked in blue. All the blue-coloured layer-2 leaders further construct a layer-3 chain, and one of its members is elected as a leader. This leader is the highest level leader, and is marked in red. A black node sends the data to its leader (green) via the green chain, a green node sends the accumulated data to its leader (blue) via the blue chain, and finally, a blue node sends its accumulated data to its leader (red). The highest level leader (red) then sends the data to the BS.

Figure 3.17 shows the simulation results and comparison between two-layered

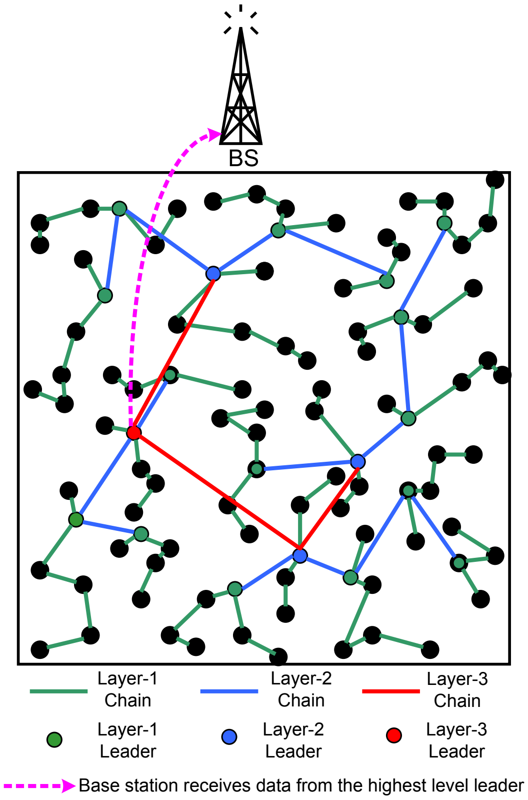


Figure 3.16: Three-layered hierarchical multi-chain oriented topology.

and three-layered chains with respect to the time required for 100 rounds. The figure demonstrates that two-layered chains take less time than three-layered chains until the number of sensors is not greater than 1600. On the other hand, when the number of sensor nodes exceeds 1600, a three-layered system takes less time compared to a two-layered architecture. The same situation arises for the total energy consumption experiment. This is depicted in Figure 3.18. Until 1500 sensor nodes, two-layered architecture saves more energy than three-layered architecture. However, if the number of sensor nodes exceeds 1500, three-layered architecture saves more energy compared to two-layered architecture. Thus, it is concluded that, if the number of sensor nodes in the target field is less than 1500, two-layered architecture is used, and if the number of sensor nodes in the target field is equal to or more than 1500, three-layered architecture is more suitable. In general, as the number of node increases in the network, by increasing the number of tiers, both the time complexity and the energy consumption can be reduced. Hence, as the network size increases, more tiers are preferable.

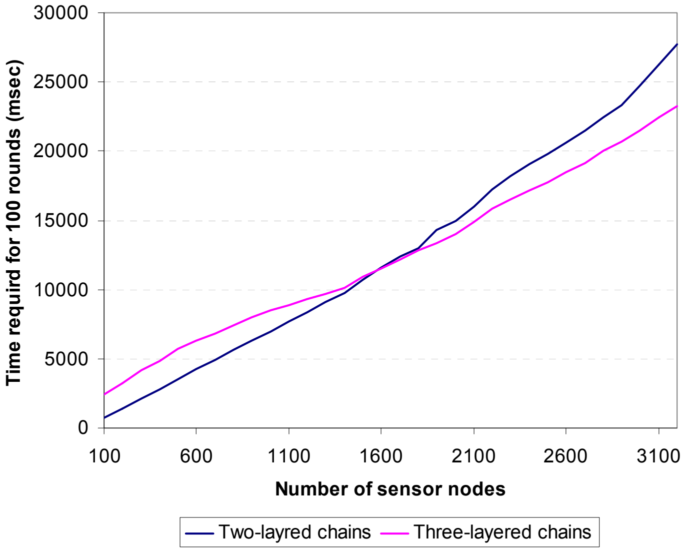


Figure 3.17: Timing differences between two-layered and three-layered chains.

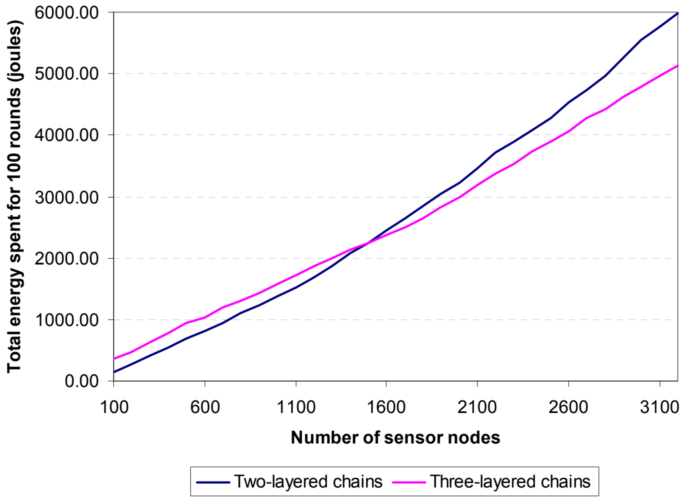
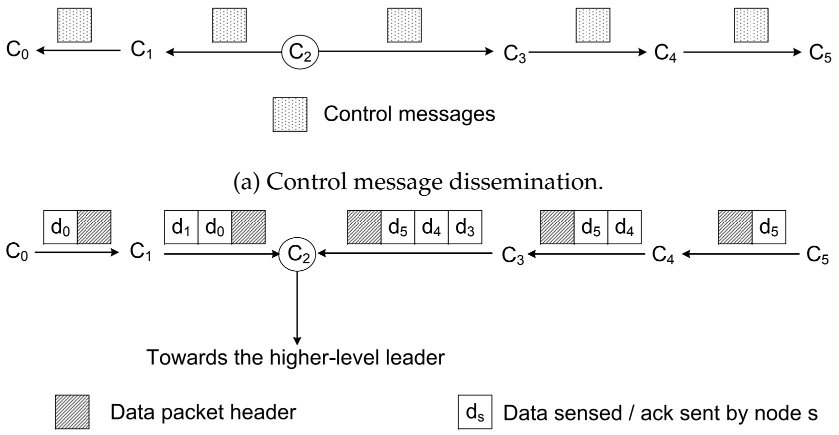


Figure 3.18: Energy consumption differences between two-layered and three-layered chains.

3.4.6 Communication abstraction of the proposed topology

This section describes the communication abstraction for the proposed multi-chain oriented logical topology. Communication is fundamental to any logical topology of WSNs. The power of a WSN comes not from the capabilities of the individual devices, but from the collective capabilities achievable through wireless communication.



(a) Control message dissemination.

(b) Sending data towards the lower-level leader.

Figure 3.19: Communications in a chain.

Addressing the intricacies of wireless communication can be a difficult, error-prone task. This is especially true of WSN applications, where the number of participating devices can be large, the communication patterns can be complex, and the network links are ad-hoc and unreliable. However, the proposed topology restricts the communications of a sensor node to only its successive nodes in its chain. Thus, the burdens of multicasting and broadcasting are taken out of the sensor nodes. The communication abstraction of the proposed topology can be divided into two parts, namely i) communications in a chain, and ii) communications among the chains.

A. Communications in a chain

In a chain, sensor nodes communicate with each other to disseminate control information and sensed data. Communications among the sensor nodes are restricted to only the successive sensor nodes. Figure 3.19 shows the communication pattern inside a chain. In this figure, six sensor nodes (C_0 to C_5) construct a chain. C_2 is the lower-level leader of the chain. The lower-level leaders disseminate information and control messages to all the member nodes of their chains. These information and control messages are propagated hop-by-hop from one sensor node to its successive neighbouring node. For example, Figure 3.19(a) shows that the leader node C_2 sends the control information to the nodes C_1 and C_3. After copying the control message, the node C_1 sends the control message to the node C_0. On the other hand, the control message is propagated from C_3 to C_4, and then C_4 to C_5. As the nodes C_0 and C_5 are the end nodes of the chain, they refrain from sending the control message further to any node.

For sending the sensed data, each sensor node sends data to its successive node towards the leader of the chain. For example, in Figure 3.19(b), the node C_0 sends its sensed data to the node C_1, while the node C_1 merges its own data with C_0's data, and sends them to the leader node C_2. Similarly, the node C_5 sends its data to the node C_4, C_4 then sends C_5's data and its own data to the node C_3. The node C_3 further accumulates those data with its own data, and sends them all to the leader node C_2.

B. Communication among the chains

Different lower-level chains communicate to each other using the higher-level chain. The lower-level leaders accumulate data sent by the member nodes of the chains, and transfer them to the higher-level leader. The higher-level leader then sends the data to the BS.

If the BS, or the higher-level leader wants to send some information, or control messages to the chain members, the communication path remains the same, except the direction is opposite. In this case, the communication pattern in similar to hub-and-spoke topology. Figure 3.20 shows a situation where the BS sends some control messages to the member nodes of a chain.

3.5 Network Management Architecture of the Proposed Topology

This section presents the network management architecture and processes for the proposed logical topology. Network management is the process of managing, monitoring, and controlling the behaviour of a network. The unique characteristics and restrictions of WSNs make the management approach different from the traditional wired networks and mobile ad hoc wireless networks. Thus, it is necessary to take those unique features into account when proposing efficient management architec-

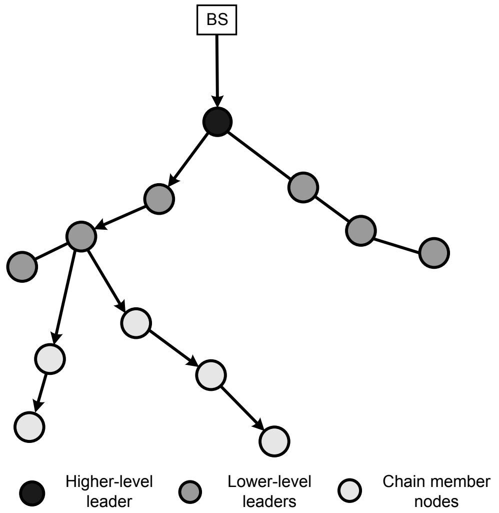


Figure 3.20: Communication from the BS to the member nodes of a chain.

tures for WSNs.

WSN management systems can be classified according to their network architecture: centralized, distributed, or hierarchical. In centralized management systems, the BS acts as the manager station that collects information from all nodes and controls the entire network. However, this approach has some problems. First, it incurs a high message overhead (bandwidth and energy) from data polling, and this limits the scalability of WSNs. Second, the central server is a single point of data traffic concentration and potential failure. Lastly, if a network is partitioned, sensor nodes that are unable to reach the central server are left without any management functionality.

Distributed management systems employ multiple manager stations. Each manager controls a sub network and may communicate directly with other manager stations in a cooperative fashion in order to perform management functions. However, this approach is complex and difficult to manage. Furthermore, distributed management algorithms may be computationally too expensive for resource-constrained sensor network nodes.

Hierarchical network management is a hybrid between the centralized and distributed approach. Intermediate managers are used to distribute management functions, but do not communicate with each other directly. Each manager is responsible for managing the nodes in its sub-network. It passes information from its sub-network to its higher-level manager, and also disseminates management functions received from the higher-level manager to its sub network. This architecture integrates the benefit from the centralized and distributed management architecture, and is more suitable for WSNs. Moreover, in WSNs, a sensor node has a small embedded processor with more limited memory and energy than a general ad-hoc node. Besides, Wireless sensor network management does not demand as many features as other network management protocols such as SNMP (Simple Network Management Protocol) and ANMP(Ad-hoc Network Management Protocol). Typically, a WSN has at least one base station, which is the most possible candidate to act as the manager. The communications with internet devices are generally implemented via the base station. For the presence of a base station, and advantages of hierarchy architecture that is effective for data aggregation (light weighting) and scalability, the wireless sensor network's management architecture should be hierarchical. As a result, hierarchical network management is chosen for the proposed logical topology.

For the proposed multi-chain oriented topology, a three-layer hierarchical management architecture is proposed.[7] Figure 3.21 represents the relationship among the different entities of the management architecture, namely manager, the sub-manager and the agent nodes. The manager is in the highest level of the hierarchy, and is placed at the BS. The lower-level chain leaders of the proposed topology work as sub-managers, and the chain member nodes work as agent nodes. The sub-managers are used to distribute management functions, and collect and collaborate management data. The manager has the global knowledge of the network states and gathers the global knowledge from the underlying network layers and sub-managers.

The proposed logical topology arranges the nodes into groups of chains and identifies a chain leader for each chain. This allows a subset of nodes to communicate with the sink nodes, conserving energy in the nodes that no longer must send data to the sinks. Often sink nodes are farther away from many nodes in the network. Chaining procedure abandons these long paths required for communication for smaller hops since nodes will only be communicating with neighbour nodes (except for the chain leaders). Besides energy and bandwidth conservation, there are other advantages of

[7] Three-layer was adopted because this is the starting point for a two-tier architecture. It is suggested in this thesis that, as the number of nodes increase more tiers for the topology is preferable. In that case, network management protocol with higher number of layers can be used.

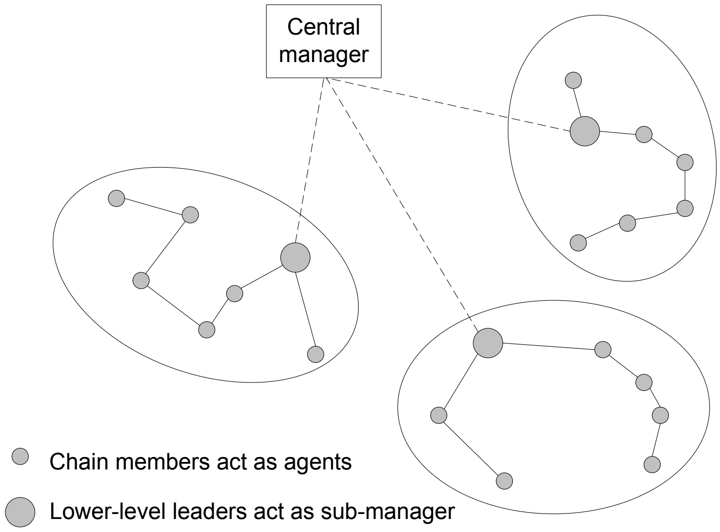

Central manager
Chain members act as agents
Lower-level leaders act as sub-manager

Figure 3.21: Different entities of the network management scheme for the proposed topology.

clustering nodes in a WSN. One advantage is that it allows for spatial reuse of resources. If two nodes exist in different non-neighbouring clusters, it may be possible for the two nodes to share the same frequency or time slot. It is also beneficial in the presence of mobility. When using clustering and a node moves, it is often only necessary to update the information in the nodes sharing a cluster with the mobile node; all nodes in the network will not have to be updated. Clustering into chains can also facilitate network management and routing since many implementations require only the chain leader to participate in these functions. In this management architecture, the chain leaders (called sub-manager often) report the data to the manager on behalf of the entire cluster.

Three major aspects of the proposed network management, namely fault detection, performance management, and security management are discussed below.

3.5.1 Fault detection

Fault detection is the process by which the network manager identifies a node which is malfunctioning or almost dead and unable to sense or transmit data. If a normal sensor node dies, it does not create much of a problem except decreasing reliability. However, if a chain leader dies, the data of that chain are lost, and in the worst case, such a failure introduces network partition in the system.

In traditional IP networks, the usual way to know whether a node is working properly or not is to get periodic *keepalive* messages from that node. However, for sensor network such message exchange is very costly. Therefore, fault detection operation in WSNs should be lightweight, and done using passive information as much as possible.

The fault of a chain member node is detected by the sub-manager (i.e., by the lower-level leaders) with the cooperation of other member nodes. A lower-level leader and all member nodes of the chain maintain a timer T for their neighbouring sensor nodes. For event driven sensor network, the sensor sends a periodic *keepalive* message to the leader node in the absence of an event. However, if the sensors are supposed to send data periodically, then by analyzing the packets, the lower-level leader can identify the sensor node that is not responding. If a sensor does not hear from its neighbouring node for a certain period of time (timer T expires), the node informs the lower-level leader about that particular sensor by sending a *negative response*. The *negative response* is piggybacked in the next data packet towards the leader. On the other hand, if a sensor hears a transmission from its sensor, it resets the timer, sends a *positive respone* towards the leader.

The lower-level leaders can miss packets from member nodes caused by collisions. For this reason, if the timer of the lower-level leader expires, then it waits a random time before declaring the alarm. If there is no *positive response* before the timer of the leader expires, or random delay is extended three times, then the leader node generates an alarm, and decides that the corresponding node is dead. The leader then informs the central manager about the dead node.

The central manager (base station) detects the faults of the sub-managers (lower-level leaders). Each lower-level leader sends information about it's neigbouring lower-level lead towards the higher-level leader using positive and negative response as described before. The higher-level leader is called *gateway node*, because all data from the sensor field are passed through this node to the base station. In a chain oriented topology, fault detection of lower-level leaders is more important than that of a chain member node. Thus, the central manger does not wait for random amount of time. The central manager maintains two timers (T_1 and T_2) for each chain leader and for the *gateway node*. In cases of periodic traffic, the central manager analyzes the packets received from the *gateway node* and can identify whether the gateway node is alive or not.

When the central manger receives a packet from the gateway node, the central manager restarts the timer T_1. If the timer expires, then the central manager suspects that a leader node is dead. As the fault should be detected immediately, the value of T_1 should not be very high. When the timer expires, central manager sends a query packet to the submanager, and waits for another time T_2. If no response is received, the central manger decides that the corresponding sub-manager is dead.

In event driven sensor networks, in the absence of events, the chain leaders or gateway send periodic message and chain leader uses the same timer mechanism to detect faults.

3.5.2 Performance management

The performance management of WSNs monitors the performance of the network and keeps resource consumption as low as possible, especially the use of energy. One of the major performance issues of the WSN is event reliability, which is defined as the number of unique data packets received by the sink node. For optimum performance, the management system sets the data generation rate of the sensors and also may keep some nodes in the sleep state and others in the normal live state.

Performance management consists of monitoring network devices and links in order to determine utilization. Utilization may vary depending on the device and link; it may include such things as processing load, network card utilization, packet-forwarding rate, error rate, or packets queued. Monitoring utilization helps to ensure there is available capacity. Monitoring the network performance assists in identifying current and future bottlenecks and aids in capacity planning. Tracking the utilization of network resources by each user is the goal of accounting management. The primary function of this information is to bill users for their use of the network and its resources. This information can be used to establish metrics and quotas. The usage information also helps the network manager to allocate network resources properly. It is also helpful to see typical user behavior; then atypical behavior can be seen and addressed. Atypical behavior may indicate a security breach or intrusion or may be an indication of a future device problem.

3.5.3 Security management

Because of the large number of sensor nodes and the broadcast nature of wireless communication, it is usually desirable for BS to broadcast commands and data to sensor nodes. The authenticity of such commands and data is critical for the normal operation of sensor networks. If convinced to accept forged or modified commands or data, sensor nodes may perform unnecessary or incorrect operations and cannot fulfill the intended purposes of the network. Thus, in hostile environments (e.g., battlefield, antiterrorists operations), it is necessary to enable sensor nodes to authenticate broadcast messages received from BSs.

A protocol that can be adopted precisely in the proposed logical topology is SecCOSEN, which has been proposed for authentication, and for establishing secret keys in wireless sensor networks for multi-chain oriented logical topology. SecCOSEN uses partial key pre-distribution and symmetric cryptography techniques. Whereas one version of SecCOSEN protocol uses shared partial keys in a sensor chain, the other version uses private partial keys. Both versions of SecCOSEN show high resilience to different security attacks. The protocol outperforms other random key predistribution protocols in the sense that it requires lower space, lower communication overheads and offers very high session key candidates.

3.6 Performance Evaluation of the Proposed Topology

Several simulation experiments were carried out to evaluate the performance of the logical topology. The proposed logical topology was used for data collection, and its performance was measured against existing data collection protocols, namely LEACH, PEGASIS, and COSEN.

The simulation program was written in object oriented programming language C++. 100 sensor nodes were assumed to be randomly distributed in the target field of $100m\times100m$, and the BS was located at (25, 150). Cartesian coordinates were used to locate the sensor nodes. It was further assumed that each sensor starts with one Joule of initial energy.

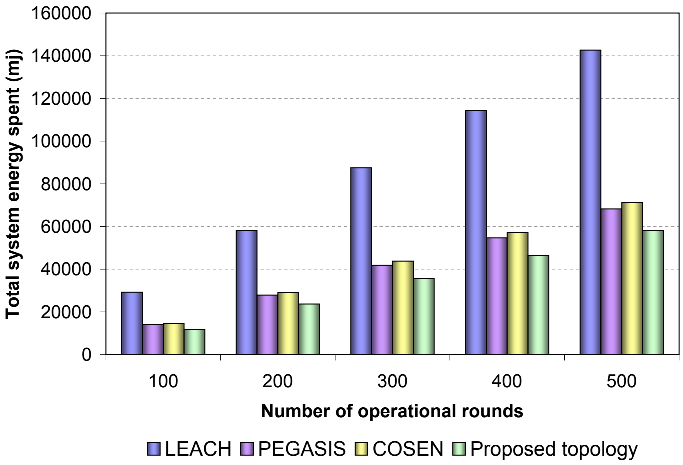


Figure 3.22: Total energy consumption comparison among LEACH, PEGASIS, COSEN and the proposed topology.

In practice it is difficult to model energy expenditure in radio wave propagation. Therefore, in order to measure the energy expenditure in the network, the same simplified radio model used in LEACH and PEGASIS was used. The value of the radio parameters of transmitter and receiver electronic that were used in the simulation are $E_{tx-elec} = E_{Rx-elec} = E_{elec} = 50$ nJ/bit. The value of transmit amplifier (ε) was assumed to be 100 pJ/bit/m^2. It was further assumed that a computation cost of 5 nJ/bit/message to fuse 2000-bit messages. The bandwidth of the channel was set to 1 Mb/s. Thus the total transmission cost for a k-bit message is given by the following equation:

$E_{tx}(k,d) = E_{elec} \times k + \varepsilon \times k \times d^2$

Here d is the distance between sender and receiver measured in meters. In the case of receiving a message, the energy consumption equation is given by the following equation:

$E_{rx}(k) = E_{eleck}$

Multiple runs of the simulation for each protocol were performed and the average value was taken. The metrics that were considered to measure the performance of each protocol are i) overall energy expenditure in the network ii) lifetime of the network, iii) time to complete a fixed number of operational rounds.

The first experiment measured the total energy consumption by the system vary-

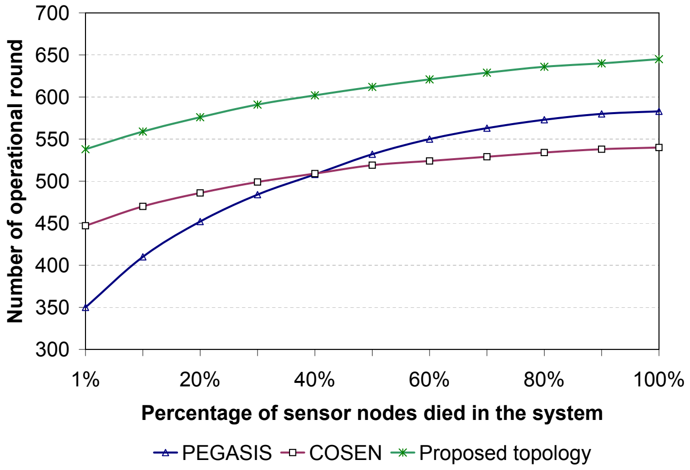


Figure 3.23: Lifetime comparisons among PEGASIS, COSEN, and the proposed topology.

ing the number of operational rounds. Figure 3.22 shows the results. PEGASIS was found to be more energy conservative than LEACH and COSEN. However, the proposed topology outperforms PEGASIS by saving more than 10% of total energy for 500 data collection rounds. This is because of the optimal chain creation by the proposed algorithm, and efficient leader selection processes.

However, conserving the total system energy is not the main achievement of the proposed topology. The main success of the proposed topology is the more even distribution of energy consumption. Uneven energy consumption by the sensor nodes adversely affects the system lifetime. Figure 3.23 demonstrate the lifetime patterns of PEGASIS, COSEN and the proposed topology. The figure shows that the death of the first node in PEGASIS occurs at an early stage compared to COSEN and the proposed topology. For PEGASIS, 10% of the nodes die at around 400 operational rounds, whereas for the proposed topology, 10% of the nodes die at around 550 rounds.

The definitive improvement of the proposed topology over PEGASIS is the latency in data collection. In the simulation, the required amount of time was calculated for different numbers of operational rounds. Figure 3.24 shows the comparison between PEGASIS and the proposed topology in this respect. The pattern of the time requirement graph suggests that PEGASIS is not suitable for large-scale wireless sensor network because of latency. For 100 operational rounds, the proposed topology requires

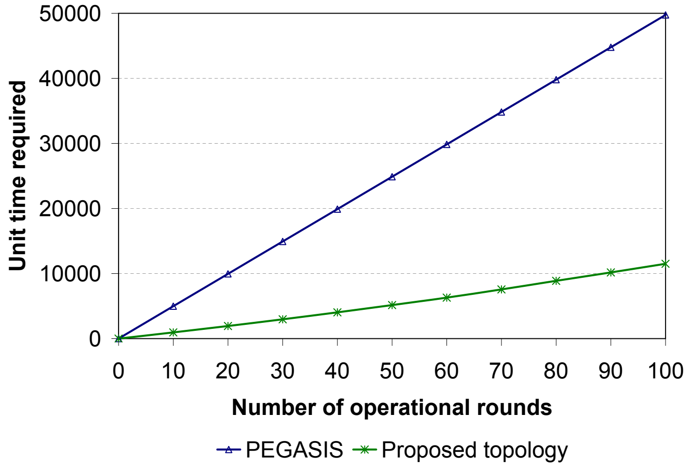


Figure 3.24: Latency comparison between PEGASIS and the proposed topology.

about one-fifth of time required by PEGASIS.

3.7 Summary

This chapter presents a multi-chain oriented logical topology for WSNs. The design of the topology is governed by various factors, such as various resource constraints like energy, time, and computational complexity, networking and architectural factors, network management issues etc. Detailed descriptions of the construction of the proposed topology are provided. Moreover, a three-layer hierarchical management architecture is proposed for the multi-chain oriented topology. The network management scheme works in line with the proposed topology for managing different issues such as fault detection, performance management, security management etc.

The proposed topology entails three phases: topology formation phase, steady state phase, and topology update phase. Whereas the first phase takes place only once during the initial stage, the remaining two phases continue in rotation. Various issues, such as the optimal number of chains in the system, the optimal number of nodes in a chain, the time when the leader nodes need to be changed, and when the chains should be reconstructed etc. are described in detail. The communication abstraction describes the way sensor nodes send and receive different control messages and sensed data.

It is stressed in designing the proposed multi-chain oriented topology that reducing the energy consumption cannot always result in a longer system lifetime. Instead, balancing resources among sensors, and saving energy for those more resource-constrained sensors are very helpful in lengthening the overall system lifetime. Using this principle, the chains were constructed, and the leader nodes were selected.

Simulation results show excellent results in favor of the proposed logical topology. The proposed logical outperforms LEACH, PEGASIS and COSEN not only in total system energy consumption, but also in system lifetime. The key reason behind this is the more even distribution of energy consumption. The proposed topology also solves the high delay problem of PEGASIS.

However, there are still some areas where the performances of the proposed logical topology can be enhanced further. First, assume the applications of WSNs where a number of sensor nodes can be turned off while maintaining the coverage or other user requirements. This node scheduling technique is a prevailing way to save more energy, and thus to prolong network lifetime.

Second, a localized chain creation is the next issue that is considered. Localized chains mean that all the chains are restricted in precise areas such that no chain crosses any other chain. In this way, more interference is avoided. This results in fewer collisions, and thus saves more energy and time.

Third, mobile data collector is the last issue that is considered. The lower-level leaders of the proposed topology are entitled with long distant communication with other leader nodes and/or the BS. Therefore, if these long distant transmissions can be avoided, a vast amount of energy can be saved.

The next three chapters (Chapter 4, 5 and 6) discuss these three adaptations. Various schemes, algorithms and protocols are proposed and designed for these adaptations.

Chapter 4

Chain Member Scheduling

4.1 Preamble

In the last chapter, several adaptations to the basic multi-chain oriented logical topology were discussed. This chapter studies the first adaptation, which is node scheduling. In WSNs, node scheduling techniques have been used extensively to conserve energy consumption [Wu et al., 2005; Wang and Xiao, 2005; Liu et al., 2006; Xiao et al., 2004]. In these techniques, some sensor nodes are put in sleep mode, whereas the other sensor nodes are kept in active mode for sensing and communication tasks. When a sensor node is in sleep mode, it shuts down all functions, except for a low-power timer to wake itself up at a certain time as defined by its node scheduling protocol [Wang and Xiao, 2006]. Therefore, the sensor node consumes only a tiny fraction of the energy, compared to the energy consumed when the sensor node is in active mode all the time [Xu et al., 2000; Feeney and Nilsson, 2001; Bachir et al., 2006]. With node scheduling algorithms, the energy consumption of the network thus becomes efficient, and hence WSNs perform the sensing task for a longer duration of time. The motivations for node scheduling algorithms are further discussed below.

In WSNs, due to the limited resources and vulnerable nature of individual sensor nodes, sensors are deployed with high density (up to 20 nodes/m^3) [Shih et al., 2001]. As a result, the same area is covered by many sensor nodes. This causes heavy redundancy because multiple sensor nodes consume energy to sense the same area, and also to send/receive the identical data. In addition, higher node density incurs more contentions among neighbouring nodes [Kuo et al., 2009]. As a result, additional time slots are required to implement time division multiple access (TDMA) techniques. The solution to avoid this redundancy is to turn off the redundant nodes, because turning off some nodes does not affect the overall system functions as long as there are enough working nodes to provide the services [Tian and Georganas, 2002; Ye et al., 2003]. Turned-off sensor nodes save a significant amount of energy, and this addresses one of the main constraints of WSNs, which is limited energy. Therefore, if sensor nodes are scheduled to perform alternately, more energy can be saved, and the system lifetime is prolonged correspondingly. In addition to redundancy, it is also worth mentioning that not all applications of WSNs require 100% coverage of the target field [Megerian et al., 2005; Wang and Kulkarni, 2006]. 80% to 90% or even a smaller amount of coverage of the target field is adequate. For example, applications, such as tracking humidity or temperature in an area, detecting forest fire etc. do not require 100% coverage by the deployed sensor nodes. It has been shown that sacrificing a little coverage substantially reduces the total energy consumption of the networks [Wang and Kulkarni, 2006] and thus helps to lengthen the lifetime of the network.

On the basis of the aforementioned grounds, a novel node scheduling algorithm is proposed in this chapter. The scheduling algorithm aims to conserve energy by selecting minimal numbers of active nodes to provide the required services. The selected sensor nodes by the scheduling algorithm would be used as the members to create chains for the proposed chain oriented logical topology. This means that, instead of creating chains by all deployed sensor nodes, chains would be created using only the nodes selected by the scheduling algorithm. This process is depicted in Figure 4.1. Figure 4.1(a) shows that several chains are created where all the deployed sensor nodes take part in creating chains. In contrast, Figure 4.1(b) shows that several chains are constructed using only selected nodes, while the rest of the nodes are turned off. Thus the sensor nodes, which do not take part in chain creation process and turn off their functionalities, save a great amount of energy.

The rest of the chapter is organised as follows. Section 4.2 describes the existing algorithms that are related to node scheduling. This section classifies the scheduling algorithms according to their design perspective, analyses different algorithms, and then identifies the requirements for the proposed node scheduling algorithm. Section 4.3 states the problem definition with associated terminologies. Section 4.4

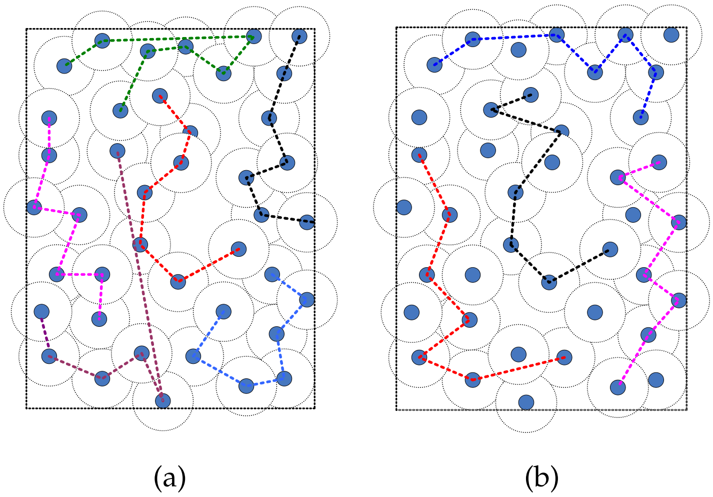


Figure 4.1: Chains are constructed using (a) all deployed sensor nodes; (b) selected sensor nodes.

describes the proposed node scheduling algorithm. In this section, the criteria for selecting the nodes are first identified according to the algorithm requirements. The algorithm to schedule the nodes is then introduced with its detailed descriptions of different calculations, states and transitions. In Section 4.5, a mathematical model is presented. This mathematical model calculates the required number of sensor nodes to attain a certain amount of coverage. To justify the simulation results, which are described in Section 4.6, the results obtained from the mathematical model are matched with the simulation results. Finally, summary of the chapter is provided in Section 4.7.

4.2 Existing Node Scheduling Algorithms

This section examines different approaches that have been used by many researchers to develop node scheduling algorithms for WSNs. The existing algorithms can be classified into two categories on the basis of the designing approaches used in constructing the algorithms. Some algorithms schedule nodes from the communication perspective, whilst others select nodes from the coverage point of view. The following sections describe the existing algorithms in each category.

4.2.1 Communication-based node scheduling algorithms

Node scheduling algorithms that only consider communication, turn off nodes from the communication perspective without considering the system's sensing coverage. Examples of this type of algorithms are found in [Lindsey and Raghavendra, 2002; Xu et al., 2001; J. Pan, 2003; Wu et al., 2005; Cerpa and Estrin, 2004; Xue and Chi, 2007].

SPAN [Chen et al., 2002] is an example of a distributed randomized node scheduling algorithm that conserves energy by turning off the redundant nodes while preserving connectivity. Each node takes a local decision on whether to sleep or join the forwarding backbone, based on an estimation of how many of its neighbours will benefit, and the amount of energy to be saved by this decision. Although this algorithm guarantees connectivity, there is no consideration of sensing coverage.

Xu et al. [2001] propose a scheme in which energy is conserved by letting nodes turn off their communication unit when they are not involved in sending, forwarding or receiving data. Also node density is leveraged to increase the duration of time that the communication unit is powered off. This algorithm, known as Geographical Adaptive Fidelity (GAF), uses geographic location information to divide the area into fixed square grids. Within each grid, it keeps only one node staying awake to forward packets.

Several other protocols, such as ASCENT (Adaptive Self-Configuring sEnsor Topologies) [Cerpa and Estrin, 2004] are also proposed for assuring network connectivity. These approaches perform better when the ratio of communication range to sensing range (Rc/Rs) is less than or equal to one, but as the ratio increases the performance degrades.

4.2.2 Coverage-based node scheduling algorithms

This type of scheduling algorithms considers the coverage only; they do not necessarily ensure network connectivity. For example, algorithms proposed in [Tian and Georganas, 2002; Ye et al., 2003; Xu et al., 2008; Zhang and Hou, 2005; Xin-lian and Bo, 2008; Cho et al., 2007; Mamun et al., 2010c] are examples of the coverage-based node scheduling approach. Brief descriptions for a few of them are given below.

Xu et al. [2008] propose a node scheduling algorithm which ensures long-life and robust sensing coverage. In this algorithm, only a subset of nodes are maintained in working mode to ensure the desired sensing coverage, and other redundant nodes are allowed to fall asleep most of the time. Working nodes continue working until they run out of their energy or until they are destroyed. A sleeping node wakes up occasionally to probe its local neighbourhood, and starts working only if there is no working node within its probing range. Geometrical knowledge is used to derive the relationship between probing range and redundancy. In this algorithm, the authors assume that all nodes have the same sensing ranges to calculate the desired redundancy by choosing their corresponding probing range. However, if nodes have different sensing ranges it is hard to find a relationship between the probing range and the desired redundancy.

Tian and Georganas [2002] propose an algorithm that provides complete coverage using the concept of sponsored area. The authors present a basic model for a coverage-based off-duty eligibility rule and back-off scheme. But the algorithm results in more active nodes because of the imprecise coverage degree calculation. Ye et al. [2003] present a probing-based density control algorithm, named PEAS, which depends on location information to derive redundancy and allows redundant nodes to fall asleep. In the PEAS, some nodes work continuously and die prematurely. This causes the uneven distribution of nodes' energy consumption across the network, reducing the quality of the network coverage. Thus, in PEAS, a sensing hole takes place permanently once it occurs. Furthermore, it may cause partitioning of the network or isolation of nodes. PECAS [Gui and Mohapatra, 2004] is a collaborating adaptive sleeping scheme to improve PEAS. Unlike PEAS, PECAS informs the probing node of the next sleep time of a current working sensor node in the reply message. It allows probing nodes to substitute for the current working node right after the working nodes goes to sleep to reduce the permanent sensing holes.

4.2.3 Requirements for an improved scheduling algorithm

From the abovementioned protocol descriptions, it is apparent that the existing node scheduling protocols treat coverage and connectivity separately. To enjoy the benefits of both communication-based and coverage-based approaches, a node scheduling algorithm should consider both connectivity and coverage while selecting the minimal number of nodes. Few algorithms have also been proposed considering both communication and coverage. For example, Zhao and Gurusamy [2005] propose a connected target coverage algorithm which schedules sensor nodes into multiple sets. Each set maintains both target coverage and connectivity of the network. But this algorithm selects a large number of sensor nodes in total, compared to other node scheduling algorithms.

Moreover, the scheduling algorithms should be aiming to achieve longer lifetime for the network. One basic requirement for maximizing the lifetime of WSNs is to assure even distribution of energy consumption [Shu et al., 2008; Cheng et al., 2008a]. Therefore, the node scheduling algorithm has to be designed to distribute energy consumption properly. In addition, there are a few more requirements for the node scheduling algorithm, which are listed below:

i) Self-configuration of sensor nodes should be mandated because it is inconvenient or impossible to manually configure sensor nodes after they have been deployed in hostile or remote working environments [Mamun et al., 2010a].

ii) The design has to be fully distributed, because a centralized algorithm needs global synchronization overheads, and is not scalable to large populated networks [Zhao and Raychaudhuri, 2009].

iii) The scheduling algorithm should allow the maximum number of nodes to be turned off for most of the time. At the same time, it should preserve the required sensing coverage.

iv) The scheduling scheme should be able to maintain the system reliability. As sensor nodes die at any time in WSNs, a certain amount of redundancy is thus needed to provide the reliability [Miao et al., 2009].

In the proposed approach, each node in the network autonomously and periodically decides itself on whether to turn on or turn off itself using only local neighbours' information. To preserve sensing coverage, each node decides to turn itself off when it discovers that it overlaps a certain amount of its sensing area with its neighbours.

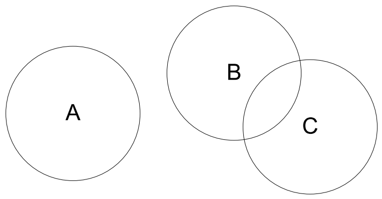


Figure 4.2: Neighbours. Node C is a neighbour of node B, but is not a neighbour of node A.

4.3 Definitions and Problem Statement

Assume a set of sensor nodes $\aleph = \{S_1, S_2, \dots\}$ are randomly deployed on a target field Λ. A scheduling algorithm has to be designed so that it selects a set of sensor nodes, Ω, where $\Omega \subseteq \aleph$. Based on this requirement, this section describes the definitions of necessary terminology for the proposed node scheduling algorithm.

Definition 4.1: Sensing Region. The sensing region of a sensor node S_i, denoted as $C(S_i)$, is the amount of area that is inside the sensing range of the sensor node S_i. To make the calculations simple, it is assumed that the sensing region of a sensor node is represented by a circle, and all sensor nodes have the same sensing ranges. These assumptions can be made without the loss of generality, and are used in many other research works, such as [Wang et al., 2008; Xiao et al., 2010].

Definition 4.2: Neighbour. A node S_j is a neighbour of node S_i, *iff* sensing regions $C(S_i)$ and $C(S_i)$ intersect. Thus, the neighbour set of the node S_i, denoted as $\psi(S_i)$, can be defined as:

$$\psi(S_i) = \left\{S_j | S_j \in \aleph, d(S_i, S_j) < 2r, i \neq j\right\}$$

,

where $d(S_i, S_j)$ denotes the Euclidian distance between the nodes S_i and S_j, and where r is the radius of the sensing region of the nodes S_i and S_j. Figure 4.2 depicts this relationship.

Definition 3: Rank. The rank of a sensor node S_i, denoted as $R(S_i)$, is defined by the cardinality of its neighbour set $\psi(S_i)$. Thus, if the sensor node S_i has a higher number of neighbours than the sensor node S_j, the sensor node S_i's rank has a higher value than that of the sensor node S_j.

Definition 4: Shared sensing region. Shared sensing region of a sensor node S_i, denoted as $\xi(S_i)$, is defined as the fraction of S_i's sensing region, that the sensor node S_i shares with its neighbouring sensor nodes. Thus,

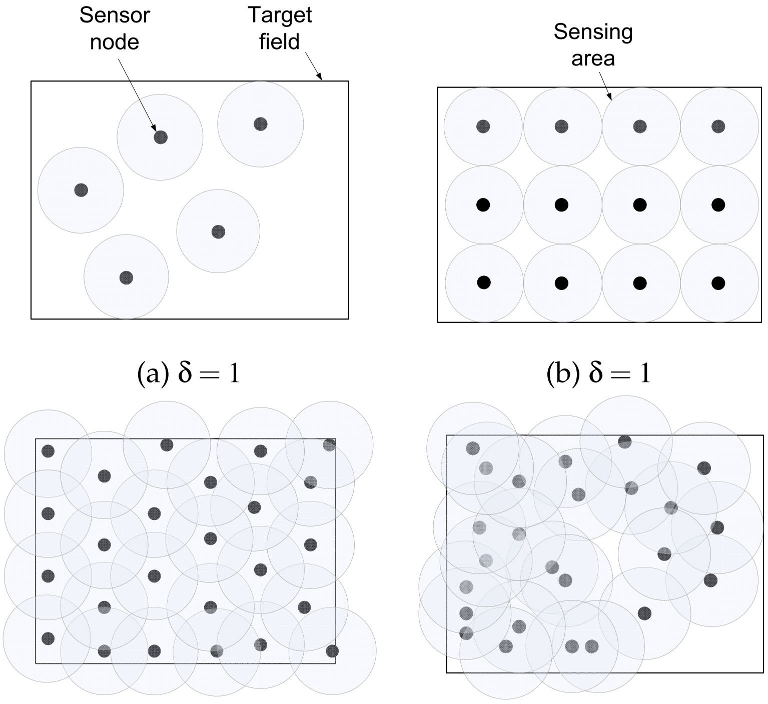
$$\xi(S_i) = \bigcup \{C(S_i) \cap C(S_j) | \forall S_j \in \psi(S_i)\}$$

Definition 5: Deployment density. Deployment density (δ) describes how evenly the sensor nodes are deployed in the target field Λ. Assuming that there are sufficient numbers of sensor nodes to cover the target field, deployment density δ is defined as the ratio between the maximum area that can be covered by the deployed sensor nodes to the actual area covered by the deployed sensor nodes. Deployment density,

$$\delta = \frac{|\aleph|\pi r^2}{\textit{Actual area covered by deployed sensors}} \tag{4.1}$$

Thus, the value of δ is independent from the target field area and is always equal to or greater than 1. When δ equals to 1, it means either that there are not a sufficient number of sensor nodes to cover the target field, or that all the sensor nodes are touching each others' sensing region (which is very unlikely for randomly deployed sensor nodes). These cases are shown in Figure 4.3(a) and Figure 4.3(b) respectively. A uniform placement of a large number of sensor nodes results in a small value of d. On the other hand, if the sensor node placement is not evenly distributed over the target field, the value of d becomes comparatively large. Assuming a target field of $\Lambda = 180m \times 240m$, the radius of sensing region $r = 30m$, the total number of sensor nodes $|\aleph| = 25$, Figure 4.3 illustrates two cases of sensor node placement. In Figure 4.3(c) sensor nodes are evenly distributed and the value of deployment density $d = 1.63$, whereas in Figure 4.3(d), the value of δ is1.96 because sensor nodes in this figure are not evenly distributed.

Definition 6: Coverage Ratio. Denoted by λ, the coverage ratio defines the portion of the sensor field which need to be covered by the selected sensor nodes. Coverage ratio can be calculated by the ratio between the total coverage area by the



(a) $\delta = 1$

(b) $\delta = 1$

(c) $\delta = 1.63$

(d) $\delta = 1.96$

Figure 4.3: Deployment density.

selected sensor nodes to the coverage area by all deployed sensor nodes. Obviously, increasing the coverage ratio makes the coverage quality of the network better.

Definition 7: k-Covered. If a point p is covered by at least k number of sensor nodes, the point p is called k-covered. That is, the point p's coverage degree is k. Coverage degree is used as the measure of quality of coverage service (*QoCS*). Customarily, the higher the coverage degree, the better the coverage quality of the network.

Problem statement

In most relevant works, the problem about k-covered is related to the question of how all points of the target region would be covered by at least k number of sensor nodes. However, for a certain kind of applications, k-covered is not always essential.

For example, some applications do not require every point in the target field to be *k*-covered. This is sufficient to achieve a certain coverage ratio. For example, 80%-90% coverage ratio, or even less is adequate for a WSN to estimate air pressure, temperature, humidity or to detect an event like forest fire. Moreover, when sensor nodes are deployed randomly in a target field, the sensor nodes may not even cover 100% of the target area. Based on this, a novel problem of *QoCS* of 1-covered with λ% coverage ratio is proposed. This thesis defines the node scheduling problem as follows: Given the deployment density δ, the question is to find a minimal number of nodes such that the coverage ratio is at least λ% of the target network.

4.4 Description of the Proposed Algorithm

This section describes the proposed node scheduling algorithm in detail. The section consists of several sub-sections which describe different issues, methods and calculations for the proposed node scheduling algorithm. First of all, in 4.4.1, the design basis of the algorithm is identified. 4.4.2 then describes the criteria for selecting each sensor node. Based on these criteria, necessary rules for scheduling sensor nodes are defined in 4.4.3. The proposed node scheduling technique based on the scheduling rules are described in 4.4.4. This sub-section also discusses the related calculations for the scheduling algorithm, and further presents the total node scheduling algorithm in pseudo-code. A description of how to adopt the proposed scheduling algorithm with the proposed logical topology is presented in 4.4.5. 4.4.6 then presents different states and their transitions in the proposed node scheduling algorithm.

4.4.1 Designing consideration of the proposed node scheduling algorithm

As discussed in Section 4.2, the design of a node scheduling algorithm should be based both on the coverage and connectivity. However, in a situation where one of these attributes is guaranteed, the node scheduling algorithm usually exploits the other attribute. In this thesis, to design an improved node scheduling algorithm, coverage was chosen as the primary design basis. The following arguments are made to justify focusing on only one of the two attributes.

i) This node scheduling scheme is a part of the proposed logical topology design process. The proposed logical topology assures the connectivity problem, and this has already been discussed in Chapter 3. Thus it is sufficient for the proposed node scheduling algorithm to concentrate only on coverage.

ii) Xing et al. [2005] prove that if the radius of the transmission range of each sensor node is at least double the radius of its sensing ranges, a WSN is connected, provided that its sensing coverage is guaranteed. Note that one of the primary assumptions of this thesis is that transmission ranges of the sensors are much greater than their sensing ranges. This assumption can be made with loss of generality, and this assumption is used in many research works such as [Tian and Georganas, 2002; Ye et al., 2003; Xu et al., 2008; Zhang and Hou, 2005; Xin-lian and Bo, 2008; Lindsey and Raghavendra, 2002; Heinzelman et al., 2000].

iii) Moreover, coverage-preserved node scheduling has been found as an efficient way to prolong system lifetime [Mao et al., 2008; Ma et al., 2004; Tian and Georganas, 2004].

Based on these arguments, this thesis focuses on coverage as the basis for designing a node scheduling algorithm. After defining the design basis, the next step for the proposed node scheduling algorithm is to identify the criteria which would be used to select each sensor node. The following sub-section discusses the node selection criteria.

4.4.2 Identifying node selection criteria

In the proposed node scheduling algorithm, four specific criteria have been considered. Based on these criteria, the priority of each node is defined. For each sensor node, these criteria are: number of neighbours of the node, the node's shared sensing region with its neighbours, residual energy of the sensor node and repeated selection number of the node (i.e., number of times the node was selected earlier). The justification for these criteria are described below.

The first criterion that should be chosen for the scheduling algorithm is the number of neighbours of each node. If a node does not have any neighbouring node at all, this node must be selected. Otherwise, the sensor node's sensing region cannot be sensed by any other sensor node(s). On the other hand, if a node is surrounded by many other

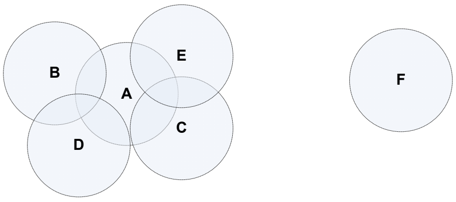


Figure 4.4: Illustration of prioritizing a node using a number of neighbours.

sensor nodes, that node's coverage area can be sensed by the node's neighbouring nodes. Thus the node with many neighbours can be turned off. Figure 4.4 depicts this situation. In this figure, node A is surrounded by its neighbouring nodes B, C, D, and E. On the other hand, node F has no neighbouring node. Thus no other sensor, except the node F, is able to monitor any particular point inside the sensing region of F, $(C(F))$. Thus, there is no other option but to select the sensor node F.

The second criterion should be the shared sensing region ($\xi(x)$) of each sensor node with its neighbouring nodes. For any two sensor nodes S_i and S_j, the relation $\xi(S_i) > \xi(S_j)$ means that the sensor node S_i shares a comparatively larger area with its neighbouring sensor node(s) than the sensor node S_j does. Note that, the value $\xi(x)$ does not depend on the number of neighbours. For example, in Figure 4.5, the sensor node A has four neighbours, whereas the sensor node M has three neighbours. However, the sensor node M shares a larger area ($\xi(M)$) with its neighbouring sensor nodes N, P and Q, compared to the sensor node A, which shares its sensing region ($\xi(A)$) with its neighbours B, C, D and E. Thus, if the sensing region of a node overlaps a large amount of area with its neighbours, the node can be replaced by one of its neighbouring nodes. As a result, sensor nodes which share small areas with their neighbouring nodes should have higher priority to be selected.

The third criterion to be considered during node selection is the residual energy of the sensor nodes. A sensor node which has lost a considerable amount of its battery energy should be avoided, unless there is no other way but to select the node. Selecting such a node accelerates the death of the sensor node, and this negatively affects the network lifetime [Shu et al., 2008; Cheng et al., 2008b].

The fourth and last criterion should be the repetition of selection of a sensor node. If the same sensor node is selected over and over again, the node loses its energy very

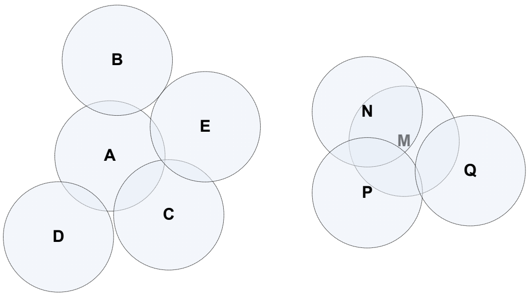


Figure 4.5: Illustration of prioritizing a node using shared sensing regions.

quickly, and this situation adversely affects the lifetime of the network [Shu et al., 2008; Cheng et al., 2008b]. Thus, the node scheduling algorithm should be fair for each sensor, so that each sensor is selected at least one time in a specific period of time.

After determining the node selection criteria, the next step for the proposed node scheduling algorithm is to construct specific rules that the algorithm would follow in order to select appropriate sensor nodes. The following sub-section uses the criteria discussed in this section to construct the required set of rules.

4.4.3 Node scheduling rules

The proposed node scheduling algorithm would follow a set of rules to schedule the sensor nodes. These rules are derived using the selection criteria (as discussed in 4.4.2), and to meet the algorithm requirements (as discussed in 4.2.3). These rules specify which node is to be selected, which should not be selected, and which should be prioritized. The node scheduling rules are as follows:

i) To make the node scheduling algorithm distributed and independent from locations of sensor nodes, each node should autonomously and periodically decide whether to go to sleep mode, or to keep itself active. In making this decision, each node would consider the following issues: residual energy of the node, number of its neighbouring nodes and number of times the node was selected previously.

ii) A sensor node with a higher level of energy holds a higher chance of being selected than a sensor node with a lower energy level. Otherwise, energy consumption throughout the network would not be evenly distributed.

iii) A sensor node with a lower rank should be prioritized to be selected, compared to those with higher ranks, because a high-ranked sensor node has a higher possibility to be redundant than a low-ranked sensor node.

iv) A sensor node which shares a comparatively smaller area of its sensing region with its neighbouring sensor nodes holds higher priority to be selected.

v) Among deployed sensor nodes, a set of sensor nodes are selected by the scheduling algorithm to ensure minimum $\lambda\%$ of coverage by the selected sensor nodes. On the other hand, as soon as desired $\lambda\%$ of coverage is achieved, the node scheduling algorithm stops selecting any further sensor node.

After establishing the rules for scheduling sensor nodes, the next task is to apply these rules for each sensor node. The methods of applying the node scheduling rules for each sensor network are described in the following sub-section.

4.4.4 Applying node scheduling rules to select sensor nodes

The main idea for scheduling sensor nodes is to use the redundancy in sensing regions, and to offer the user to select the coverage ratio (λ) necessary for the specific application. Depending on the value of coverage ratio λ and deployed density δ, the proposed node scheduling algorithm is able to determine the minimum number of sensor nodes required to achieve the coverage ratio λ. Different steps involved in the node scheduling algorithm are described below.

In the proposed node scheduling algorithm, each sensor node makes its own decision depending on the information it collects from its neighbouring nodes. This decision is made by each sensor node at the start of the node scheduling algorithm. To make this decision, each sensor node generates a *pseudorandom number*[1], using the seed state that includes two pieces of information, which are i) residual energy of the node, and ii) number of times the node was previously selected. The sensor node then informs this pseudorandom number to all of its neighbours using a *'hello'* message. If the generated *pseudorandom number* is less than a threshold value, the node decides

[1] A pseudorandom number is a number that appears to be random but is not. Pseudorandom sequences typically exhibit statistical randomness while being generated by an entirely deterministic causal process. Any pseudorandom number formula depends on the seed value to start the sequence.

to take part in the scheduling process. The node then informs its willingness to join the scheduling process by sending *'notify'* messages to all of its neighbouring sensor nodes. On the other hand, if the generated pseudorandom number is greater than the threshold value, the sensor node does not do anything.

A sensor node that is not participating in the scheduling process, discards any *'notify'* messages from its neighbouring sensor nodes. On the other hand, sensor nodes participating in the scheduling process collect all *'notify'* messages from their neighbouring sensor nodes. From the collected *'notify'* messages, each participating sensor node calculates two parameters, namely i) its rank and ii) the shared sensing ranges with its neighbours. These two parameters would be used in the scheduling process in the following way.

The rank of a sensor node is defined by the cardinality of its neighbour set (see Section 4.3). For example, if a sensor node does not have any neighbour, its rank is zero; if there is a single neighbour, the rank of the node is one, and so on. According to the node scheduling rules, a sensor node with a lower rank enjoys higher priority to be selected than a sensor node with higher rank, and vice versa. Thus node selection procedure starts with the sensors with lower ranks. The sensors with rank zero are considered first; after that sensors with rank one are considered, and so on.

To consider whether a sensor node S_i is to be selected or not, its shared sensing region ($\xi(S_i)$) with the currently selected neighbouring sensor nodes is calculated. This is because if a sensor node is not selected by the scheduling algorithm, there is no point to counting the sensor node as a neighbour. For clarification, consider the sensor node S_i has four neighbouring sensor nodes, S_m, S_n, S_o and S_p. Among the neighbours, for example, the sensor nodes S_m and S_o have already been selected. While the sensor node S_i would determine whether or not it would be selected, the sensor node S_i calculates $\xi(S_i)$. In calculating $\xi(S_i)$, the sensor node S_i considers only two of its selected neighbouring nodes S_m and S_o. In other words, the sensor node S_i does not consider the sensor nodes S_n and S_p in calculating $\xi(S_i)$.

For the sensor node S_i, after calculating $\xi(S_i)$, this value is compared with a threshold value ξ_{max}. In this proposed node scheduling algorithm, this threshold value is called *'maximum allowed shared sensing region'*. The sensor node is selected if $\xi(S_i) \leq$

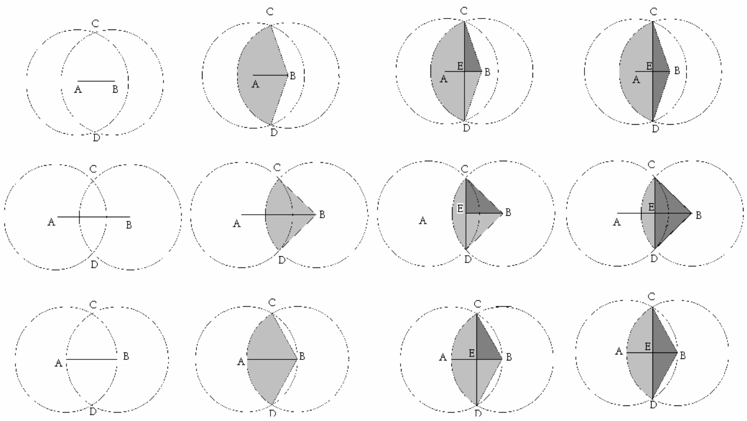


Figure 4.6: Calculating shared sensing region by two nodes.

ξ_{max}, otherwise the sensor node S_i is turned off.

The following two sub-sections describe how a sensor node S_i calculates $\xi(S_i)$ and how ξ_{max} value is estimated. Note that, as it is assumed that the sensors are deployed randomly in the target field, it is not feasible to provide a fixed valued for ξ_{max}.

4.4.4.1 Shared sensing region calculation

Assume the sensing range of a sensor node S_i is r. By definition, a node's sensing region is a circle centred at this node with radius r, if all nodes lie on a 2-dimensional plane. To simplify the calculation, consider only two nodes that share their sensing regions.

Figure 4.6 shows the calculation of shared sensing region of two sensor nodes, S_i and S_j, step by step, from left column to right column, for three different situations: $0 < d < r$, $r < d < 2r$ and $d = r$, where d is the Euclidian distance between two sensor nodes ($AB = d$), and r is the radius of the sensing region (d cannot be greater than $2r$, otherwise they are not considered as neighbours). Let C and D be the two intersecting points by the sensing areas of two nodes and angle $\angle CBD = \alpha$.

Now, $\triangle BCD = \frac{dr\sin(\alpha/2)}{2}$ and $\alpha = 2\arccos(d/2r)$

So, half of the shared area of two circles = $\frac{r^2\alpha}{2} - \frac{dr\sin(\alpha/2)}{2}$

Thus, the shared area by the two sensor nodes S_i and S_j

$$\xi(S_i, S_j) = r^2\alpha - dr\sin(\alpha/2) \; = 2r^2\arccos(\alpha/2) - dr\sin(\arccos(d/2r)) \qquad (4.2)$$

Using equation 4.2, the sensor node S_i can easily find out the shared sensing area if the sensor node S_i has only a single neighbour. However, in most of the cases, a sensor node has more than one neighbour. Let a sensor node S_i has m number of neighbours. $(\psi(S_i) = \{S_1, S_2, \ldots, S_m\})$. Further assume, $\xi(S_i, S_1) = A_1, \xi(S_i, S_2) = A_2, \ldots, \xi(S_i, S_m) = A_m$. Without loss of generality, it can be assumed that a higher number of neighbours produce higher probability to coincide the sensor node S_i's shared areas with its neighbouring sensor nodes. Also assume, $d(S_i, S_1) < d(S_i, S_2)$, $d(S_i, S_2) < d(S_i, S_3)$, and so on (i.e., S_1 is the closest neighbour of S_i whereas S_m is the furthest neighbour of S_i). Thus, $A_1 > A_2 > A_3 \ldots > A_m$.

To calculate the shared sensing region of S_i with its neighbour nodes, the contributions of each sensor node (from the closest to the furthest) would be considered. Essentially, the closest neighbour contributes the most.

Thus, considering the first neighbour S_1, $\xi(S_i) = A_1$.

Considering the second neighbour S_2, $\xi(S_i) = A_1 + A_2 - (A_1 \bigcap A_2)$.

Considering the third neighbour S_3, $\xi(S_i) = A_1 + A_2 + A_3 - (A_1 \bigcap A_2 + A_2 \bigcap A_3 + A_3 \bigcap A_1 + A_1 \bigcap A_2 \bigcap A_3)$.

$\vdots$

and so on.

Now, finding the value of common areas of shared regions(such as $A_1 \bigcap A_2 \bigcap A_3$) is not trivial. As the number of neighbours increases, the complexity if the computation increases exponentially. This computation may not be suitable for resource-constrainted sensor nodes. For this reason, a heuristic approach was adopted. This approach is described below.

Consider the calculation of $A_1 \bigcap A_2$.Two extreme cases can be assumed, where in one extreme case, A_1 and A_2 would be disjoint. In this case the contribution of the second neighbour to $\xi(S_i)$ is the whole shared area A_2. On other extreme case, the shared area of the second neighbour A_2 would be fully covered by the A_1. In this case the contribution of the second neigbour to $\xi(S_i)$ is 0. The huristic followed in this case is to take the average of the two extreme cases. Thus, using this huristic, $A_1 \bigcap A_2 = \frac{A_2}{2}$.

Therefore, after considering the second neighbour S_2, $\xi(S_i) = A_1 + A_2 - (A_1 \bigcap A_2) \approx A_1 + \frac{A_2}{2}$.

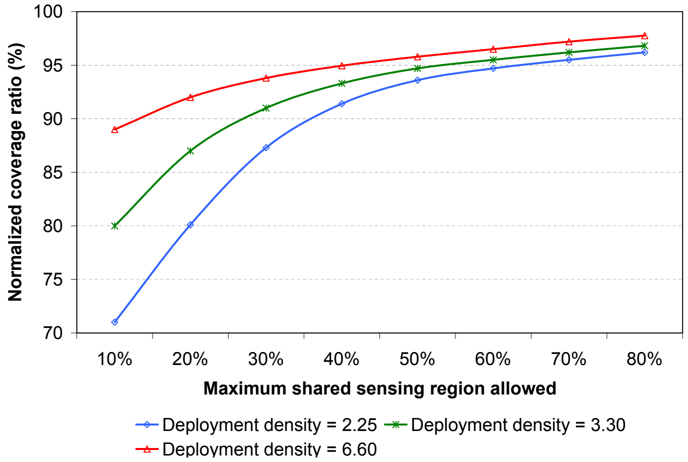


Figure 4.7: ξ_{max} Vs. λ.

Now, applying the hurustics, $A_1 \bigcap A_3 + A_2 \bigcap A_3 - A_1 \bigcap A_2 \bigcap A_3 \approx \frac{A_3}{2} + \frac{A_3}{2} - \frac{A_3}{3}$.

Therefore, after considering the third neighbour S_3, $\xi(S_i) = A_1 + A_2 + A_3 - (A_1 \bigcap A_2 + A_2 \bigcap A_3 + A_3 \bigcap A_1 + A_1 \bigcap A_2 \bigcap A_3) \approx A_1 + \frac{A_2}{2} + \frac{A_3}{3}$

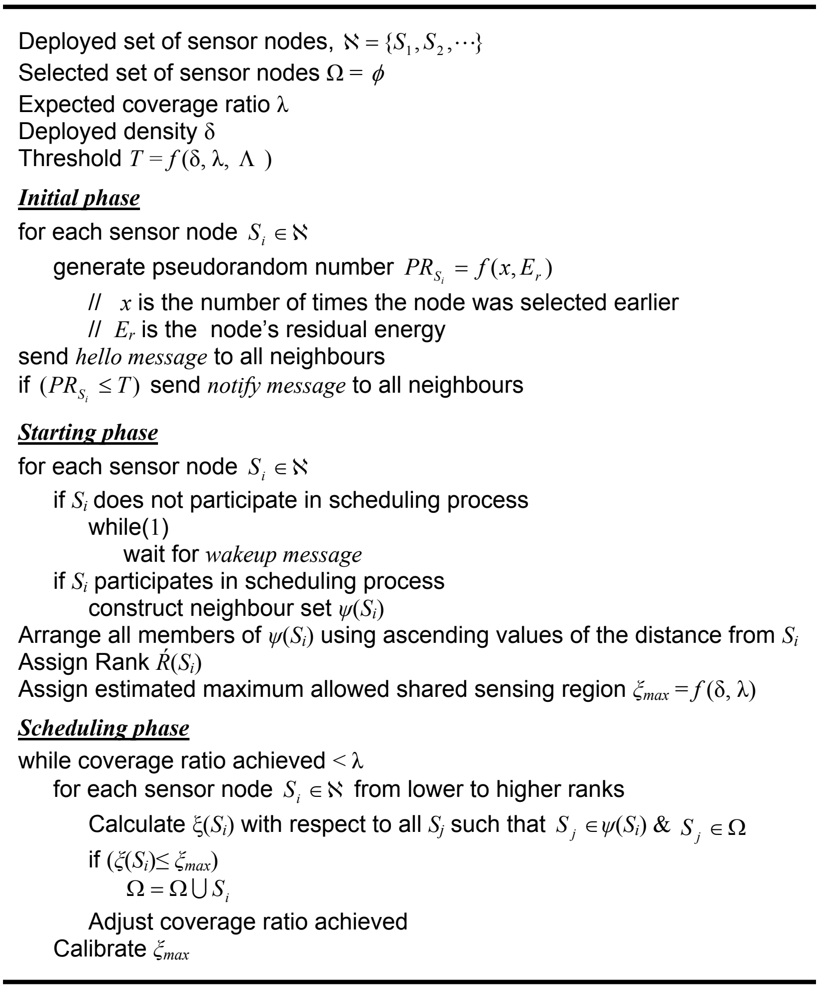
Thus, it can be shown that the total shared sensing region of S_i with its neighbours,

$$\xi(S_i) \approx \xi(S_i, S_1) + \frac{1}{2}\xi(S_i, S_2) + \frac{1}{3}\xi(S_i, S_3) + \cdots + \frac{1}{m}\xi(S_i, S_m) = \sum_{j=1}^{m} \frac{\xi(S_i, S_j)}{j} \tag{4.3}$$

Equation (4.3) is used to calculate the total shared sensing region of a sensor node Si in respect to its neighbouring nodes which have already been selected. If the calculated value of $\xi(S_i)$ is less than or equal to ξ_{max}, the sensor node S_i is selected, otherwise it is turned off. How to estimate the value of ξ_{max} is shown below.

4.4.4.2 Estimate maximum shared sensing region (ξ_{max}) value

Selecting the value for ξ_{max} is crucial in this proposed node scheduling algorithm. If ξ_{max} value is too small, the scheduling process would require a longer period of time. On the other hand, choosing a very large ξ_{max} actually diminishes the efficiency of the algorithm. Also note that random deployment of the sensor nodes in the target field precludes a pre-determined value of ξ_{max} . The value of ξ_{max} mainly depends on deployed density (δ) of the sensor nodes. Deployed density (δ) is calculated using the definition discussed in Section 4.3. Figure 4.7 shows the relationship among deployed



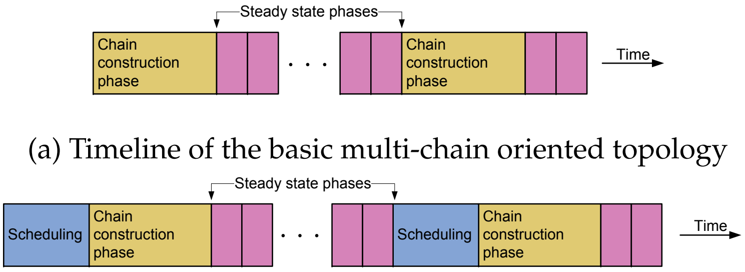
```
Deployed set of sensor nodes, $\aleph = \{S_1, S_2, \cdots\}$
Selected set of sensor nodes $\Omega = \phi$
Expected coverage ratio $\lambda$
Deployed density $\delta$
Threshold $T = f(\delta, \lambda, \Lambda)$
Initial phase
for each sensor node $S_i \in \aleph$
    generate pseudorandom number $PR_{S_i} = f(x, E_r)$
        // $x$ is the number of times the node was selected earlier
        // $E_r$ is the node's residual energy
send hello message to all neighbours
if $(PR_{S_i} \leq T)$ send notify message to all neighbours
Starting phase
for each sensor node $S_i \in \aleph$
    if $S_i$ does not participate in scheduling process
        while(1)
            wait for wakeup message
    if $S_i$ participates in scheduling process
        construct neighbour set $\psi(S_i)$
Arrange all members of $\psi(S_i)$ using ascending values of the distance from $S_i$
Assign Rank $\acute{R}(S_i)$
Assign estimated maximum allowed shared sensing region $\xi_{max} = f(\delta, \lambda)$
Scheduling phase
while coverage ratio achieved < $\lambda$
    for each sensor node $S_i \in \aleph$ from lower to higher ranks
        Calculate $\xi(S_i)$ with respect to all $S_j$ such that $S_j \in \psi(S_i)$ & $S_j \in \Omega$
        if $(\xi(S_i) \leq \xi_{max})$
            $\Omega = \Omega \cup S_i$
        Adjust coverage ratio achieved
    Calibrate $\xi_{max}$
```

Figure 4.8: Chain member scheduling algorithm.

density (δ), maximum shared sensing region allowed (ξ_{max}) and normalized coverage ratio (λ). For example, if the deployment density is 3.3, and the user requires a coverage ratio $\lambda = 90\%$, the algorithm starts with the value of maximum shared sensing region $\xi_{max} = 25\%$. One point to note here is that, as it is assumed that the sensor nodes are deployed randomly over the target region, it is not possible to calculate the exact value of ξ_{max} beforehand. Using this algorithm, however, the value of ξ_{max} can be estimated as closely as possible.

Using these estimations and calculations, sensor nodes from the target field are selected. The total node scheduling algorithm is shown in Figure 4.8.

The following two sub-sections discuss two important issues: i) how to adopt the



(a) Timeline of the basic multi-chain oriented topology

(b) Timeline of basic topology with node scheduling extension

Figure 4.9: Adoptation of the proposed node scheduling algorithm.

node scheduling algorithm in the existing logical topology, which was described in Chapter 3, and ii) different states of sensor nodes during the node scheduling algorithm run and the transitions of the sensor nodes from one state to another.

4.4.5 Adoption of node scheduling algorithm

To extend the basic logical topology (that is described in Chapter 3) with the node scheduling scheme, a straightforward way is to insert the self-scheduling phase of the proposed scheme before the chain formation phase of the basic logical topology. The advantage of such a timeline is that the proposed node-scheduling scheme is embedded into the basic logical topology seamlessly without any modification of its original workflow. The timeline of the implementation is illustrated in Figure 4.9. Figure 4.9(a) shows the original workflow of the proposed logical topology, which was described in Chapter 3. By 'Scheduling' in Figure 4.9(b) the placement of the proposed node scheduling algorithm is shown. Note that the adopting the node scheduling algorithm can be embedded very easily. If the node scheduling algorithm is embedded in the proposed logical topology design, the algorithm would be initiated in each chain construction round.

4.4.6 Scheduling states and transitions

This sub-section describes the different states and their transitions during the sensor network runs using our logical topology. In Chapter 3, it is described that reconstruction of a chain is required when around 20% of its member nodes die. The node scheduling scheme aims to engage only a subset of the deployed nodes in the field to

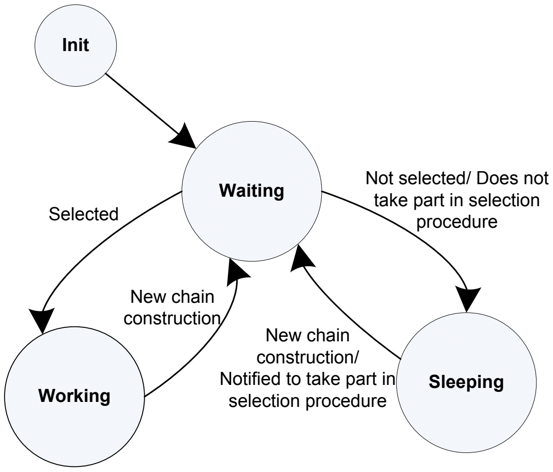

Init
Waiting
Selected
Not selected/ Does not take part in selection procedure
New chain construction
New chain construction/ Notified to take part in selection procedure
Working
Sleeping

Figure 4.10: States and transitions of the proposed node scheduling algorithm.

construct chains. Intuitively, this leaves an option to change the member nodes of a chain more frequently. This helps energy dissipation by the sensor nodes to be more even, and thus increases the lifetime of the network.

In the proposed node scheduling algorithm, all the nodes stay in one of the three states: i) waiting state ii) sleeping state and iii) working state. Figure 4.10 shows the states and their transitions.

At the very initial stage (just after the sensor deployment), or after the end of each chain construction round, all the nodes are in waiting state. Each node waits for a random back-off time (to avoid collisions), and then broadcasts a *'hello'* message. It is used for a node to collect the pseudorandom numbers generated by its neighbour nodes. Each node maintains a neighbour table and refreshes it periodically. Maintaining the pseudorandom numbers of neighbours is worthwhile when a sensor node in sleeping state has to take part in the scheduling procedure to make up coverage ratio. In other words, in waiting state, a node broadcasts a *'hello'* message. It then makes decision whether or not to take part in the scheduling procedure, and then notifies its intension sending a *'notify'* message. When a node does not take part in the scheduling procedure, it goes to sleeping mode directly without notifying its neighbours. On the other hand, if a node takes part in the scheduling procedure and is selected, it enters the working state, otherwise it goes to sleeping state. At the end of a new chain construction round, all nodes come back to waiting state. In working state, a node actively monitors the area and takes part in communication along the chain, as described in Chapter 3. A node remains in working state until the beginning of a chain construction round. It is assumed that when a node fails, it simply stops working and does not send or receive any messages.

4.5 Mathematical Analysis

In this section, a mathematical model is used to calculate the number of sensor nodes required to achieve a fixed coverage ratio. The aim to present this mathematical model is to validate the simulation results. The results from this mathematical model will be matched with the simulation results and compared.

Let the target field Λ has an area α^2. Further, assume $q \subseteq \Lambda$ be the part of the target field Λ, which will be covered by the circular sensing ranges of k number of sensor nodes residing in the target field. Then, the fraction $\frac{area(q)}{\alpha^2}$, where $area(q)$ denotes area of q, is the user's desired sensing coverage at each reporting round. Any point $(x,y) \in \Lambda$ is considered to be covered if it is inside the circular sensing coverage of a selected sensor node in the target field. To measure the probabilistic sensing coverage, the probability of a point $(x,y) \in \Lambda$ not to be covered by a selected sensor node S_i, $P(1)^i_q(x,y)$is measured. Let (u,v) be the location of Sensor S_i, and $A(x,y)$ be a circular area centred at point (x,y) with radius r. Then, the point will not be covered when $(u,v) \in \Lambda - A(x,y)$. Therefore, the probability of the point (x,y) not to be covered by a randomly selected sensor node, is given by:

$$P(1)^i_q(x,y) = \int_{(u,v)\in\Lambda - A(x,y)} \int \phi(u,v)dudv \tag{4.4}$$

where $\phi(u,v) = \frac{1}{\alpha^2}$ is the probability of S_i to be located on a point $(x,y) \in \Lambda$. Equation (4.4) represents the fraction of Λ not covered by a randomly-selected sensor node's circular sensing range. Thus, the probability of a point not covered by randomly selected k sensor nodes is obtained as:

$$P(k)_q(x,y) = \prod_{i=1}^{k} \left(P(1)^i_q(x,y)\right)^i \tag{4.5}$$

Let $\overline{q}$ be the area that is not covered. For randomly selected k sensor nodes, the expected value of $\overline{q}$ can be given by:

$$E\left[\overline{q}\right] = \int_{\Lambda} \int P(k)_q(x,y)dxdy \tag{4.6}$$

Now, consider how much area in Λ can be covered by randomly-selected k sensor nodes. For this purpose, consider the fraction of Λ not covered by these k sensor nodes within Λ. This can be obtained by dividing $E\left[\overline{q}\right]$ (Equation 4.6) by the area of Λ, α^2 assuming all (x,y) points are uniformly distributed over Λ. Using Equations (4.4) and (4.6), the fraction of Λ not covered by k selected sensor nodes, is given as:

$$E\left[\frac{\overline{q}}{\alpha^2}\right] = \left(\frac{\Lambda - A(x,y)}{\Lambda}\right)^k = \left(\frac{\alpha^2 - \pi r^2}{\alpha^2}\right)^k \tag{4.7}$$

Finally, when k sensor nodes are randomly selected from Λ, the probabilistic sensing coverage that any point of Λ will be covered by at least one of k selected sensor nodes' circular sensing range is equivalent to the desired coverage ratio λ. Thus,

$$\lambda = 1 - E\left[\frac{\overline{q}}{\alpha^2}\right] = 1 - \left(\frac{\alpha^2 - \pi r^2}{\alpha^2}\right)^k \tag{4.8}$$

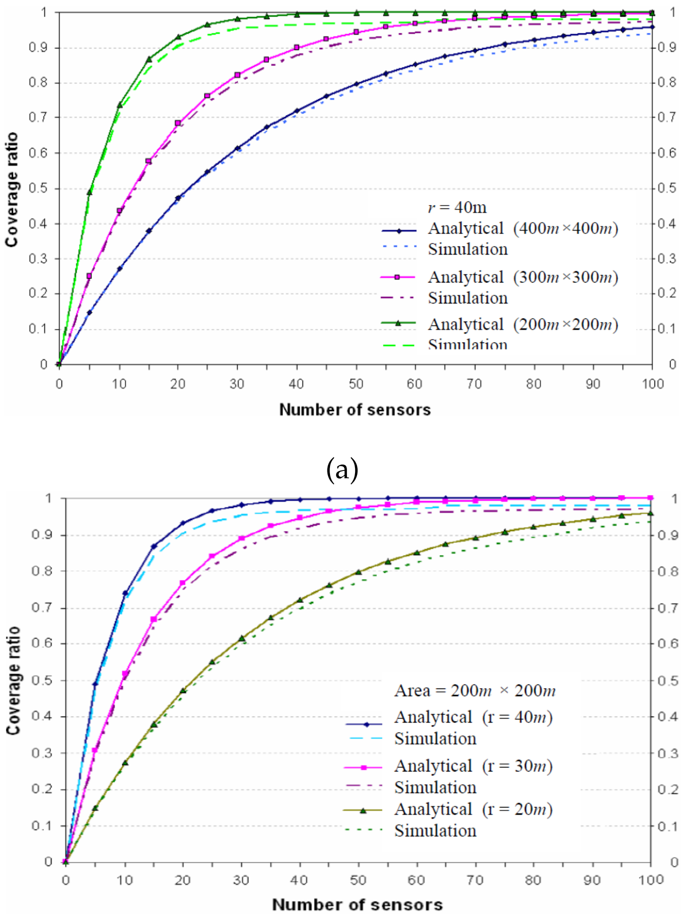
Therefore, the smallest integer k which meets the desired sensing coverage, λ, can be defined as:

$$k = \left\lceil \frac{\log(1-\lambda)}{\log\left(\frac{\alpha^2 - \pi r^2}{\alpha^2}\right)} \right\rceil \tag{4.9}$$

In order to verify the correctness of k, the analytical model is simulated, and the simulation results are compared with the numerical results measured from Equation (4.8). Figures 4.11(a) and (b) show the comparison of the results in covering a requested portion of the monitored area with varying network sizes and sensor nodes' circular sensing ranges. The simulation results shown in each plot correspond to the average of 100 simulation runs. Regardless of the sizes of the network and sensing range, it can be observed in Figures 4.11(a) and (b) that both the numerical and simulation results are found to match well.

4.6 Experimental Results

This section evaluates the performance of the proposed node scheduling algorithm based on various experimental results. The proposed node scheduling algorithm is compared with a method which selects a same number of sensor nodes using uniform distribution. Figure 4.12 shows the comparison of the scheduling algorithm with randomly chosen nodes from uniformly distributed nodes. For example, to achieve the



(a)

(b)

Figure 4.11: Comparison of simulation and analytical results for covering a target field.

coverage ratio $\lambda = 80\%$ while uniform distribution method needs around 50 nodes, the proposed algorithm needs to select only 35 nodes to produce the same coverage ratio. Because in the proposed algorithm, nodes are selected on the basis of shared sensing regions, the selected nodes effectively produce better coverage ratio.

The next experiment measures the energy consumption of the nodes using the proposed node scheduling algorithm. For the purpose of energy consumption calculation, energy expenditure model is required.

In practice, it is difficult to model energy expenditure in radio wave propagation. Therefore, in order to measure the energy expenditure in the network, the simplified

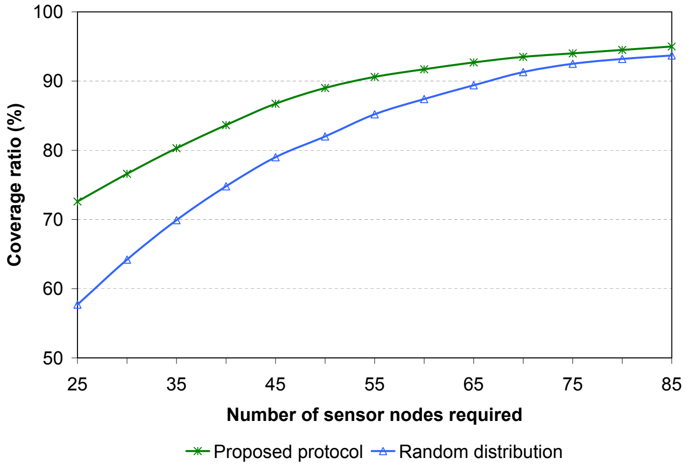


Figure 4.12: Comparison of proposed node scheduling algorithm with a method which chooses nodes from random distribution. ($|\aleph| = 100, A = 400m \times 400m, r = 40m$).

radio model, which is used in [Heinzelman et al., 2000; Lindsey and Raghavendra, 2002], is assumed. The model uses the first order radio model. In this model it is assumed that the sources of energy dissipation are the transmitters which dissipate energy to run radio electronics and power amplifier. Another source of energy dissipation is assumed as the receivers which dissipate energy to run the radio electronics. In [Heinzelman et al., 2000; Lindsey and Raghavendra, 2002], it is approximated that the transmitter amplifier requires $E_{amp} = 100\text{pJ/bit}/m^2$ to amplify the signal at an acceptable signal to noise ratio (SNR). In addition, energy required in running transmitter and receiver electronics are equal and given by $E_{tx-elec} = E_{rx-elec} = E_{elec} = 50\text{nJ/bit}$. Moreover, the energy cost for data aggregation is considered as 5nJ/bit/message [Li and Halpern, 2001]. The bandwidth of the channel is set to 1 Mb/s [Kulik et al., 2002]. In the experiments, it was assumed that each data message is 2000 bits long and information processing time in a node was between 5 to 10 milliseconds [Kulik et al., 2002]. The medium is assumed to be symmetric, so that the energy required for transmitting a message from node A to B and from B to A are the same at a fixed signal to noise ratio (SNR). So it can be said that, for free space propagation loss, energy dissipation is certainly dominated by the long distance transmissions. Thus, the total transmission cost for a k-bit message is given by the equation 4.10. In case of receiving a message,

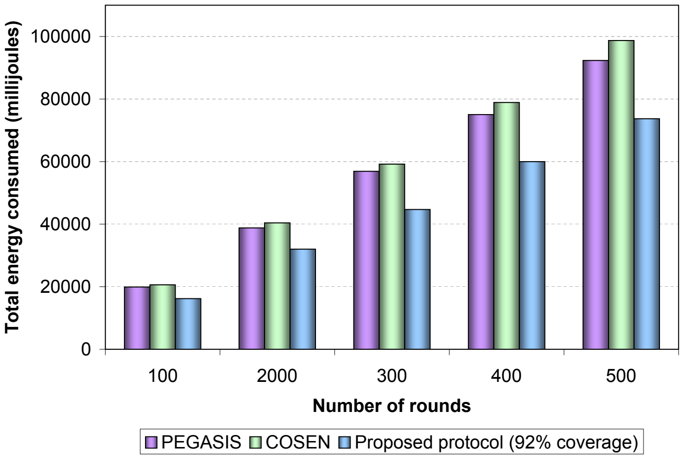


Figure 4.13: Energy dissipation comparison among PEGASIS, COSEN and proposed protocol. ($|\aleph| = 100, A = 50m \times 50m, r = 10m, \lambda = 92\%, \delta = 2.7$).

the energy consumption equation is given by equation 4.11.

$$E_{tx}(k,d) = E_{elec} \times k + E_{amp} \times k \times d^2 \tag{4.10}$$

$$E_{rx}(k) = E_{elec} \times k \tag{4.11}$$

where d is the distance between the sender and the receiver measured in meters.

Figure 4.13 shows the comparative energy consumption of the proposed method with that of COSEN and PEGASIS. PEGASIS is chosen in this case, because PEGASIS is also a chain oriented algorithm which acts like COSEN, except that it uses a single chain. To compare energy consumption, a large value of $\lambda = 92\%$ is chosen. In the experiments it was found that, by offering 92% coverage ratio, the proposed scheduling algorithm saves around 21% energy than COSEN in 500 rounds.

Figure 4.14 shows the network lifetime patterns using PEGASIS, COSEN and the proposed algorithm. In PEGASIS, the first node dies at around 350 rounds, and 90% sensor nodes die at around 600 rounds. In contrast, using COSEN, the first nodes dies at about 400 and more than 90% sensor nodes die at around 550 rounds. Using the proposed algorithm however, it was found that the first node dies at around 500 rounds and 90% of the sensor nodes die after 875 rounds. That means for $\lambda \approx 90\%$, the

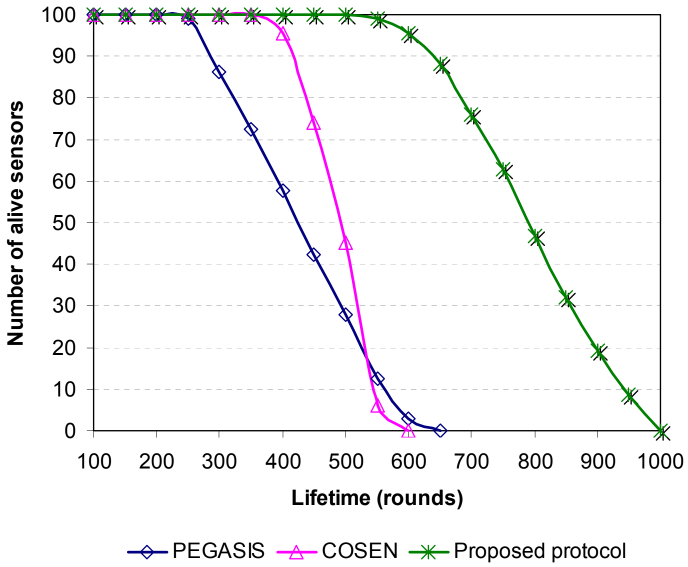


Figure 4.14: Lifetime pattern comparison among PEGASIS, COSEN and proposed protocol. ($|\aleph| = 100, A = 50m \times 50m, r = 10m, \lambda = 92\%, \delta = 2.7$).

proposed algorithm doubles the lifetime of network. If the user requires less coverage ratio, the lifetime can be extended even further.

In addition, to verify the effectiveness of the proposed node scheduling algorithm, extensive simulation experiments were performed to compare the performance of the proposed algorithm with PEAS and PECAS. The reason why these two protocols were chosen is that these two protocols also schedule nodes based on coverage of the target field. The comparison was performed in terms of number of alive nodes over time period. An alive node is a node which has remaining energy and is in one of three states (working, sleeping and waiting). In PEAS, more sensors are in working state but a large part of their sensing area is redundantly overlapped by their neighbours' sensing area. Thus the sensing coverage is not sufficient over time due to the fact that sensors die rapidly. In PECAS, which is an advanced version of the PEAS in terms of energy balance, more sensors also maintain a working state in its early stages. PECAS has a slightly longer lifetime than PEAS. Nevertheless, its sensing coverage is similar to that of PEAS with time. It was observed that the proposed node scheduling algorithm performed much better than PEAS and PECAS in terms of number of alive nodes over time. Figure 4.15 shows the comparison among these three protocols using $\lambda = 95\%$ for 200 sensor deployed in a $200m \times 200m$ target field with a deployment

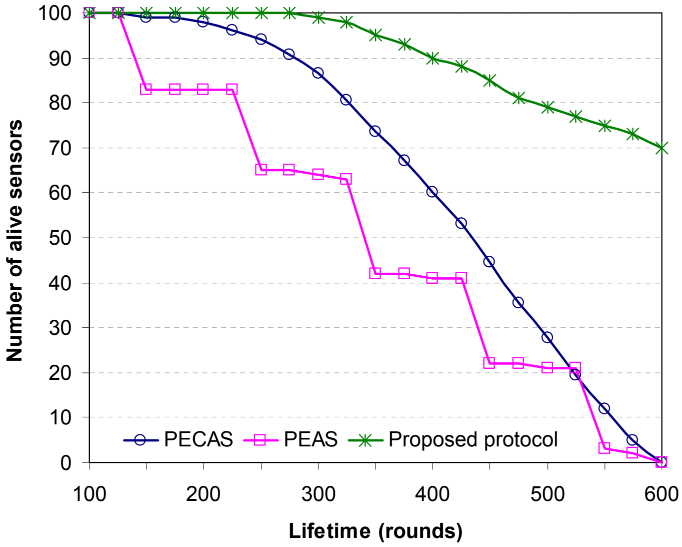


Figure 4.15: Comparison among proposed node scheduling algorithm, PEAS and PECAS. ($|\aleph| = 100, A = 50m \times 50m, r = 10m, \lambda = 92\%, \delta = 2.67$).

density of $\delta = 1.92$.

4.7 Summary

This chapter propose a node scheduling algorithm. The algorithm is designed to select member nodes for the proposed chain oriented logical topology. The node scheduling scheme is motivated by the reason that some applications of wireless sensor networks do not require 100% coverage. Furthermore, in a target field, sensor nodes are usually deployed densely, and this creates redundancy. By exploiting both redundancy of sensor nodes and the requirement of less than 100% of coverage, the proposed member node selection/scheduling algorithm is developed.

As the inherent logical topology guarantees the connectivity of the network, coverage is chosen as the primary basis of designing the algorithm. The primary criteria used to schedule sensor nodes are: number of neighbours, amount of shared sensing of the sensors, residual energy and repetition of selection number. Simulation results show that the proposed node scheduling algorithm saves a significant amount of energy, while sacrificing only a little amount of coverage. For example, the proposed scheduling algorithm saves more than 20% energy as compared to the conventional chain-oriented algorithm while reducing only 7% - 8% of the coverage ratio. For various applications, such as temperature/humidity or sea level monitoring or forest fire detection systems, where 100% coverage is not required, this offers a very useful trade-off to the users.

After selecting a set of nodes from deployed sensor nodes, the selected node would initiate the construction of the logical topology. The next chapter discusses these issues.

Chapter 5

Voronoi Diagram Based Chain Construction

5.1 Preamble

This chapter discusses the second adaptation of the proposed multi-chain oriented logical topology, and this adaptation aims to create an efficient localized chain for the proposed logical topology. The notion of localized chains is primarily motivated by the idea that, if the constructed chains are restricted to local areas, rather than spanning around the network, they encounter fewer problems from radio interference. Adding interference-awareness in designing the chains leads to increased throughput, and further reduces the energy consumption by avoiding unnecessary retransmissions caused by interference-induced packet losses [Johansson et al., 2008]. In addition, restricting chains into local areas also limits the maximum length of chains, and thus reduces the rate of energy consumption. Furthermore, faster convergence time for data gathering and data dissemination are also achievable using localized chains. In order to secure these advantages, this chapter proposes a scheme to create localized chains using a tessellation method.

Tessellation means careful juxtaposition of shapes in a pattern. In WSNs, different tessellation techniques have been used [Chen et al., 2008; Banimelhem and Khasawneh, 2009; Peng et al., 2007; Lima and de Abreu, 2008]. For example, Banimelhem and Khasawneh [2009] used a grid based tessellation technique to divide the target field into multiple square shaped areas. Most of the tessellations are performed manually, which is not feasible in many cases (for example, in an unattended area or a battlefield). On the other hand, automatic tessellations are achieved by the deployed sensor nodes using their communications. Irrespective of whether it is organized manually or automatically, when the sensor nodes are deployed randomly in the target field, the tessellation techniques that use regular geometrical shapes (square, rectangle, hexagon etc.) cannot produce uniform density of the sensor nodes inside the tessellation cells. Uneven distribution of sensor nodes in different cells results in uneven distribution of energy consumption, which in turn adversely affects the network lifetime. Thus a technique is required to divide the target field into smaller areas (cells) with respect to the density of the deployed sensor nodes, instead of geographical size of the target field. In doing so, this chapter proposes to use a well-known tessellation technique based on *Voronoi diagram*. The primary aim of this chapter is to design protocols: i) to tessellate the target field automatically so that each cell contains a similar number of sensor nodes, and then ii) to create efficient localized chains in each tessellation cell taking both interference and energy cost into account.

The rest of the chapter is organized as follows. Section 5.2 further discusses the motivation for employing localized chains. Section 5.3 describes some of the existing tessellation based chain construction algorithms, and identifies the requirements for an improved tessellation based chain construction algorithm. Section 5.4 describes the tessellation technique which would be used for the proposed localized chain creation scheme. Section 5.5 describes the proposed scheme in detail with various illustrations. Experimental results and analysis are presented in Section 5.6 to demonstrate the performance of the proposed scheme. The chapter concludes in Section 5.7.

5.2 Motivation for the Creation of Localized Chains

Radio interference is an inevitable consequence of any wireless communications. Various interference models have been used in existing literature [Burkhart et al., 2004; Von Rickenbach et al., 2005]. This section uses a concise and intuitive definition and metric for interference, as proposed by Burkhart et al. [2004]. In this paper, the interference of a network is defined as the maximum edge coverage occurring in the network. If the distance between two nodes u and v is $d(u,v)$, the coverage of the edge $e = (u,v)$ is defined as the number of nodes within distance $d(u,v)$ from each node. Thus the coverage of an edge $e = (u,v)$ to be the cardinality of the set of nodes covered

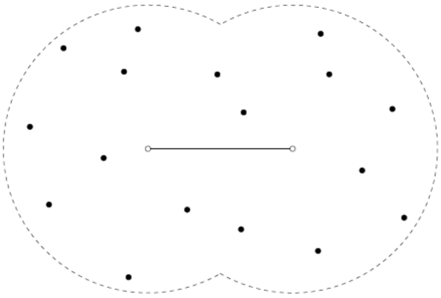


Figure 5.1: Nodes covered by a communication link.

by the disks induced by u and v:

$$\varsigma(e) = \left| \{w \in V | d(u,w) \leq d(u,v)\} \bigcup \{w \in V | d(v,w) \leq d(u,v)\} \right| \tag{5.1}$$

In other words, when the nodes u and v communicate with each other using the minimum required transmission power, the number of other nodes affected by this communication is defined as edge coverage (see Figure 5.1). The edge level interference, defined above, is then extended to a graph interference measure as the maximum coverage occurring in a graph $G(V,E)$:

$$I(G) = \max_{e \in E} \varsigma(e) \tag{5.2}$$

Thus, to reduce interference, the distance between two successive nodes of a chain should be minimized. Although greedy method (choosing the nearest unselected node) keeps the distance between two successive nodes at a minimum during the early stage, it produces longer links at the late stage of chain formation. This is illustrated in Figure 5.2. In this figure, six nodes (A to F) are used to create two chains, supposing that the maximum length of a chain should be three. Figure 5.2(a) shows the greedy approach for chain construction. Assume the chain creation starts from node A. The node A selects B, because B is the nearest node to A, and then the node B selects node C. Thus the construction of the second chain starts from the node D. Node D selects the node E as its successive node. Now, the node E has to select the node F, because F is the only unselected node in the present situation. Note that, the link EF spans almost the width of the network. Thus, when nodes E and F send messages to each other, even nodes of another chain (ABC) suffer the interference. Further

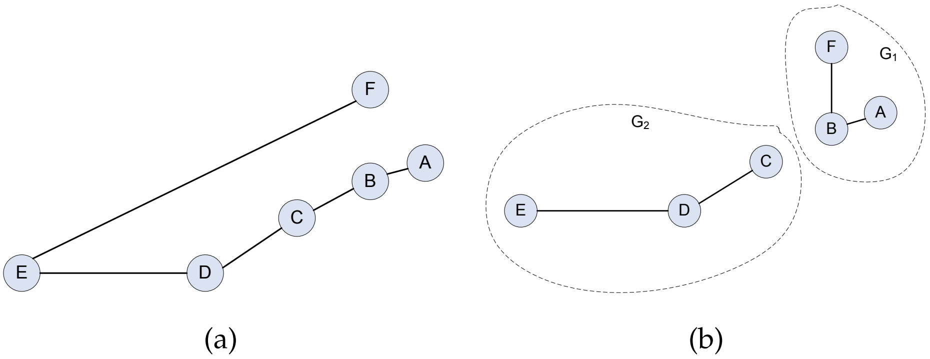


Figure 5.2: Chain construction approaches: (a) using greedy method, (b) creating localized chains.

nodes *A*, *B*, *C* and *D* spend the receiving energy although the message was not intended for them. On the other hand, Figure 5.2(b) shows that the nodes are divided into two groups G_1 and G_2 according to their locations. In G_1, nodes *A*, *B* and *F* create a chain (*ABF*) and in G_2, nodes *C*, *D* and *E* create another chain (*CDE*). Note that, none of these chains spans around the network, and each of them are restricted in an area. Thus chains *ABF* and *CDE* are called localized chains.

5.3 Existing Localized Topology Related Works

Recently, localized topology creation has received enormous attention, and research [Manolopoulos et al., 2010; Li et al., 2004; Mamun et al., 2010d] has been conducted to control localized topology in multi-hop wireless networks. For chain oriented topologies too, a few protocols have been proposed to develop chains which are restricted in local areas. In the first part of this section, some protocols for controlling topology of WSNs are discussed, while the second part of this chapter describes the protocols specifically proposed and designed for chain oriented topologies.

5.3.1 Existing protocols for controlling localized topology

Topology control with transmit power adjustment [Manolopoulos et al., 2010] has been considered an effective tool to generate reduced networks with desired properties such as sparse network connectivity, low interference, and improved network throughput. It works by assigning (reduced) transmission powers to network nodes so as to create a reduced topology while meeting certain specific properties. There are in general several objectives to be considered in designing topology control algorithms [Manolopoulos et al., 2010; Li et al., 2005], including localized operations, low node transmission power, and preserving global connectivity. Some existing works designed to address the issue of topology control with transmit power adjustment are described below.

Rodoplu and Meng [1999] propose the concept of relay region and enclosure graphs, with which each node builds a closure graph and maintains only a set of neighbours with which direct communication is more power-efficient than in the case where an intermediate node is introduced. On the reduced topologies obtained, the distributed Bellman-Ford algorithm is employed to establish the min-power connectivity from all nodes to a master-site. Liu and Li [2003] design a localized Shortest Path Tree (SPT) algorithm, with which each node first builds a local SPT on top of its one-hop neighbourhood and then keeps only those one-hop on-tree neighbours as its neighbours (called neighbour set). Li et al. [2005] propose a Localized Minimal Spanning Tree (LMST) algorithm. With LMST, each node independently builds a local minimal spanning tree on top of its one-hop neighbourhood in generating its neighbour set.

5.3.2 Existing protocols for localized chain creation

Localized chain construction algorithms that are described in this section are collected from routing protocols for WSNs. The protocols are Energy-efficient Chain-cluster Routing protocol (ECR) [Tian et al., 2007], Energy-Balanced Chain Cluster Routing Protocol (EBCRP) [Rong et al., 2010], and Chain-based Hierarchical Routing protocol (CHIRON) [Chen et al., 2009].

Tian et al. [2007] propose a protocol, named ECR, to create several localized chains confined in rectangular areas. Similar to this thesis, ECR also assumes that the BS is situated outside the target field. In the ECR protocol, after deployment, sensor nodes build a relative network coordinate system with the distributed algorithm described in [Capkun et al., 2001], so that every node can calculate its self position in the network system using the Cartesian coordinate system. According to the distance in Y-direction between sensing region and the BS, the sensing region is then divided into several horizontal sub-areas with the same width (see Figure 5.3). Nodes in the

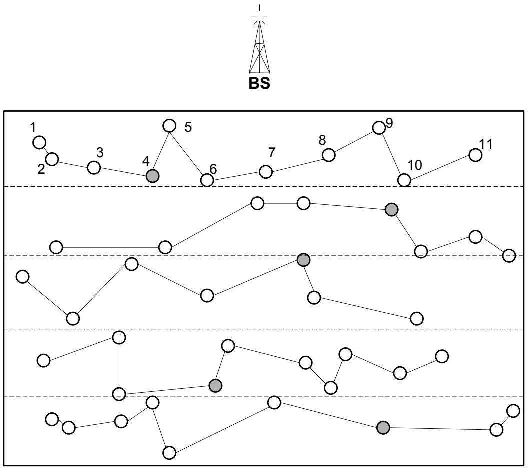


Figure 5.3: ECR divides the target field into several horizontal rectangular shaped regions. In each rectangular area, a chain is then constructed.

same cluster organize themselves into a chain according to the order of X-coordinate from one side to the other. The network topology is maintained until the last node dies even though dead nodes emerge in the network.

Energy-Balanced Chain-cluster Routing Protocol (EBCRP) [Rong et al., 2010] divides the network into several rectangular sections and constructs a routing chain using the ladder algorithm rather than greedy algorithm. The protocol works in a similar manner to ECR protocol, except it divides the target field into several vertical rectangular areas. This is shown in Figure 5.4.

Both ECR and EBCRP suffer from the same serious problems. Firstly, division of a target field into rectangular shaped smaller areas does not guarantee that the distance between two successive nodes of a chain will be smaller. Using these protocols, there can still be a chain constructed which would span the width of the network. Secondly, because of the long distance between two successive nodes in a chain, the interference would be high, and this would affect the lifetime of the network.

Chen et al. [2009], propose a hierarchical chain-based routing protocol (CHIRON), which creates localized chains using the technique of BeamStar [Mao and Hou, 2007] to divide the sensing area into several fan-shaped areas (see Figure 5.5). The sensor nodes within each group are then self-organized into a chain for data dissemination. In this protocol, instead of taking turns, the authors consider the node with a maxi-

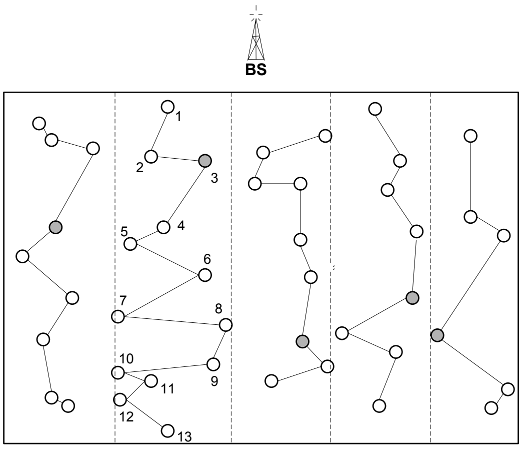


Figure 5.4: EBCRP divides the target field into several vertical rectangular shaped regions. In each rectangular area, a chain is then constructed.

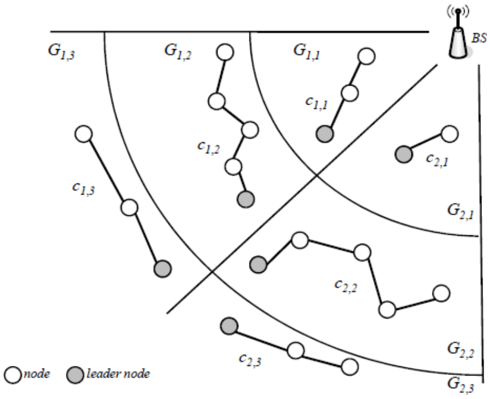


Figure 5.5: CHIRON divides the target field into several fan-shaped areas. In each area, a chain is then constructed.

mum residual energy as the candidate for being a chain leader. In addition, the nearest downstream chain leader is elected for relaying the aggregated sensing information. The problems of this protocol are i) the areas generated in this protocol are very uneven, thus some chains consist of very few sensor nodes while other chains contain a very large number of nodes, ii) this protocol is not scalable, because for a large scale sensor network, the fan-shaped areas would be uncontrollably big, iii) additionally, in a situation where the area is uncontrollably big, the protocol suffers from serious transmission delay, iv) as the same nodes are always selected, those nodes consume more energy, and die soon, and v) energy consumption is not evenly distributed.

5.3.3 Requirements for improved localized chains

From the discussion and examples discussed above in this section the following requirements for localized chains can be identified:

i) Created chains should be limited to a small part of the network.

ii) The distance between any two successive nodes in a chain should also be restricted.

iii) All created cells should have similar node density.

iv) The length of all created chains should be similar.

v) Chains should be of optimal size. If chains are too short, they reduce interference; however, energy consumption would be higher due to the incorporation of multiple higher-level chains. On the other hand, if a chain is too long, both interference and latency would increase.

5.4 Description of the Tessellation Technique Used

In this chapter, a tessellation technique, known as *Voronoi tessellation*, is used to divide the sensor nodes into different groups based on their location. The proposed tessellation technique uses *Voronoi diagram*s. The following section provides a description of elementary, though important, properties of *Voronoi diagram*s, and their applicability in creating localized topology structures.

Voronoi diagrams: definition and properties

Voronoi diagram is a useful planar subdivision of a discrete point set. It represents distance relationships and growth phenomena. Thus, it is not surprising to use *Voronoi diagram*s to model crystal growth or large objects in the universe, and to observe them in natural objects, such as in the carapace of turtles, or in the necks of giraffes. *Voronoi diagram*s have also been used for solving many problems, such as nearest neighbour search, motion planning, and cluster analysis [Aurenhammer, 1991]. In WSNs, *Voronoi cell*s can provide a means to help monitoring and tracking targets [Chen et al., 2004], conserving energy [Zhou et al., 2004], and balancing workloads [Chen et al., 2008].

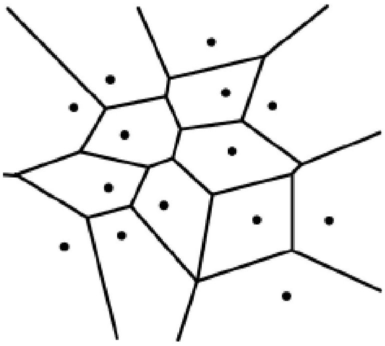


Figure 5.6: *Voronoi diagram.*

A generic definition of *Voronoi diagram* can be given as follows. Let S denote a set of n points (called sites) in a plane R^2. For two distinct sites $p, q \in S$, the dominance of p over q is defined as the subset of the plane R^2 being at least as close to p as to q. Formally,

$$dom(p,q) = \left\{x \in R^2 | d(x,p) \leq d(x,q)\right\} \tag{5.3}$$

where $d()$ denotes the Euclidean distance function. Clearly, $dom(p,q)$ is a closed half plane bounded by the perpendicular bisector of p and q. This bisector separates all points of the plane closer to p from those closer to q, and will be termed the separator of p and q. The region of a site $p \in S$ is the portion of the plane lying in all of the dominances of p over the remaining sites in S. Formally,

$$reg(p) = \bigcap_{q \in S - \{p\}} dom(p,q) \tag{5.4}$$

Since the regions are coming from intersecting $n - 1$ half planes, they are convex polygons. Thus the boundary of a region consists of at most $n - 1$ edges (maximal open straight-line segments) and vertices (their end points). Each point on an edge is equidistant from exactly two sites, and each vertex is equidistant from at least three. As a consequence, the regions are edge to edge and vertex to vertex, that is to say, they form a polygonal partition of the plane. This partition is called the, $V(S)$, of the finite point-set S (see Figure 5.6). One of the important properties of the *Voronoi diagram*s is that dense subsets of sites give rise to *Voronoi cell*s of small area and sparse subsets of sites produce larger *Voronoi cell*s. Thus *Voronoi diagram*s balance the site density. For this reason, this chapter proposes to use *Voronoi diagram* to tessellate the target field.

5.5 Description of the proposed Chain Construction Scheme

The aim of this chapter is to design a scheme that constructs a number of localized chains, and selects a sensor node as a leader for each chain. To construct localized chains, *Voronoi tessellation* method is used. The proposed chain construction scheme goes through four sequential steps. Related protocols are designed and applied in each of these steps. In short, a number of sensor nodes are selected as tentative leaders, and then *Voronoi diagram*s are created with respect to the tentative leaders. After that, in order to conform to sizes and node density of *Voronoi cell*s, a *Voronoi diagram* management protocol is applied. Finally, in each *Voronoi cell*, an efficient chain is constructed. The four sequential steps, namely i) tentative leader selection, ii) *Voronoi diagram* construction, iii) *Voronoi cell* management, and iv) chain construction in a *Voronoi cell*, are discussed in the following sections.

5.5.1 Tentative leader selection

At the very first stage of the scheme, tentative leaders are selected. The selected tentative leaders are used in the second step to construct *Voronoi diagram*s. The nodes selected in this step may become the leaders of constructed chains, depending on the *Voronoi cell* size. If the *Voronoi cell* created is of optimal size, the node which is used to construct this *Voronoi cell* becomes the leader of the chain in that *Voronoi cell*. In order to select a tentative leader, the residual energy of each node is primarily considered. The other factor that is considered is the number of times the node was previously selected as a leader. In doing so, each node chooses a random number from 0 to n. If the number is less than a threshold T_n, the node is nominated as a tentative leader. To select a tentative leader, the threshold T_n and the node's residual energy E_R are used as parameters in the following function $p(S)$,

$$p(S) = f(E_R, T_n)$$

where

$$T_n = \begin{cases} \frac{p}{1-p\times\left(r \mod \frac{1}{p}\right)}, & n \in G; \\ 0, & \text{otherwise.} \end{cases} \tag{5.5}$$

Here p is the desired number of chains. In the simulation the value of p is found from 0.05 to 0.08. This means that, with 100 sensor nodes deployed in the target field, the optimal number of chains should be from 5 to 8. Here r is the current round, and G is the set of nodes that have not been local leaders in the last n rounds. Using this threshold, each node will be nominated as a leader at some point within n rounds.

5.5.2 *Voronoi cell* construction

In a two-dimensional plane, the *Voronoi diagram* of a set of discrete points (sites) partitions the plane into a set of convex polygons such that all points inside a polygon are the closest to only one site. This construction effectively produces polygons with edges that are equidistant from neighbouring sites. Several *Voronoi diagram* construction algorithms exist in literature [Zhou et al., 2004; Fortune, 1986; Sharifzadeh and Shahabi, 2004]. However, the difference between these algorithms and the proposed algorithm is the number of nodes on which the protocol would run. All of the existing algorithms construct *Voronoi diagram* using all the sensor nodes in the target field. However, the proposed *Voronoi diagram* construction algorithm considers only a few of the nodes which are leaders (recall from Section 3.4.5 that only 6%-8% of the total sensor nodes are elected as lower-level leaders).

Given a set of points $P = \{p_1, p_2, \ldots p_n\}$ in a two-dimensional plane R, the *Voronoi Diagram* divides R into n cells. Each cell C_i centers at point p_i. Any point in cell C_i is closer to p_i than to any other centers. Thus,

$$C_i = \bigcap_{i \neq j, j=1}^{n} \left\{ p | d(p_i, p) \leq d(p_j, p) : p \in R \right\} \tag{5.6}$$

where p is a point in plane R; d denotes the Euclidean distance between p_i and p.

The lines at the boundaries of the *Voronoi diagram* extend to infinity. However, since there are only a few nodes taking part, the *Voronoi diagram* can be clipped to the boundaries of the field. Since travelling along the boundaries of the sensor field also constitutes a valid path, extra edges in the *Voronoi diagram* corresponding to the bounds are introduced. In subsequent discussions, *Voronoi diagram* refers to the bounded diagram.

To construct the *Voronoi diagram*, the selected tentative leader nodes are required to know their locations. This is usually achieved through GPS or other techniques [Wang et al., 2007; Girod and Estrin, 2001]. The Euclidean distance from a sensor to a

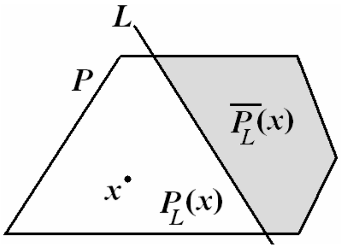


Figure 5.7: Straight line L divides the polygon P into two parts.

given point in the plane is thus computable. The following paragraph describes the essential notations used in the algorithm for creating *Voronoi cell*s.

Let a point x be inside a polygon P (see Figure 5.7). A straight line L divides the polygon P into two parts $P_L(x)$ and $\bar{P}_L(x)$, such that the following relations hold: i) $P_L(x) \bigcup \bar{P}_L(x) = P$, ii) $P_L(x) \bigcap \bar{P}_L(x) = \phi$, iii) $x \in P_L(x)$, and iv) $x \notin \bar{P}_L(x$.

Let $S = \{S_1, S_2, \ldots, S_k\}$ be the set of sensor nodes for which a *Voronoi diagram* would be constructed. An algorithm is proposed (see Figure 5.8) to calculate the *Voronoi cell* for a single sensor node. When all the tentative leaders run this algorithm, a *Voronoi diagram* for the whole network would be constructed.

In the proposed algorithm, each sensor node S_i broadcasts its location information $S_i.loc$, to all other sensor nodes. Each sensor node maintains a queue, $S_i.Q$. After collecting the broadcasted messages from its neighbours, each sensor node stores the location information of its neighbours in an ascending order according to the distance of senders from itself. Using the information stored in the queue, each sensor node S_i then constructs a set of straight lines, which are computed as the perpendicular bisectors between the node S_i and the rest of the nodes in $S_i.Q$. The sensor node S_i constructs a set of perpendicular bisector lines $B_i = \{b_1, b_2, \ldots, b_k\}$ where b_1 is the bisector line between sensor S_i and sensor S_1, b_2 is the bisector line between sensor S_i and sensor S_2, and so on.

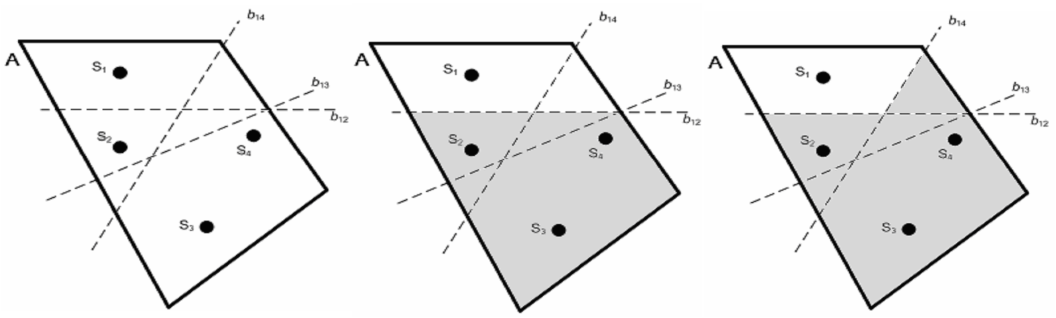
Initially, the sensor node S_i assumes the whole area is its *Voronoi cell*. When S_i receives messages from its neighbours and calculates B_i, sensor node S_i can rectify its *Voronoi cell* using the algorithm shown in Figure 5.8.

The algorithm in Figure 5.8 constructs *Voronoi cell*s for each of the tentative leaders selected in the previous phase. The proposed algorithm requires only a few messages.

```
Input:  A bounded area $A$,
        A sensor node $S_i$
Output : The Voronoi cell of the node $S_i$

Procedure Voronoi_cell ($S_i$, $A$)
  //initialize the cell
$S_i.cell = A$
  //broadcast $S_i$'s location to all adjacent nodes
$S_i$.send_message(broadcast, $S_i.loc$)
  //listen for messages from neighbours, and store the messages in $S_i.Q$
    in ascending order according to the sender's distance from $S_i$
Construct a set of perpendicular bisectors $B = \{b_1, b_2, \ldots, b_k\}$
For each $b_i$ in $B$
        if $(S_i.cell \cap \overline{A}_{b_i}(S)) \neq \phi$
                $S_i.cell = S_i.cell - \overline{A}_{b_i}(S)$
return $S_i.cell$
```

Figure 5.8: *Voronoi diagram* construction algorithm for a sensor node S_i.



(a) (b) (c)

Figure 5.9: *Voronoi cell* construction for the node S_1.

Moreover, a sensor node does not need to collect location information from all other nodes, especially from distant nodes, as those nodes do not essentially affect the shape of the *Voronoi cell*. Figure 5.9 depicts an example of the situation. In Figure 5.9a, S_1 computes b_{12}, b_{13}, and b_{14}. Initially S_1 assumes the entire region as its *Voronoi cell*, S_1.cell=A. Figure 5.9b shows that S_1 refines its *Voronoi cell* (shaded region is excluded) by S_1.cell=S_1.cell - $((\overline{A})_b)_{12}(S_1)$. Finally, in Figure 5.9c, S_1 refines its *Voronoi cell* by S_1.cell=S_1.cell - $((\overline{A})_b)_{14}(S_1)$; as S_1.cell $\cap$ $((\overline{A})_b)_{13}(S_1)$=$\phi$, b_{13} does not affect the *Voronoi cell* S_1.cell.

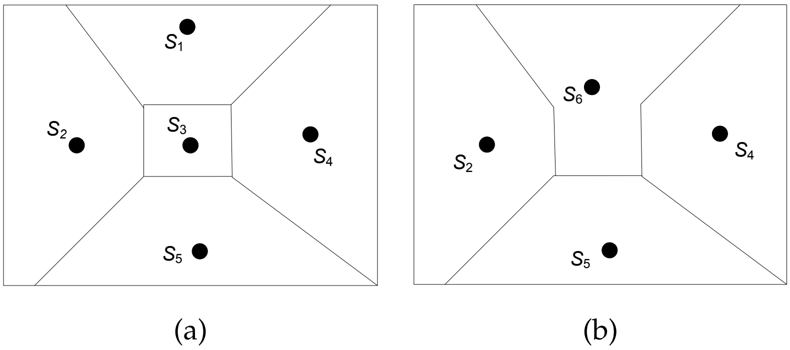


Figure 5.10: Merging two *Voronoi cell*s to get rid of an undersized *Voronoi cell* for S_3.

5.5.3 *Voronoi cell* management

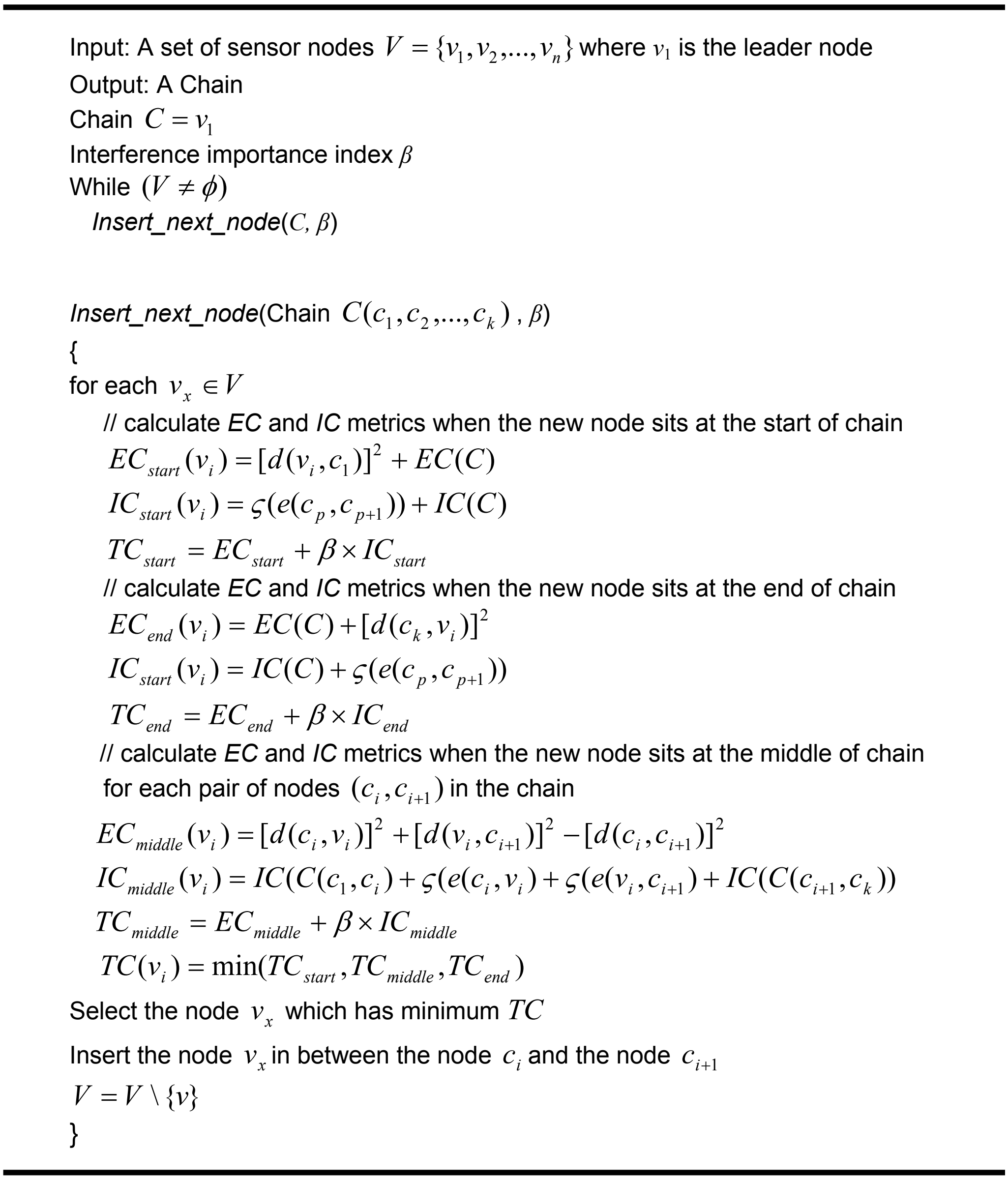
After constructing the *Voronoi diagram* using the tentative leaders, *Voronoi cell*s are compared against a threshold value. If the cell is found to be too small, it is merged with one of its neighbour cells. On the other hand, if a cell becomes too large, the cell is split into two cells, and new leaders for the new cells are elected using the same steps followed in Section 5.5.1. Figure 5.10 shows an example of merging two cells into one.

5.5.4 Constructing a chain inside a *Voronoi cell*

When all the *Voronoi cell*s are established, the chain construction starts in each *Voronoi cell*. The very first step of this process is to recognize the tentative leaders as leaders. The leaders then initiate the chain construction process in each of the *Voronoi cell*s. To construct a chain, both energy cost and interference cost are considered. As data transmission energy is directly proportional to the square of distance, the energy cost metric (EC) is calculated to be the total of the square of the distance between two successive nodes in the chain. Thus, assuming two successive nodes u and v in a chain C, the energy cost metric is calculated by:

$$EC = \sum_{\forall(u,v)\in C} (d(u,v))^2 \tag{5.7}$$

On the other hand, the interference cost metric (IC) is calculated to be the total coverage of all edges along the chain. Assume an edge between two successive nodes u and v in a chain C. As discussed in Section 5.2, the coverage of the edge is defined



```
Input: A set of sensor nodes $V = \{v_1, v_2, ..., v_n\}$ where $v_1$ is the leader node
Output: A Chain
Chain $C = v_1$
Interference importance index $\beta$
While $(V \neq \phi)$
  Insert_next_node($C, \beta$)

Insert_next_node(Chain $C(c_1, c_2, ..., c_k)$ , $\beta$)
{
for each $v_x \in V$
   // calculate EC and IC metrics when the new node sits at the start of chain
   $EC_{start}(v_i) = [d(v_i, c_1)]^2 + EC(C)$
   $IC_{start}(v_i) = \varsigma(e(c_p, c_{p+1})) + IC(C)$
   $TC_{start} = EC_{start} + \beta \times IC_{start}$
   // calculate EC and IC metrics when the new node sits at the end of chain
   $EC_{end}(v_i) = EC(C) + [d(c_k, v_i)]^2$
   $IC_{start}(v_i) = IC(C) + \varsigma(e(c_p, c_{p+1}))$
   $TC_{end} = EC_{end} + \beta \times IC_{end}$
   // calculate EC and IC metrics when the new node sits at the middle of chain
   for each pair of nodes $(c_i, c_{i+1})$ in the chain
   $EC_{middle}(v_i) = [d(c_i, v_i)]^2 + [d(v_i, c_{i+1})]^2 - [d(c_i, c_{i+1})]^2$
   $IC_{middle}(v_i) = IC(C(c_1, c_i) + \varsigma(e(c_i, v_i) + \varsigma(e(v_i, c_{i+1}) + IC(C(c_{i+1}, c_k))$
   $TC_{middle} = EC_{middle} + \beta \times IC_{middle}$
   $TC(v_i) = \min(TC_{start}, TC_{middle}, TC_{end})$
Select the node $v_x$ which has minimum $TC$
Insert the node $v_x$ in between the node $c_i$ and the node $c_{i+1}$
$V = V \setminus \{v\}$
}
```

Figure 5.11: Chain construction algorithm inside a *Voronoi cell*.

by the number of nodes within the distance $d(u, v)$ from either node u or node v. Thus,

$$IC = \sum_{\forall e \in C} \varsigma(e(u, v)) \tag{5.8}$$

The total accumulated cost metric is thus calculated as $(EC + \beta \times IC)$, where β is decided by the user of the WSN as the relative importance of the interference and energy cost factors.[1] Each time a node is selected as the next member of the chain, based on its contribution to the accumulated cost metric. Thus a new member node increases the minimum possible energy consumption and interferences, compared to the old chain. In this process, the chain may be broken to insert the new node to the chain. The chain construction algorithm is demonstrated in Figure 5.11.

5.6 Analysis and Simulation Results

Various protocols were designed for the proposed chain construction scheme. The main two protocols among them are the *Voronoi diagram* construction protocol and the chain construction protocol. The following presents the discussions of these protocols.

First, assume the protocol that is used in constructing a *Voronoi diagram*. Although the construction of a *Voronoi diagram* is a well-studied topic in the field of computational geometry, computing it in a distributed fashion, especially in wireless sensor networks, is a relatively new topic. In wireless sensor networks, the challenges of distributed computation are added to the vulnerability of the network, such as energy limitations, wireless link failures, and low bandwidth. Therefore, not only time and space of computation are important, but also efficiency in terms of power consumption, bandwidth usage, and fault tolerance are equally important. It is noteworthy that the *Voronoi cell*s are created not for all the sensors of the networks. Only very few (5% to 8%) of the nodes take part in this process. Also, such a node does not need to collect all broadcasted messages, only those from its neighbours.

Second, assume the protocol that is used for the chain construction in each *Voronoi cell*. The chain construction algorithm always guarantees to construct a chain that has the lowest total cost, which consists of energy cost and interference cost. In wireless communication, transmission energy is directly proportional to at least a square of the distance between the sender and the receiver nodes. Thus, the constructed chains, cre-

[1] Interference affects the reliability, and causes high delay and congestion. Using these considerations, an *importance index* can be created that will consider the performance of the network in respect of interference. On the other hand, energy consumption factor directly affects mainly the lifetime of the network along with reliability and QoS. Thus, another *importance index* can be created that will consider the performance of the network in respect of energy consumption. Using the ratio of these two indexes, β can be derived. It dictates the importance of considering the interference factor of a WSN. For example, for a sparse WSN the value of should be lower than that of dense WSN.

ated using the proposed chain construction algorithm, consume the lowest transmission energy. The protocol also reduces the interference caused by the communication between any two successive nodes in the chain. Furthermore, *Voronoi cell*s keep the chain lengths under restriction. None of the chains created is too large or too small. Moreover, interference-awareness restricts the distance between any two successive nodes in a chain. In this way, the energy consumption of constructed nodes are evenly distributed. This affects the lifetime of the networks. The following sections describe the results from simulation.

5.6.1 Simulation environment and parameters

The simulation results of the proposed protocols are compared with ECR, EBCRP, PEGASIS, COSEN, and CHIRON. Various experiments are performed to compare the following: i) total length of created chains, ii) distribution of energy consumption, iii) interference produced by the nodes when any two successive nodes communicate to each other, and iv) network lifetime. The simulation program is developed and written in object oriented programming language *C++*. Two dimensional Cartesian coordinates are used to locate the sensor nodes and the BS. The BS is fixed in position and is located at (25, 150). It is assumed that the sensor nodes are placed randomly in the target field. The initial energy of each sensor is assumed as one Joule.

The transmitter amplifier (E_{amp}) is assumed to be 100 pJ/bit/m^2 to amplify the signal at an acceptable signal to noise ratio (SNR). In addition, energy required in running transmitter and receiver electronics are assumed to be equal and given by $E_{tx-elec} = E_{rx-elec} = E_{elec} =$ 50nJ/bit. Moreover, the energy cost for data aggregation is considered as 5nJ/bit/message [Heinzelman et al., 2000]. The bandwidth of the channel is set to 1 Mb/s [Kulik et al., 2002]. In the experiments, each data message is assumed to be 2000 bits long and information-processing time in a node is taken between 5 to 10 milliseconds. The medium is assumed symmetric such that the energy required for transmitting a message from nodes A to B and from B to A are the same at a fixed SNR. Therefore, for free space propagation loss, energy dissipation is certainly dominated by the long distance transmissions. Thus, the total transmission cost for a k-bit message is given by the Equation 5.9.

$$E_{tx}(k,d) = E_{elec} \times k + E_{amp} \times k \times d^2 \tag{5.9}$$

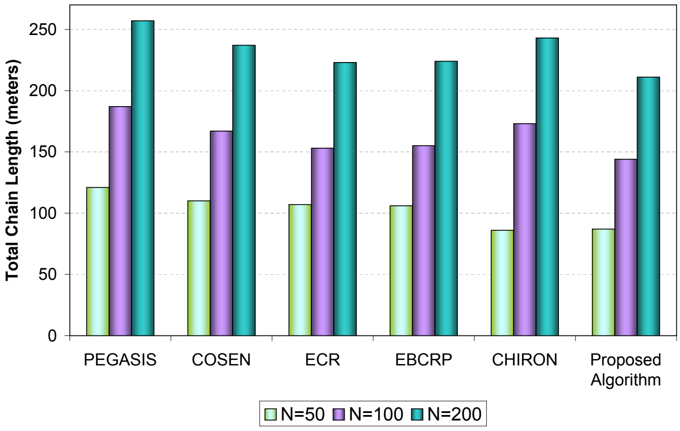


Figure 5.12: Total chain length comparison.

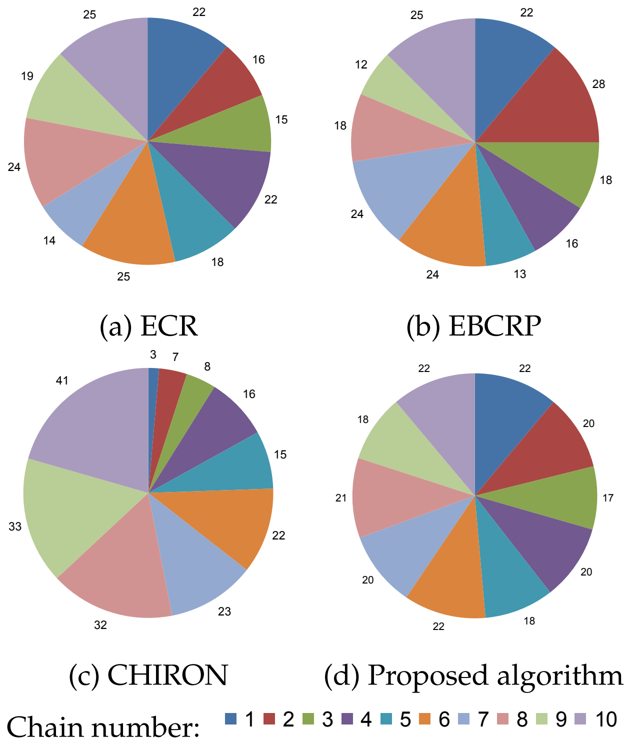
Here d is the distance between sender and receiver measured in meters. In the case of receiving a message, the energy consumption equation is given by Equation 5.10

$$E_{rx}(k) = E_{elec} \times k \tag{5.10}$$

5.6.2 Simulation results

Extensive simulation experiments were conducted to quantify the effectiveness of the proposed localized chain construction scheme. The experiments were performed to demonstrate the energy efficiency, even distribution of energy consumption, reduced interference and longer lifetime achievement using the proposed scheme. The results of all experiments are described below.

The first simulation experiment calculates the total chain lengths of PEGASIS, COSEN, ECR, EBCRP, CHIRON and the proposed algorithm. In PEGASIS, only one chain is constructed, whereas the rest of the protocols create multiple chains. Figure 5.12 shows the experimental results. The results shown in Figure 5.12 include only the lowest hierarchical level chain lengths. For all the multiple chain construction protocols, higher hierarchical level chains were not calculated. In this experiment, 50, 100 and 200 nodes were placed in an area of $100m \times 100m$. In each instance, the proposed algorithm demonstrated the best performance, by creating chains with the shortest total chain length. In the case of PEGASIS, the total chain length remained good until the very last moment where the last few links contributed the most to increase the total



(a) ECR

(b) EBCRP

(c) CHIRON

(d) Proposed algorithm

Figure 5.13: Number of nodes per chain.

chain length. ECR and EBCRP produced almost the same results in every set of data. CHIRON, although giving good results initially, demonstrated its inefficiency for the large set of sensor nodes. The main reason is that CHIRON produces fan shaped tessellation cells, which grow exponentially as the distance from the BS increases. Thus very large areas of cell are produced. If, accidentally, the node density of a big cell becomes lower, very longer links are created in that cell.

The second experiment compares the distribution of energy consumption by ECR, EBCRP, CHIRON and the proposed scheme. In chain oriented topology, the length of chains is a critical issue, because a chain with similar length in each cell is the prerequisite of even distribution of energy consumption. In this experiment, energy distribution was compared by the lengths of chains in different cells created by the above mentioned protocols. Chain length was measured by the number of nodes on a chain. In the experiment, 200 sensor nodes were placed an area of $100m \times 100m$ area. Ten cells were created by each protocol, and then the number of sensor nodes in each cell was counted. In all of the above mentioned protocols, all the sensors in a cell created a sin-

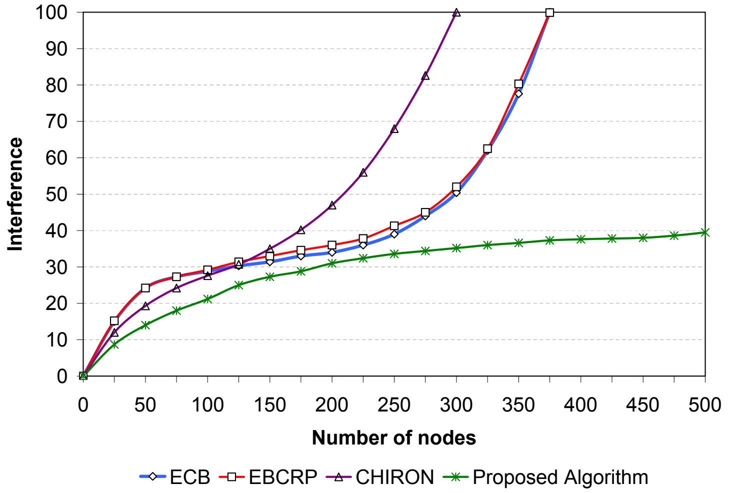


Figure 5.14: Amount of interference comparison among chains constructed by various protocols.

gle chain. Thus, the number of sensor nodes in a cell referred to the length of the chain created in that cell. Figure 5.13 shows the results of the experiment using pie charts, where each sector represents the length of a chain in each cell. The proposed algorithm ensures that each chain has similar numbers of nodes. On the other hand, CHIRON creates chains with dissimilar lengths. Standard deviation[2] of the chain lengths were measured. The standard deviation descriptor value for the proposed algorithm is only 1.82, while that of CHIRON is 12.52. This is because, in CHIRON, the shortest chain consists of only 3 nodes, whereas the longest chain contains 41 nodes. Using the proposed algorithm, the shortest and longest chains are consisting of 17 and 22 nodes respectively. With respect to the similarity of chain lengths, ECR (standard deviation 4.16) shows slightly better performance than EBCRP (standard deviation 5.40). Similar chain lengths ensure even distribution of energy consumption, and hence prolonged lifetime of the network. Additionally, time division multiple accesses (TDMA) scheduling becomes more efficient when all chains are of same length. In this respect, the proposed algorithm again outperforms all other protocols discussed in this section.

The third experiment measures the interference according to the interference model discussed in Section 5.2. It is known that higher interference results in restricting

[2] The standard deviation is a measure of how widely values are dispersed from the average value (the mean).

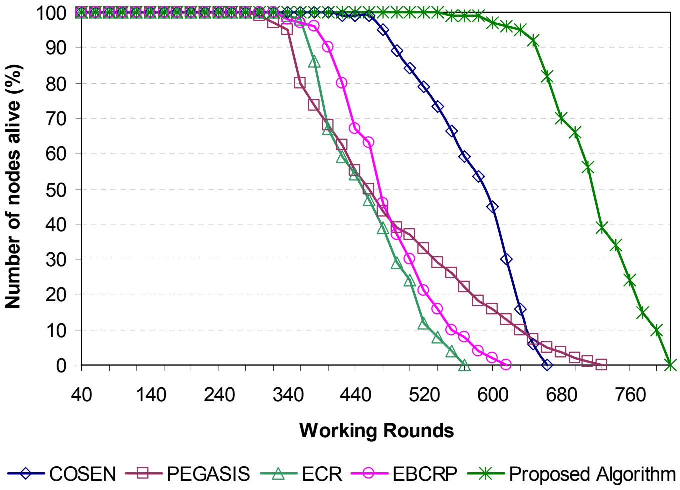


Figure 5.15: Lifetime comparison (N=200, Area=100m × 100m).

higher number of sensor nodes to transmit data. Thus, an improved chain construction algorithm should consider not only the energy consumption, but also the interference. In this experiment, the number of sensor nodes is increased gradually to observe the performance of different protocol with regard to interference. Figure 5.14 shows the results. It is found that, with respect to interference, too, the proposed scheme outperforms each single protocol listed above. Actually, none of the protocols, except the proposed one, consider interference as a design issue while constructing the chains. That is why the interference amount of ECR, EBCRP and CHIRON increases exponentially when the number of nodes increases.

The fourth and last experiment is related to the lifetime of the network. Network lifetime is directly related with the energy consumption and its even distribution. Figure 5.15 shows the comparison of lifetime among PEGASIS, COSEN, ECR, EBCRP and the proposed algorithm. Note that, for the proposed scheme, the first node dies at around 600 rounds, whereas for ECR, EBCRP and PEGASIS, the first node dies before 400 rounds. Additionally, for these protocols, 50% or more of the sensor nodes die before 450 rounds, whereas in the proposed algorithm 50% of the sensor nodes die at around 700 rounds. Figure 5.15 clearly shows the superior performance of the proposed chain construction scheme.

5.7 Summary

In this chapter, a *Voronoi diagram* based chain construction scheme for the multi-chain oriented topology is proposed. Constructed chains using the proposed scheme show the following characteristics: i) all constructed chains are restricted to a geographical area, ii) the chains are optimal in chain lengths, iii) all the created chains are of similar length, and iv) no long link between any two successive nodes of a chain. These characteristics of chains assure low interference and low energy consumptions. The proposed scheme is compared with other chain construction algorithms and it is found that the 'total chain length' of the proposed scheme is smaller than that of other algorithms (30% smaller than PEGASIS and 12% smaller than COSEN). As energy consumption is directly proportional to the distance between the source and destination nodes, the proposed algorithm saves more energy. The proposed scheme also constructs chains of similar sizes, and puts restrictions on the distance between any two consecutive nodes. Thus, energy consumption of the proposed scheme is evenly distributed. This contributes to the longer lifetime of the network.

The enhanced performance of the constructed chains using *Voronoi diagram* motivates to apply this diagram for further use in the proposed logical topology. In the next chapter, *Voronoi diagram* is reused to design the third and final adaptation of the proposed multi-chain oriented logical topology.

Chapter 6

Data Gathering using Mobile Data Collectors

6.1 Preamble

This chapter discusses the third and final adaptation of the proposed logical topology for WSNs. This adaptation aims to design an efficient data gathering /collection scheme for the network.

Besides monitoring the environment, by taking spatial or temporal measurements, sensors in a WSN are also responsible for routing the sensed data back to the BS [Akyildiz et al., 2002; Shah et al., 2003]. The BS is usually situated outside the sensing field. Sending the data by the sensors to the remote BS may lead to non-uniform energy consumption among the sensors, because the sensor nodes (or chain leaders in the context of the proposed topology) that are responsible for sending data to the BS, need to cover a long-range distance. As a result, they deplete energy much faster than other sensor nodes. Thus, the problem of efficiently collecting data from the scattered sensors, which is generally referred to as a data gathering problem, is an important and challenging issue, as it largely determines the network lifetime.

To solve the data gathering problem, this chapter proposes a scheme of utilizing multiple mobile data collectors (MDCs) and the spatial division multiple access (SDMA) technique. The sensing field is divided into several non-overlapping regions, and for each of the regions, an MDC is assigned. Each MDC takes the responsibility of gathering data from the chain leaders in the region while traversing their transmission ranges. The traversal paths of the MDCs are determined using the *Voronoi diagram* constructed with respect to the chain leaders (see Section 5.5.2). This chapter also considers exploiting the SDMA technique by equipping each MDC with two antennas. With the support of SDMA, two distinct compatible chain leaders in the same region can successfully make concurrent data uploading to their associated MDC. Intuitively, if each MDC can simultaneously communicate with two compatible chain leaders, the data uploading time in each region can be cut in half in the ideal case.

To solve the data gathering problem, this chapter further focuses on the problem of minimizing data gathering time among different regions. Besides this, the data gathering problem using multiple MDCs and the SDMA technique requires optimal solutions for the following queries: i) which chain leader is to be associated with which MDC, ii) how the target field should be divided into several regions so that each region would have an optimal number of chain leaders (or sensor nodes), iii) how to determine the location from where MDCs would collect data from chain leaders, iv) how the traversal paths for each MDC would be designed so that the cost of running an MDC would be minimal, and v) how the MDCs report data to the BS. These optimization problems can be formulated using an Integer Linear Programming (ILP) approach. However, the complexity of an ILP solution is generally high, which is not suitable for a large scale WSN. Therefore, this chapter proposes a heuristic region-division and tour-planning algorithm to provide a practicaly good solution to the problem. Extensive simulation experiments are performed to analyse and evaluate the performance of the proposed data gathering scheme. Simulation results demonstrate that the proposed scheme significantly outperforms other data gathering techniques, saves a significant amount of energy, and helps to achieve a longer lifetime of the WSN. Additionally, the proposed scheme is also compared with other non-SDMA, or single mobile collector schemes, where it is found that the proposed scheme performs better than the other schemes by efficiently shortening and balancing the data gathering time among different regions.

The rest of the chapter is organized as follows. Section 6.2 discusses different types of techniques used in WSNs for data gathering. Section 6.3 gives details of the motivations for deploying MDCs and the SDMA technique for the proposed data gathering scheme, which employs multiple MDCs and SDMA. Section 6.4 defines the terminologies used in the proposed scheme. This section also describes the assumptions, and an overview of the proposed scheme. A detailed description of the proposed scheme, along with an example, is presented in Section 6.5. Analysis and simulation results are described in Section 6.6. Finally, a summary for this chapter is given in Section 6.7.

6.2 Different Data Gathering Techniques Used in WSNs

Various types of data gathering mechanisms have been investigated for large-scale sensor networks. They can be roughly classified into the following three categories. The first category employs a static sensor network, which contains a large number of static sensor nodes and a static sink (usually situated inside the network) or a BS (usually situated outside the network). The sink or the BS must be reachable by all the sensor nodes. Data packets are sent to the sink or the BS by one or more hops of forwarding. In such a network, all data traffic flows to the sink or the BS. Thus, the sensors, which are close to the sink, consume more energy than the sensors, which are at the margin of the network. On the other hand, as the BSs are situated outside the target field, the sensor nodes which are far from the BS lose more energy to transmit data to the BS. In order to improve the scalability, Heinzelman et al. [2000], and Younis and Fahmy [2004] propose protocols for randomized cluster forming and cluster head selection. However, they assume that all sensor nodes in the network are homogeneous, and have the same computational and communication capability. The authors also assume that every sensor could be elected as the cluster head. Thus, every sensor node has to be powerful enough to communicate with other nodes within the cluster and with the BS. Furthermore, in such a static network, every sensor has to perform all the functions by themselves, such as finding routing paths, obtaining its location information [Kaplan, 1996; Capkun et al., 2001], scheduling the packet transmission, and so on. Thus, the network architecture with one static sink or BS is only suitable for a small network.

The second type of architecture introduces a hierarchy to the network (see [Ma et al., 2005; Zhang et al., 2008]). By adding a small number of powerful cluster heads, the network can be divided into clusters. In such a network, sensor nodes are organized into clusters and form the lower layer of the network. At the higher layer, cluster heads collect sensed data from sensors and forward the data to the outside BS. Such two-layer hybrid networks are more scalable and energy efficient than homogeneous sensor networks. However, though increasing the number of cluster heads may reduce the burden of sensor nodes, the cost of cluster heads should also be taken into consideration.

The third type of sensor networks introduces one or more mobile data collectors to collect data dynamically. A mobile data collector could be a mobile robot or a vehicle equipped with a powerful transceiver, battery, and large memory. The mobile data collector starts a tour from the base station, traverses the network, and collects sensed data from nearby nodes while moving. It then returns and uploads the data to a remote data processing centre. The moving path and the direction of the mobile collector can be random or planned. When the mobile collector moves into the transmission range of some sensors, the sensors send the data to the collector directly. Other sensors, which are too far away from the moving path of the mobile collector, can upload the data through the relaying of other sensors. The relaying path and transmission time of each packet can be determined by the mobile collector. In addition, a GPS receiver may be an option for sensors, since sensors can estimate the relative location to the mobile collector when the mobile collector moves closer to them [Kaplan, 1996]. By introducing the mobility of the data collector, the energy consumption for transmitting the packets can be reduced significantly, and sensor nodes can be made simpler and less expensive.

To solve data the gathering problem, this chapter proposes to use mobile data collectors. Although the proposed scheme uses multiple MDCs, the basic idea is the same, with some modification, as the category three, which is described above. The motivations for selecting this type of technique are described in the next section.

6.3 Motivations for the Proposed Data Gathering Scheme

This section describes the motivations of using MDCs and the SDMA technique in the proposed data gathering scheme. In recent studies, mobility has received much attention as an effective solution to alleviate the non-uniformity problem of power consumptions [Shah et al., 2003; Zhao et al., 2004b; Luo and Hubaux, 2005; Somasundara et al., 2006; Zhao et al., 2008]. Different from the traditional routing, these schemes use a special type of mobile nodes (MDCs) for facilitating connectivity among static sensors. These mobile nodes can move arbitrarily closer to the sensors, and collect data from them via short-range communication. Thus long-range communication, for example, from a chain leader to other chain leaders, or from a chain leader to the base station, can be avoided. Moreover, using these schemes, the routing burden is taken over by mobile nodes, and energy can be greatly saved at sensors. The recent work in [Zhao et al., 2008] employs a single mobile collector in WSNs, and focuses on optimizing the data gathering tour. However, in a large-scale WSN, utilizing only a single mobile data collector may lead to a long data gathering cycle and data buffer overflow at sensors. To deal with these problems, this chapter considers deploying multiple MDCs which would work simultaneously and independently in a sensing field to collect data from sensors.

In addition to deploying multiple MDCs, this chapter also uses the SDMA technique in data gathering by mounting multiple antennas on each MDC. SDMA is a multi-user multiple-input and multiple-output (MIMO) technique, specifically with multiple receiving antennas [Tse and Viswanath, 2009]. It enables multiple senders simultaneously to transmit data to a receiver. To elaborate, if each MDC is equipped with two antennas, and if each sensor still has a single antenna, two compatible sensors in the same region can concurrently upload data to the associated MDC by utilizing the SDMA technique. Thus, data uploading time in each region would be cut in half in the ideal case, and the data gathering cycle can be dramatically shortened. Hence, applying the SDMA is quite suitable to data gathering in WSNs, achieving the spatial reuse further to advantage the effect of employing multiple MDCs.

This chapter focuses on data gathering with multiple MDCs and the SDMA technique, which involves a joint design of mobility control on the MDCs and the utilization of SDMA in sensor data transmissions. For each MDC in a region, the data gathering time in a tour mainly consists of two parts: data uploading time of the sensors in this region and moving time of the MDCs on the tour. The objective of this study is to minimize the data gathering time among different regions. In particular, answers are sought for several questions, including how to divide the sensing field into a certain number of regions to balance the data gathering time among different regions, how to better enjoy the benefit of SDMA to shorten the data uploading time in each region, and how to move each MDC on a short tour in each region. This chapter examines these issues and finds an effective solution to the proposed data-gathering problem.

The main contributions of this chapter can be summarized as follows: (i) it introduces a joint design of multiple mobile collectors and SDMA technique for data gathering in WSNs, (ii) it shows that the proposed data gathering scheme can radically resolve the non-uniformity of energy consumption among sensors by applying mobility control to achieve single-hop transmission between each sensor and its associated MDC, (iii) it demonstrates that the proposed data gathering scheme works well not only in a connected WSN, but also in a disconnected network, since MDCs play the role as virtual ″bridges″ to connect separated regions, (iv) the proposed data gathering scheme achieves much shorter data gathering time with respect to other mobile data gathering schemes due to the concurrent use of multiple MDCs and simultaneous data uploading among sensors by utilizing SDMA, and (v) a commercially appealing feature of the proposed data gathering scheme is that sensors can be kept as simple as before and all the intelligent operations are performed by MDCs.

6.4 Proposed Data Gathering Scheme

This section describes the definitions, assumptions and overview of the proposed data gathering scheme. Firstly, some terminologies used in this chapter are defined. This section then describes the model architecture of the sensing field where the proposed data gathering scheme will be applied.

6.4.1 Terminologies and definitions

This section provides the following terminologies used in this chapter and their definitions.

- ***Region***. Based on its size, the total area of the target field is divided into several regions. The created regions are non-overlapping, i.e., each sensor node belongs to only one region. The regions are created using an imaginary curved line that follows the *Voronoi edge*s. For example, in Figure 6.3, the target field is divided into two regions, separated by the red dotted curved line. In each region, one MDC would be operating for collecting data from all chain leaders of the region.
- ***Polling points***. In the proposed data gathering scheme, an MDC roams within each region and stops at some locations to collect data from the chain leaders.

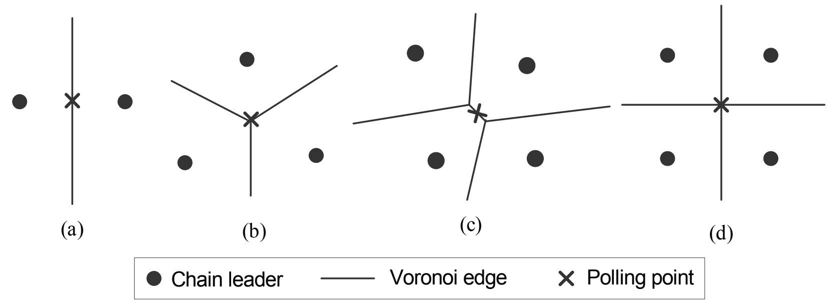


Figure 6.1: Polling points are marked on the *Voronoi edge*s.

These positions are called polling points. To take full advantage of the SDMA technique, polling points should be equidistant from the associated chain leaders. As the *Voronoi edge* between two points is always equidistant from the two points, the polling points can be measured by using the *Voronoi edge*s of the chain leaders. Figure 6.1 shows some positions of polling points in four different cases. If there are only two chain leaders, the position of the polling point can be found at the intersection between the *Voronoi edge* and the line joining the two chain leaders (Figure 6.1(a)). For more than two chain leaders, the polling point can be found at the intersection of different *Voronoi edge*s (Figure 6.1(b-d)).

- ***Coverage area of polling point***. The disk-like shaped area, centred at a polling point with the radius equal to the transmission range of a chain leader, is defined as the coverage area of a polling point. All the chain leaders in the coverage area of a polling point form the neighbour set of this polling point. Though a chain leader may be located in the coverage areas of multiple polling points, it is finally associated with only one polling point for data uploading.
- ***Compatible chain leaders***. Any two chain leaders associated with the same polling point are said to be compatible if an MDC arriving at this polling point can successfully decode the multiplexing signals concurrently transmitted from these two chain leaders. Note that, due to limited transmission power of sensors, not all chain leaders in the neighbour set of the same polling point are compatible. Detailed discussions on utilizing SDMA at physical layer for concurrent data uploading is provided in [Zhao et al., 2008]. Different association

patterns of sensors with the polling points correspond to different compatibility relationship among them because the channel state information varies with the association pattern. Two compatible sensors associated with the same polling point are qualified to be a compatible pair to upload data simultaneously, when the MDC arrives at this polling point.

6.4.2 Assumptions

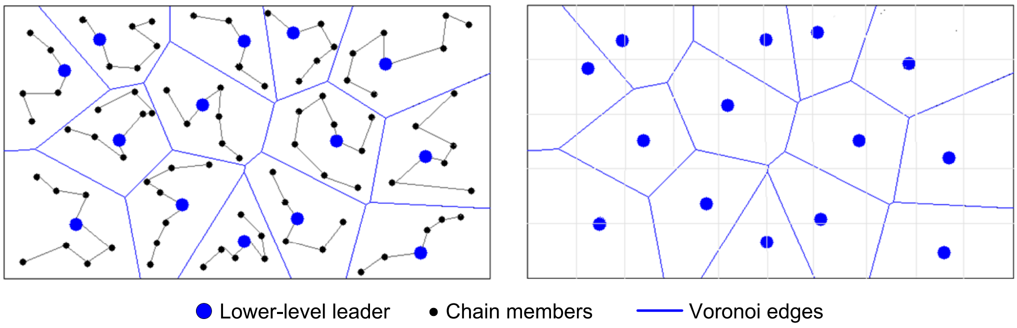
Apart from the general assumptions discussed in Section 2.5, the following assumptions are made specific to the proposed data gathering scheme.

- It is assumed that MDCs have access to a continuous power supply. Usually the base stations are equipped with the source of continuous power supply. Thus, when an MDC visits the base station, it can replace its battery.
- It is assumed that the MDCs are familiar with the target field. Location images of the target field can be stored in each MDC. Thus, an MDC is able to visit any point within the target field.
- It is also assumed that each MDC can forward the gathered data to one of the nearby MDCs when they are close enough, such that data can eventually be forwarded to the MDC that will visit the static data sink.

6.4.3 Overview of the proposed scheme

Recall that, in the proposed logical topology, each sensor is associated with a single chain and sends all its sensed data to or towards its chain leader. In other words, the chain leader of a chain collects data sensed by all member nodes of that chain. While MDCs are moving through the target field, they stop at the polling points to poll the nearby chain leaders to gather data packets that the chain leaders collected from their chain members. Thus, when an MDC collects data packets from a chain leader, it virtually collects data from all sensor nodes associated with that chain.

When an MDC moves to a polling point, it polls the nearby chain leaders with the same transmission power as that of the chain leaders, such that the chain leaders that receive the polling messages can upload packets to their associated MDC within a



(a) All sensor nodes (b) The chain leaders

Figure 6.2: Setup for data gathering from chain leaders.

single hop. Each MDC is equipped with two antennas, which means that at each time slot, up to two chain leaders can send data simultaneously to an MDC by utilizing the SDMA technique.

The operations of the data gathering tour of an MDC can be divided into two parts: traversal path of the MDC that specifies the movement of the MDC, and uploading of data, which specifies how the chain leaders interact with the MDC. Thus, the data gathering time in a region is the aggregation of the moving time of the MDC and data uploading time of the chain leaders in the region. An MDC arriving at a polling point in its region collects data from the associated chain leaders and then moves straight to the next polling point in the tour. Thus, the moving tour of an MDC consists of a number of polling points in its region and the line segments connecting them.

Figure 6.2 provides an example of the architecture of the proposed data gathering scheme. In Figure 6.2(a) all the sensors deployed in the target field are shown. The black dots are members of different chains, whereas the big blue dots are the chain leaders. Each chain is restricted in a region separated by the *Voronoi edge*s. In the proposed data gathering scheme, the chain leaders collect data from the chain members, whereas the MDCs collect data from the leaders only. Figure 6.2(b) shows only the chain leaders. Polling points would be calculated only with respect to these chain leaders.

Figure 6.3 shows a model of implementation of the proposed data gathering scheme. In this figure, two MDCs roam within the network and collect data from the chain

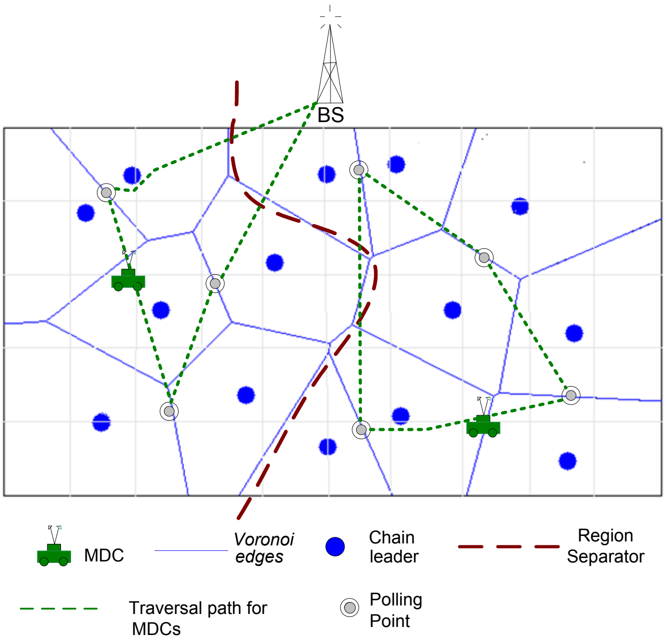


Figure 6.3: A model of implementation for the proposed data gathering scheme using two MDCs and SDMA technique.

leaders. The two MDCs work at the same time, and when an MDC arrives at a polling point, chain leaders associated with this polling point are scheduled to communicate with the MDC. Two chain leaders in a compatible pair can upload data simultaneously in a time slot, while an isolated chain leader (i.e., a chain leader by itself or not in any compatible pair) sends data to the MDC separately.

The optimum data gathering can be achieved by minimizing the maximum data gathering time in different regions. This problem is referred to as a data gathering problem, which consists of several tightly coupled sub problems, such as finding compatible pairs among chain leaders, determining the polling points and chain leader association patterns, dividing the polling points into several regions, scheduling the polling points, and determining the traversal paths for each MDC to visit the polling points in its region.

6.5 Detailed Description of the Proposed Scheme

This section describes the proposed data gathering scheme that employs multiple mobile data collectors and the SDMA technique. The total data collection scheme can be divided into four sequential steps, namely (i) to find the sets of compatible chain leaders, (ii) to determine polling points, (iii) to divide the target fields into regions in accordance with the polling points, and (iv) for each region, to determine the traversal paths of the MDCs. All these steps are described below.

The first step of the proposed data gathering scheme takes full advantage of the SDMA technique. To collect data as quickly as possible, chain leaders should be organized into the maximum number of compatible pairs for data uploading. This can be formalized as a matching problem in a *compatibility graph*, where each vertex represents a chain leader, and two vertices are adjacent to each other when the two chain leaders are compatible. In order to keep the graph simple, it is assumed that there is only one edge between two vertices even if the two corresponding chain leaders are compatible in the coverage area of different polling points. A group of compatible pairs corresponds to a set of vertex-disjoint edges in the graph. A set of vertex-disjoint edges is defined as a matching in graph theory [West, 2000]. Therefore, finding an optimum number of compatible pairs among all the chain leaders, which is the first step of the proposed data gathering scheme, is equivalent to finding a maximum matching in the compatibility graph. Optimum matching can be found by some existing algorithms, such as the Edmonds' Blossom Algorithm [Edmonds, 1965]. Figure 6.4 shows a compatibility graph as an example, where there is an edge between any two chain leaders, and they are compatible in the coverage area of some polling points.

The second step of the scheme is to determine the polling points. Since it is expected that the polling points result in short moving paths for MDCs, a small number of polling points that can achieve the maximum number of compatible pairs is preferable. As it is required that any two chain leaders in a compatible pair are associated with the same polling point, the neighbour sets of each polling point are updated first, based on the maximum compatible pairs obtained in the first step.

Assume a set of chain leaders $L = \{L_1, L_2, \ldots, L_k\}$, such that, for each i, the chain leader L_{i+1} is the closest from the chain leader L_i while L_i and L_{i+1} are compatible.

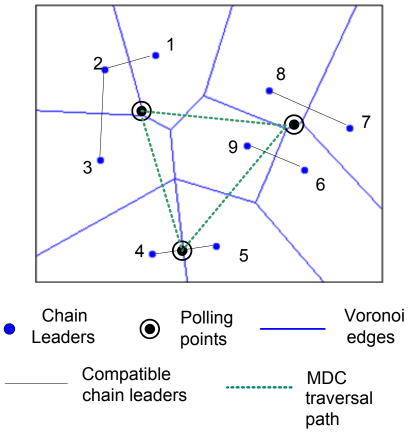


Figure 6.4: Determining the polling points.

Let P_p is the polling point for a set of chain leaders $L_1, L_2, \ldots, L_{p},$. Then, P_p should be equidistant from all chain leaders $L_1, L_2, \ldots, L_p$. In Figure 6.4, this is shown with an example. In this figure, there are nine chain leaders numbered from one to nine. Assume that the compatible sets of the chain leaders found after step one are {1, 2, 3}, {4, 5}, and {6, 7, 8, 9}. Thus, three polling points would be created. In the first poll point, an MDC would poll data from the chain leaders 1, 2 and 3. Similarly, the second poll point would be used for the chain leaders 4 and 5. The third and final polling point would be used by an MDC to collect data from the chain leaders 6, 7, 8 and 9. Note that all the polling points marked in Figure 6.4 are situated on the *Voronoi edge*s, and are equidistant from the corresponding chain leaders.

After finding the sets of compatible chain leaders and the polling points, in the next two steps, the polling points are divided into several regions, and then the sequences of the polling points and traversal paths from one polling point to another polling point are determined.

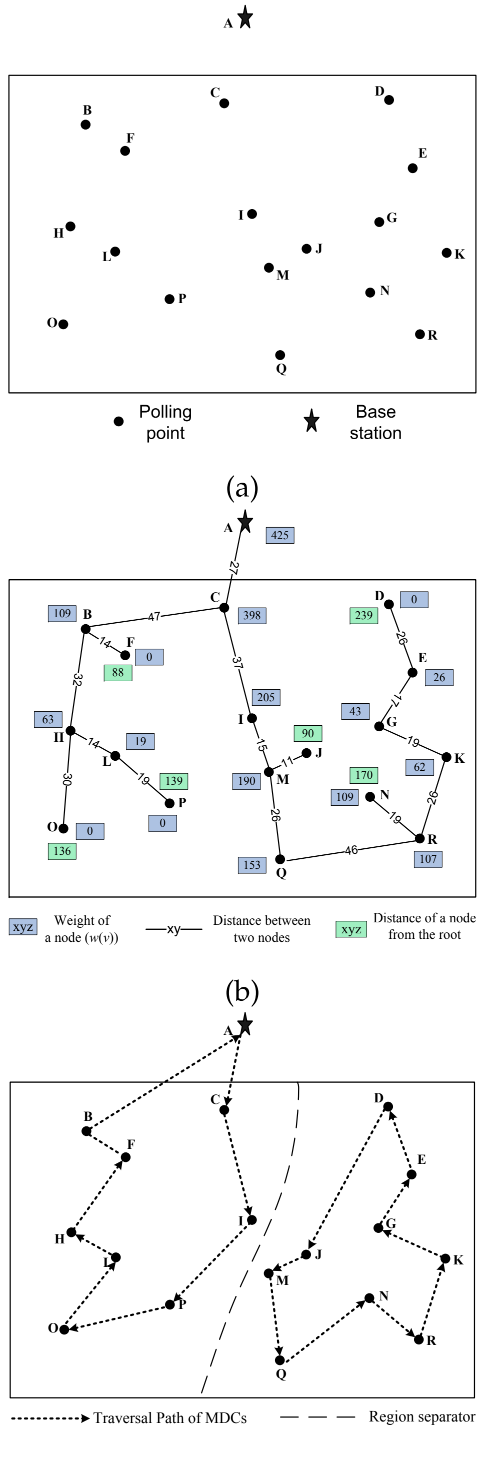
In the third step of the scheme, using the positions of the polling points, the target field is divided into several regions. In this process, the polling points are structured into a tree, and weights are assigned for each node of the tree. Specifically, a minimum spanning tree $T(V, E)$ is constructed using the polling points in P, where the base station is treated as the root of the tree, denoted by r_T. For a node v of the tree T, its weight, denoted by $w(v)$, is calculated using the following equation:

$$w(v) = \sum_{e \in E(subT(v))} L_e \tag{6.1}$$

where, $subT(v)$ is the sub-tree of T rooted at v, $V()$ and $E()$ represent the vertices and edges on the tree respectively, and L_e is the length of an edge e.

Now, the remaining problem is how to divide the polling points and their associated chain leaders into a certain number of parts, each for a region, to balance the data gathering time among these regions. Suppose that, $w(r_T) = W_T$, and there are a total of N_k number of MDCs, which means that the polling points will be divided among N_k regions. The basic idea is to partition the minimum spanning tree T into N_k parts, by iteratively finding a sub-tree t based on the weight of each vertex on T and pruning t from T. A variable m is defined to represent the current available MDCs in each stage of the algorithm and initialize it to N_k. To build a sub-tree, the following steps are followed: the procedure starts from the farthest leaf node v of T. The parent node of v, denoted by $PA(v)$, is inserted into the sub-tree t. Let, $u = PA(v)$ on T. The node u is considered as the root of the sub-tree t. This up-tracing process is repeated until $(w(u) + d(u, v)) \geq W_T/m$, where $d(u, v)$ is the distance between the current root (u) of the sub-tree and the node (v) where the procedure started. When the value of $(w(u) + d(u, v))$ crosses W_T/m, all the nodes of t are removed from T, which means that the corresponding selected polling points on t will be assigned to the same region (or to the same MDC). The values for T, W_T, m and $w(v)$s are updated. When there is a single MDC left, all the remaining selected polling points and their associated sensors are assigned to the MDC and the procedure terminates. The algorithm is illustrated in Figure 6.5 and the pseudo-codes are given in Figure 6.6.

For better understanding of the region division procedure, Figure 6.5 is used as an example. Figure 6.5(a) shows several polling points in a target field, and a base station. The base station is marked as A, whereas the polling points are marked as from B to R. All the polling points (B to R) would be divided among different regions of $R = \{R_1, R_2, R_3, \ldots - +\}$. Figure 6.5(b) shows a shortest-path-tree T, which is constructed by the polling points, using the base station A as the root of the tree T. For each node v of T, a weight, denoted by $w(v)$, is calculated by the total edge length of a sub-tree t, assuming the node v as the root of the sub-tree t. For each node of the tree T, the weight of the node is shown in the blue-coloured rectangle. For example, the weight of node B



(a)

(b)

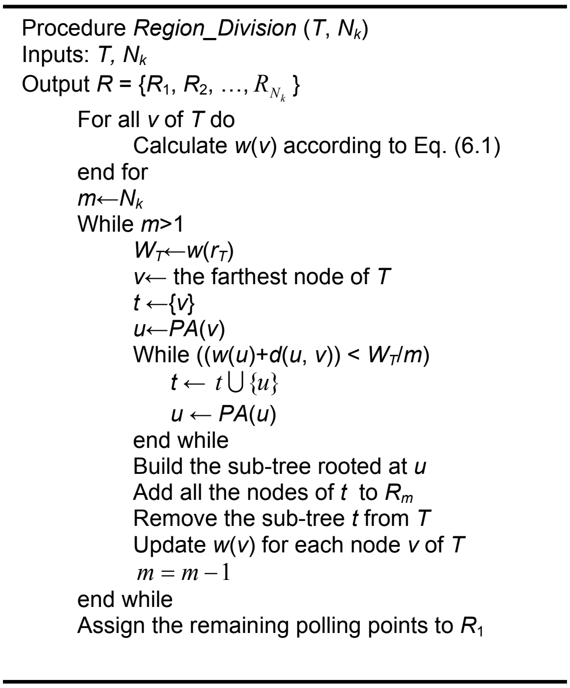
(c)

Figure 6.5: An example of the region creating process for MDCs.

is 109, which is calculated by adding the lengths of edges $BF(14)$, $BH(32)$, $HL(14)$, $HO(30)$, and $LP(19)$. Each of the leaf nodes of the tree T has an additional distance-from-root value, which is shown in the green-coloured rectangle. For example, the distance-from-root value of the node P is 139, which is calculated adding the lengths of the edges $AC(27)$, $CB(47)$, $BH(32)$, $HL(14)$, and $LP(19)$. Assume, for this example, the value of m is 2. That means the target field would be divided into two regions and two MDCs would be employed to gather data from the chain leaders. Among the leaf nodes D, F, J, N, P, and O, the leaf node D is the farthest node from the root A. Thus, the region division procedure starts from the node D. Note that the weight of the root node A is 425, m number of sub-trees are constructed, where the total traversal path completing a cycle by all nodes of each sub-tree would be around $(425/m)$. The node D is inserted into the sub-tree tm. The procedure then checks the weight of D's parent node E. As $(w(E) + d(D,E) = 52)$ is less than 212.5, the procedure traces the upper nodes, i.e., nodes G, K, R, Q, and M respectively. As M is the first node for which $(w(M) + d(D,M) = 254)$ exceeds 212.5, the procedure stops at this point, and all nodes of the sub-tree whose root would be M are assigned to region R_m. That means, for this example, where $m = 2$, region two would have the polling points D, E, G, K, R, N, Q, M, and J. The rest of the polling points (B, C, F, H, I, L, O, and P) would be in region one. Figure 6.5(c) shows two regions separated by a dashed line. This figure also shows the traversal paths for the MDCs. The discussion of determining the traversal paths is provided next.

In the fourth and last step, after partitioning the target area into different regions, and assigning the chain leaders and MDCs to each region, the traversal paths of the MDCs would be determined. The MDC of a region visits each polling point in its region exactly once before the MDC completes a cycle. Thus, finding the shortest tour of an MDC is equivalent to the Travelling Salesman Problem (TSP) [Cormen et al., 2001]. An approximation or heuristic algorithms for the TSP problem can be used to determine the moving tour for each MDC. As a result, the moving tours of the two MDCs for the example of Figure 6.5(c) are: $PATH_{MDC1} = D \rightarrow J \rightarrow M \rightarrow Q \rightarrow N \rightarrow R \rightarrow K \rightarrow G \rightarrow E \rightarrow D$ and $PATH_{MDC2} = A \rightarrow C \rightarrow I \rightarrow L \rightarrow P \rightarrow O \rightarrow H \rightarrow F \rightarrow B \rightarrow A$.

In determining the path from one node to the next node, the presence of any obstacles should also be considered, because, in most real-world applications, the work-



```
Procedure Region_Division ($T$, $N_k$)
Inputs: $T$, $N_k$
Output $R = \{R_1, R_2, \ldots, R_{N_k}\}$
    For all $v$ of $T$ do
        Calculate $w(v)$ according to Eq. (6.1)
    end for
    $m \leftarrow N_k$
    While $m>1$
        $W_T \leftarrow w(r_T)$
        $v \leftarrow$ the farthest node of $T$
        $t \leftarrow \{v\}$
        $u \leftarrow PA(v)$
        While $((w(u)+d(u, v)) < W_T/m)$
            $t \leftarrow t \bigcup \{u\}$
            $u \leftarrow PA(u)$
        end while
        Build the sub-tree rooted at $u$
        Add all the nodes of $t$ to $R_m$
        Remove the sub-tree $t$ from $T$
        Update $w(v)$ for each node $v$ of $T$
        $m = m - 1$
    end while
    Assign the remaining polling points to $R_1$
```

Figure 6.6: Procedure of dividing polling points into N_k parts.

ing areas may be partially bounded, or have some irregular-shaped obstacles located within the sensing area. In order to make the moving path-planning algorithm feasible in these situations, MDCs have to be able to avoid the obstacles. It was assumed that the complete map of the sensing field is obtained before an MDC begins collecting data, which should include the location and shape information of the obstacles in the sensing field. Once this is done, it is not difficult to adjust the basic moving path-planning algorithm to avoid obstacles. For each candidate location of a turning point, each MDC checks if the line segment from the last turning point to it, and the line segment from it to the next turning point are blocked by the obstacles. If so, the candidate location is then not eligible to be the turning point.

Figure 6.7 shows an example of how it can be checked for the eligibility of each possible location of the turning point. A new path from point A to point B will be chosen from $A \rightarrow 1 \rightarrow B$ or $A \rightarrow 2 \rightarrow B$. Since the straight lines between A and 2, and between 2 and A, are blocked by obstacles, 2 is not eligible to be a tuning point. Thus, the new path from A to B can only go through point 1.

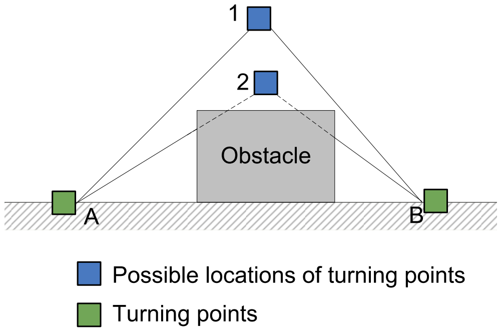


Figure 6.7: Planning the moving path in the sensing field with obstacles. The line segments from A to 2 and from 2 to B are blocked by obstacles. Thus, 2 cannot be a turning point, whereas 1 is eligible to be a turning point.

6.6 Analysis and Simulation Results

The proposed protocol uses *probe message*s to find out the lowest level chain of the destination node. *Probe message*s are passed from a lower-layer to a higher layer in search of the lowest-level leader of the destination node. For a large network of n-tiers ((in a network of 10,000 nodes value of n would be 3 or a maximum of 4)), although, there might be a large number of probe message generated (based on the location of the destination node), the cost associated with *probe message*s exchanges is not that high, because probe messages are very short in length. Furthermore, once the position of the destination node is known, there is no need to send any further *probe message*s for that destination node. Although it incurs some buffer expenses, this is a good approach, in particular when many sensor nodes frequently try to send data to a specific node.

To further analyse the proposed data gathering scheme, and to evaluate the performance of the proposed scheme, various simulation experiments were performed. The detailed descriptions of these experiments and their results are discussed below.

The first experiment analyses the proposed data gathering scheme. In this simulation experiment, the sensing field was considered an area of $100m \times 100m$, where a total of N_s sensors ($N_s = 100 - 1500$) were randomly distributed. Depending on the number of sensor nodes, N_p polling points ($N_p = 5 - 30$) were located on the *Voronoi edge*s. The transmission range of each sensor was assumed $r = 30m$. It was also assumed that the size of the sensing data q in each sensor is 1Mb, the effective data

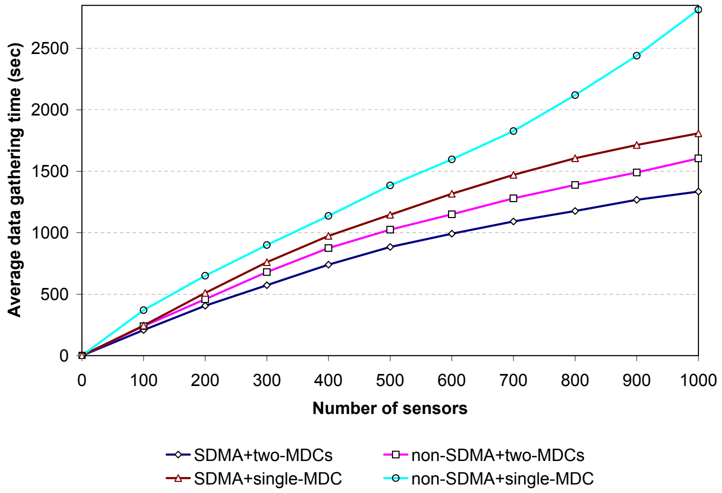


Figure 6.8: Data gathering time as a function of N_s under different settings of N_k.

uploading rate of each chain leader $v_d = 80$ kbps,[1] and the moving velocity of each MDC $v_m = 0.8$ m/s[2], if not stated otherwise.

Figure 6.8 plots the data gathering time over the sensing field by different schemes when N_s varies from 100 to 1000. In the simulation, four different mobility category were compared, namely i) without SDMA and with a single MDC (non-SDMA + single-MDC), ii) with SDMA and with a single MDC (SDMA + single-MDC), iii) without SDMA and with two MDCs (non-SDMA + two-MDCs), and iv) with SDMA and with two MDCs (SDMA + two-MDCs). Note that, category (iv) is shown as an example of the proposed scheme. When multiple MDCs are used, data gathering time refers to the maximum time of a data gathering tour among different regions. It can be seen that data gathering time of all the schemes increases as N_s increases. However, the proposed scheme always outperforms other schemes due to the concurrent use of multiple collectors and simultaneous data uploading among sensors with the support of the SDMA technique. For instance, it achieves 56% time saving compared to non-SDMA+single-MDC scheme when $N_s = 1000$. Shorter data gathering time leads to longer network lifetime, because sensors can turn to power-saving mode once the data gathering in their region is done, and also leads to shorter latency among the

[1] v_d is the data uploading rate of each chain leader: whenever an MDC reaches near a chain leader, the leader node uploads its data packets to the chain leader with the speed of v_d.

[2] v_m is the velocity of an MDC: the speed with which an MDC moves along its traversal path depicted in Figure 6.4

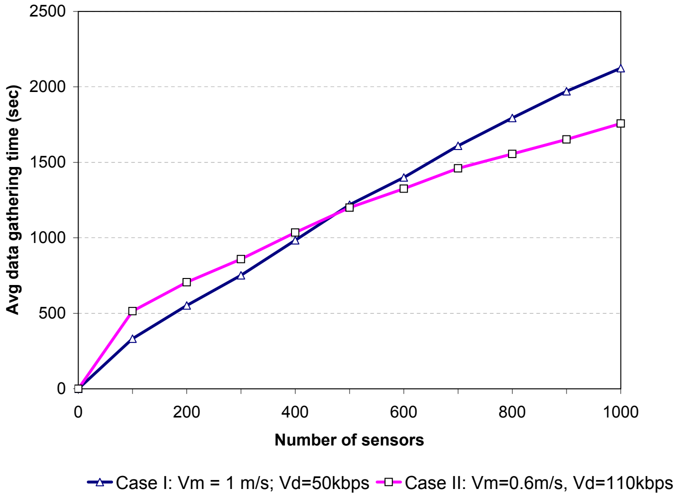


Figure 6.9: Data gathering time under different settings of v_d and v_m.

sensing data. It is also noticed that the advantages of the proposed scheme over other schemes become more evident when the network becomes denser with more sensors. This is reasonable because more sensors would provide more opportunities to utilize SDMA for concurrent data uploading.

Figure 6.9 shows that data gathering time of the proposed scheme varies with N_s under different settings of v_m and v_d. There were 30 polling points and two available MDCs. Two configurations of (v_m, v_d) were considered, which are ($v_m = 1$ m/s, $v_d = 50$ kbps) and ($v_m = 0.6$ m/s, $v_d = 110$ kbps) to represent two different cases. It was noticed that, when N_s was small, the moving velocity of an MDC had a greater impact on data gathering time than v_d. Higher moving velocity, such as 1m/s in Case I, resulted in shorter data gathering time even with a smaller v_d than the other case. It is reasonable since the moving time of each MDC is dominant when sensors are sparsely scattered. On the contrary, when N_s was large, the effect of v_d on data gathering time of an MDC overwhelmed that of v_m. For example, when N_s increased to more than 60, the data gathering time for Case II, which had higher effective data uploading rate as $v_d = 110$ kbps, was smaller than that of Case I. This is because more chain leaders make data uploading time dominant in each region and they provide more opportunity to extract the benefit of SDMA technique to the maximum extent.

The following two experiments were performed to evaluate the efficiency of the proposed data gathering scheme. The results of the first experiment show the en-

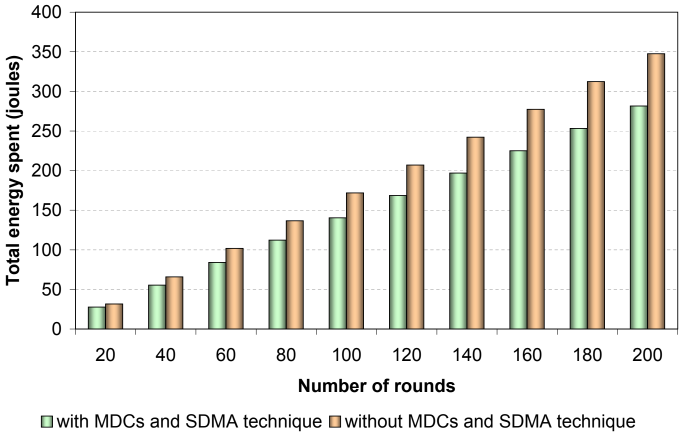
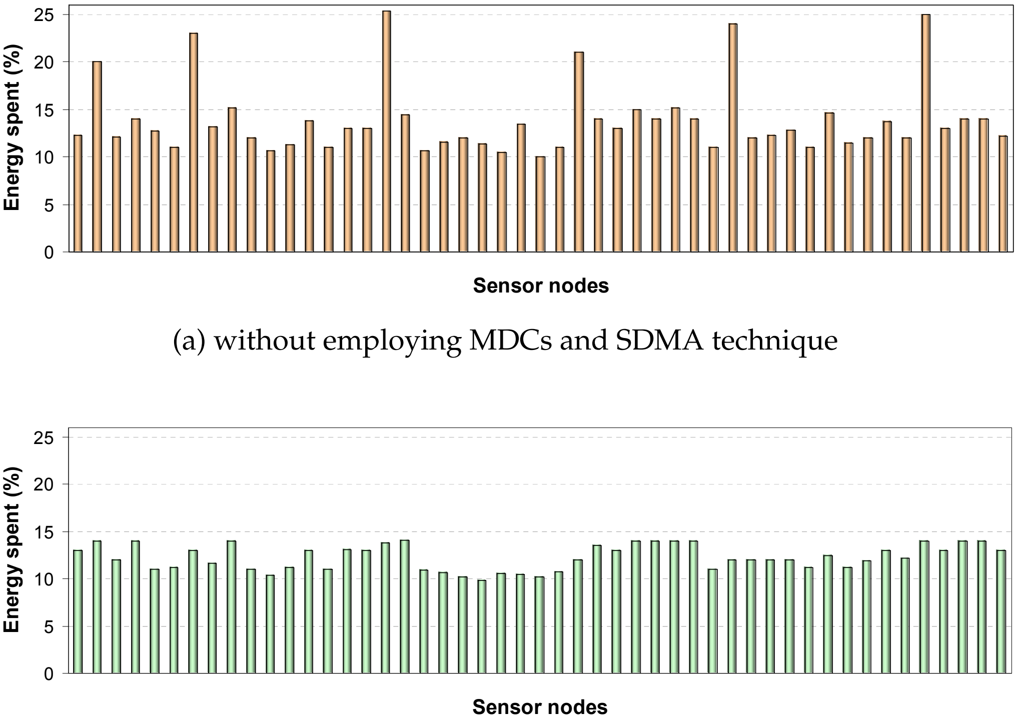


Figure 6.10: Total energy spent comparison for the proposed data gathering scheme.

ergy efficiency of the proposed scheme, whereas the results of the second experiment demonstrate the uniformity of energy consumption by all sensors of the network. For these two experiments, it was assumed that 200 sensor nodes were deployed in a target field of $100m \times 100m$. The initial energy of each sensor node was assumed to be 1 joule. The base station was assumed to be located at the coordinates of (50, 150). These 100 nodes constructed twelve chains using the *Voronoi diagram* based chain construction algorithm discussed in Chapter 5. The experiment measured the total energy consumption of the network with and without employing MDCs and the SDMA technique after every 20 data collection rounds.

Figure 6.10 shows the energy saving performance of the proposed scheme. It is obvious from the figure that the proposed data gathering scheme saves a significant amount of energy by allowing the chain leaders not to send data to the base station or among themselves. Using the proposed data gathering scheme the system can save around 10% of total energy after 50 rounds, 14% of total energy after 100 rounds, and 17.5% of total energy after 200 rounds, and so on.

The proposed data gathering scheme not only saves energy consumption, but also makes sure that energy consumption by the nodes are evenly distributed. There is no point saving total energy consumption without ensuring the uniformity of energy consumption, because uneven energy consumption adversely affects the system lifetime. The second experiment for evaluating the performance measured the uniformity of energy consumption. In doing so, energy spent by each sensor node of the



(a) without employing MDCs and SDMA technique

(b) with employing MDCs and SDMA technique

Figure 6.11: Energy dissipation comparison for the proposed data gathering scheme. (after 150 rounds of data collection by 200 sensor nodes in an area or $100m \times 100m$).

network was calculated and compared. Figure 6.11 shows the percentage of energy spent by the sensor nodes after 150 rounds of data collection. Due to space limitation, Figure 6.11 shows energy consumption of 50 nodes among the 200 nodes of the experimental setup. Figure 6.11(a) shows the energy consumption of sensor nodes without employing the proposed data gathering technique. Note that few nodes (especially the chain leaders) spend comparatively higher energy than other sensor nodes. The difference between the highest and lowest energy consumption is 15.37% and the standard deviation is 3.77. On the other hand, Figure 6.11(b) shows uniform energy distribution by all sensor nodes. In this case, the difference between the highest and lowest energy consumption is only 04.27% and the standard deviation is 1.34. The uniform energy distribution resulting from the proposed data gathering scheme would directly help the WSN for longer lifetime. Without the employing the MDCs and the SDMA technique, it was found that, the first node of the network would die after 595 rounds of data collection. On the other hand, the first node of the network would die after

1070 rounds of data collection if multiple MDCs and the SDMA technique were used for data collection. Thus, using the definition of network lifetime as the time until the first sensor node dies, it can be argued that the proposed data gathering/collection scheme increases the lifetime of the network by 80%. On the other hand, using the definition of network lifetime as the time until all the sensor nodes die, the proposed scheme increases the network lifetime by 45% (780 rounds without MDCs and the SDMA technique, and 1130 rounds with multiple MDCs and the SDMA technique.)

6.7 Summary

In this chapter, a new data collecting scheme is proposed for chain oriented sensor networks. The proposed scheme employs multiple mobile data collectors, called MDCs, and uses the spatial division multiple access (SDMA) technique. The total data collection process works as follows. Each of the deployed sensor nodes, which is not a chain leader, sends its sensed data to or towards its leader node of the chain, while an MDC collects data from the chain leaders of the same region. MDCs work like mobile base stations. An MDC starts the data gathering tour from a polling point or the base station, traverses a region of the sensor network, stops at several polling points inside its region, collects the data from nearby chain leaders, and then returns to the starting polling point or the base station, and finally uploads the collected data to the base station. Detailed description for determining the polling positions is provided in this chapter, and a heuristic algorithm for planning the moving path/circle of MDCs is presented. The tour planning algorithm can be used both in connected networks and disconnected networks. Thus, the data gathering scheme also ensures the connectedness of WSNs. It is also shown in the chapter that the proposed data collection scheme affects the network lifetime significantly by saving a large amount of energy. The simulation results show that the proposed data gathering mechanism can prolong the network lifetime by 45% to 80%, compared to a network that has only a static base station situated outside the target field.

The design of the basic multi-chain oriented logical topology and its three adaptations are completed with this chapter. In the next chapter, a few application protocols are designed using the proposed topology to verify the claim that topology design should come first before the protocol design.

Chapter 7

Application Protocols Using the Proposed Topology

7.1 Preamble

This thesis argues that logical topology can play a vital role in designing various protocols for WSNs, and a well-designed logical topology facilitates the better designing of different application protocols. In Chapter 3, a multi-chain oriented logical topology has been proposed, and in Chapters 4, 5, and 6 this logical topology has further been extended by three adaptations. This thesis also argues that the logical topology of a WSN should be constructed first, and then the protocol design should take place. To justify this concept, in this chapter, a number of application protocols are designed using the proposed multi-chain oriented logical topology. The aim of this chapter is to evaluate the performance of the protocols to demonstrate how a well-designed logical topology influences the designing of other protocols. In doing so, the logical structure and the communication abstraction of the logical topology are used in designing various application protocols for WSNs, and these application protocols are then applied on the top of the proposed logical topology. Figure 7.1 shows the relationships among logical topology, its communication abstraction, and different prospective protocols.

In choosing the protocols to be designed, different types of applications of WSNs are studied. It is found that WSNs are very much data oriented. Almost all WSNs, if not all, are mainly used for data collection, data dissemination, and data transfer purposes. One of the most important applications of WSNs is sensor data collection, where sensed data are continuously/periodically collected at all or some of the sensor nodes, and forwarded to a remote BS for further processing [Tang and Xu, 2008;

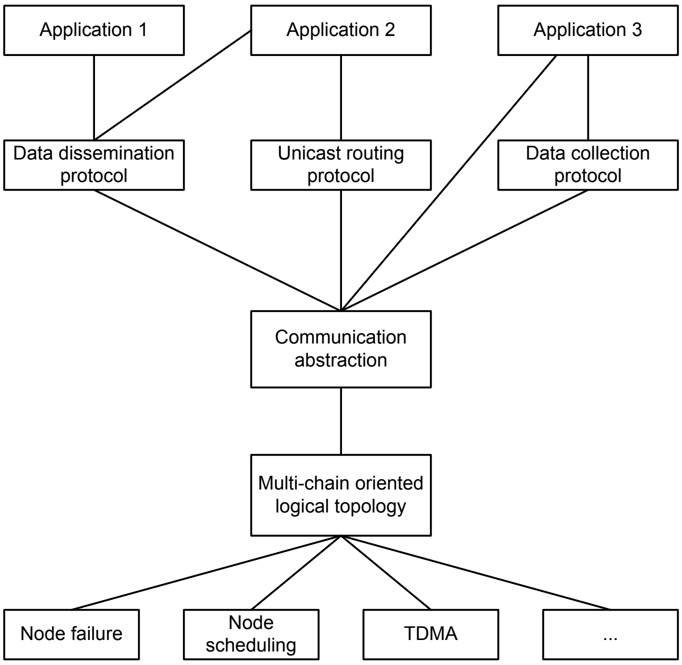


Figure 7.1: Designing application protocols using the proposed logical topology, and its communication abstraction.

Jin et al., 2010]. In Chapter 3, the communication abstraction of the proposed logical topology was described, and this communication abstraction can directly be used to collect data from the target field. Furthermore, in Chapter 6, a data collection scheme was proposed and designed using mobile data collectors. This scheme can be used when mobile collectors are available and viable to deploy. For this reason, no additional data collection protocol is required.

This chapter, therefore, discusses and proposes protocols for data dissemination and data transfer in WSNs. Data dissemination protocols are used to disseminate or distribute data/code/information among all or some of the sensor nodes deployed in the target field [Zhang and Wang, 2008; Wu et al., 2009; Rossi et al., 2010]. On the other hand, data transfer protocols are used to transfer a piece of data from the source sensor node to the destination sensor node [Poojary and Pai, 2010; Datta and Kundu, 2007; Shakshuki et al., 2006]. For this instance too, a protocol is proposed and applied on the top of the proposed logical topology.

The rest of the chapter is organized as follows. Section 7.2 proposes a secret key management protocol. The protocol generates secret keys and then disseminate those keys to the sensor nodes, so that a sensor node can communicate with other sensors securely. Although this is a key management protocol, the protocol disseminates information/data to all nodes in the target field. Thus, this protocol can be regarded as a data dissemination protocol. Section 7.3 proposes a data transfer protocol, which deals with a single sensor node as source and destination. For each of these two protocols, the related works, protocol descriptions, performance evaluation, and simulation results are provided. Finally, this chapter concludes with a summary in Section 7.4.

7.2 Protocol 1: Key Management Protocol (Data Dissemination Protocol)

The aim of this protocol is to disseminate particular data/information to all deployed sensor nodes. This section describes a protocol for disseminating secret keys to all sensor nodes, so that the sensor nodes can communicate to each other securely in a hostile environment. Thus, this protocol can also be called a secured key management protocol. The following section, therefore, describes the key management process in WSNs.

7.2.1 Key Management Process in WSNs

Nowadays, security is one of the key concerns for WSNs [Du and Chen, 2008]. This is because of the envisioned growth in utilizing sensor networks in a wide variety of environmwnt, which are not benign. Most of the usage areas of WSNs are sensitive, and thus prone to different kinds of attacks. A secured key management protocol is the pre-requisite for secured WSNs.

Key management is the process by which cryptographic keys are generated, stored, protected, transferred, loaded, used, and destroyed. The main objective of key management is to establish and maintain secure channels among communicating parties [Lee et al., 2007]. Typically, key management schemes use administrative keys for the secure and efficient (re-)distribution, and at times, generation of the secure channel communication keys to the communicating parties. Communication keys may be pair-wise keys used to secure a communication channel between two nodes that are in direct or indirect communications [Eschenauer and Gligor, 2002; Chan et al., 2003;

Du et al., 2004; Liu and Ning, 2005], or they may be group keys shared by multiple nodes [Eltoweissy et al., 2005; Younis et al., 2006]. Network keys (both administrative and communication keys) may need to be changed (re-keyed) to maintain secrecy and resilience to attacks, failures, or network topology changes.

Key management entails the basic functions of analysis, assignment, generation, and distribution [Younis et al., 2006; Eltoweissy et al., 2006]. During key analysis, keying requirements are analysed to determine the required number of keys for the network as well as the number of keys needed by each node. In addition, analysis may take place to determine keys that need updating. Next, key assignment is performed. Key assignment refers to the mapping of keys to different parties. Administrative key assignment is considered here since communication keys are simply assigned by agreement of parties wanting to establish a secure communication channel. In the third step, generation of administrative keys may take place once or multiple times over the lifespan of the network. The generation of communication keys is the responsibility of the communicating parties. In all cases, the key generating node(s) must be trusted by all key-receiving nodes. The fourth and last step, named as key distribution, refers to the delivery of keys to their designated nodes after they have been generated and assigned to the nodes. The distribution of communication keys usually takes place after the network has been deployed. Communication keys are used for a short period and should be regularly updated (this may include analysis, assignment, generation, and [re]distribution) of network keys. Re-keying essentially comprises these basic functions providing acceptable levels of security and conserving scarce resources, in particular energy, needed for network operations. The next section categorizes the existing key management protocols in WSNs, and identifies the scope of the proposed protocol.

7.2.2 Existing works for the key management process

Traditionally, the four basic functions discussed in the previous section, have been tightly coupled, where all were performed by a centralized server or collaboratively by the nodes in a network, with each node performing the same functions. Existing proposals, [Eschenauer and Gligor, 2002; Liu and Ning, 2005; Eltoweissy et al., 2005; Younis et al., 2006; Mamun et al., 2008], have moved to decouple these functions to various degrees. Such decoupling can immensely benefit sensor networks due to their large scale, high vulnerability to attacks, and limited resources. Keying functions are triggered by keying events. These events include network deployment, node addition, node eviction, or periodic (or on-demand) key refresh. Entities with key management responsibilities may include a key server, BS, gateway nodes, or even sensor nodes.

Several key distribution schemes have been proposed for WSNs. Key management schemes in sensor networks can be classified broadly into dynamic [Eltoweissy et al., 2006] or static [Eschenauer and Gligor, 2002; Chan et al., 2003; Liu and Ning, 2005] solutions based on whether re-keying (update) of administrative keys is enabled post network deployment. Schemes can also be classified into homogeneous [Eschenauer and Gligor, 2002; Liu and Ning, 2005], or heterogeneous [Hill et al., 2000] schemes with regard to the role of network nodes in the key management process. All nodes in a homogeneous scheme perform the same functionality. On the other hand, nodes in a heterogeneous scheme are assigned different roles. Homogeneous schemes generally assume a flat network model, while heterogeneous schemes are intended for both flat and clustered networks. Other classification criteria include whether nodes are anonymous or have pre-deployment knowledge (location, degree of hostility, etc.) imparted to the nodes. Another classification criteria includes asymmetric (public key) cryptography or symmetric cryptography. Although there are some works done in [Malan et al., 2004; Lopez, 2006; Huang et al., 2004] to customize public key cryptography and elliptic key cryptography for low-power devices, such approaches are still considered as costly due to high processing requirements. The last classification criterion includes pre-distribution [Eschenauer and Gligor, 2002; Chan et al., 2003; Liu and Ning, 2005; Ren et al., 2005; Kausar and Masood, 2006] or post distribution of secret keys. Recent research suggests that symmetric secret key pre-distribution is possibly the only practical approach for establishing secure channels among sensor nodes [Ren et al., 2005].

The milestone protocol for secret key management in WSNs was developed by Eschenauer and Gligor [2002]. This protocol uses a probabilistic key pre-distribution technique to bootstrap the initial trust between sensor nodes. The main idea is to have each sensor randomly pick a set of keys from a key pool before deployment. Then, in order to establish a pair-wise key, two sensor nodes only need to identify the common keys that they share. To bootstrap security using Eschenauer and Gligor's original scheme, a network goes through three phases. In the first phase (key pre-distribution), which takes place prior to network deployment, a large pool of *S* keys and their *ID*s are generated. Each node is then assigned a ring of *m* keys, drawn from the pool at random, without replacement. In the second phase (shared-key discovery), which takes place during network setup, all nodes broadcast the *ID*s of the keys on their key rings. Through these broadcasts, a node finds out with which of their neighbours (as determined by communication range) they share a key. These keys can then be used for establishing secure links between the two neighbours. Finally, during the path-key establishment phase, pairs of neighbouring nodes that do not share a key can set up their own keys, as long as they are connected by two or more secure links at the end of shared key discovery. Because of the way keys are assigned, a key can be found in more than two nodes, and used in multiple communication links. When a node is compromised, all its keys, and all the links secured by these keys are also compromised. The Eschenauer-Gligor scheme is further improved by Chan, Perrig and Song Chan et al. [2003], by Hung-Min et al. Sun et al. [2009], and by Liu and Ning Liu et al. [2005].

The proposed key management protocol uses the proposed multi-chain oriented logical topology, and can be classified by the characteristics of i) dynamic nature in re-keying of network keys (both administrative and communication), ii) heterogeneity as different nodes (member nodes, lower-level leaders, BS) perform in different ways, iii) being anonymous as the scheme does not assume any pre-deployment knowledge like location etc., iv) symmetric cryptography, and v) pre-distribution of partial keys.

7.2.3 Overview of the proposed key management protocol

The key management protocol proposed in this chapter is based on partial keys pre-distribution and symmetric cryptography. Because of the resource-constraint nature of WSNs, both pre-distribution of keys and symmetric cryptography are appropriate for WSNs.

The protocol proposes that each of the sensors stores a set of partial key (half-keys) rather than the set of full keys. This has two-fold advantages: i) lower storage requirement and ii) even if a sensor is captured by an attacker it cannot obtain the encryption / decryption keys. Two neighbouring sensors in a chain establish their encryption/decryption key by concatenating the partial keys.

In the proposed protocol, keys are not assigned randomly from a key-pool as in [Eschenauer and Gligor, 2002] and [Kausar and Masood, 2006]. Consequently, the number of keys generated is much lower when compared with [Eschenauer and Gligor, 2002] and [Kausar and Masood, 2006]. Nonetheless, the key management system remains secure, because a large number of keys can be generated by the sensors participating in a chain, and each pair of sensor nodes use a different communication key. Another important feature of the scheme is that two communicating nodes always use a new secret key for data encryption/decryption in each round. This feature enables WSNs to achieve resilience to attacks, as well as data freshness without generating a long nonce.[1]

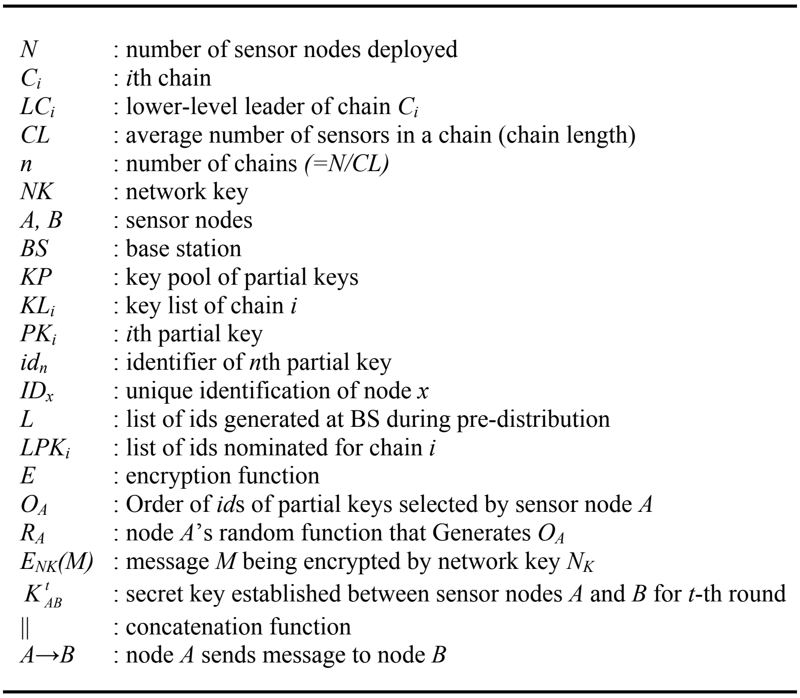
Perrig et al. [2002] point out that public key cryptography is not well suited for securing WSNs. Indeed, the memory of a sensor is typically insufficient to hold the long keys necessary to guarantee secure asymmetric cryptography. Moreover, sensors are usually equipped with processors that require high energy and time to compute the modular exponentiations involved in the implementation of public key cryptography. Therefore, symmetric cryptography is used for the proposed key management protocol. A symmetric cryptography can be defined as follows: $\acute{\mu} = E_k(\mu)$ is an encrypted message where μ is the plain text, k is the secret key, and E is the encryption algorithm. Accordingly, $\mu = E_k^{-1}(\acute{\mu})$ is the decryption of the same message μ.

Two versions of the key management protocol are proposed. In the first version of the protocol, all sensor nodes in a chain use the same set of partial keys, while in the second version each node uses a private copy of partial keys. The detailed description of the proposed protocol is provided in the next section.

7.2.4 Detailed description of the proposed key management protocol

This section describes the proposed key management protocol in detail. In describing the proposed protocol, various notations are used. Figure 7.2 lists these notations. Figure 7.3 describes the key management protocol as a whole. For easy understand-

[1] A nonce is a cryptographic value that is used only once.



N	: number of sensor nodes deployed
C_i	: *i*th chain
LC_i	: lower-level leader of chain C_i
CL	: average number of sensors in a chain (chain length)
n	: number of chains *(=N/CL)*
NK	: network key
A, B	: sensor nodes
BS	: base station
KP	: key pool of partial keys
KL_i	: key list of chain *i*
PK_i	: *i*th partial key
id_n	: identifier of *n*th partial key
ID_x	: unique identification of node *x*
L	: list of ids generated at BS during pre-distribution
LPK_i	: list of ids nominated for chain *i*
E	: encryption function
O_A	: Order of *id*s of partial keys selected by sensor node *A*
R_A	: node *A*'s random function that Generates O_A
$E_{NK}(M)$	: message *M* being encrypted by network key N_K
K^t_{AB}	: secret key established between sensor nodes *A* and *B* for *t*-th round
$\|\|$	: concatenation function
$A \rightarrow B$	: node *A* sends message to node *B*

Figure 7.2: Notation used in the proposed key management protocol.

ing, the protocol is divided into several steps which are also marked in Figure 7.3. The steps are described below.

Step 1 : Pre-distribution

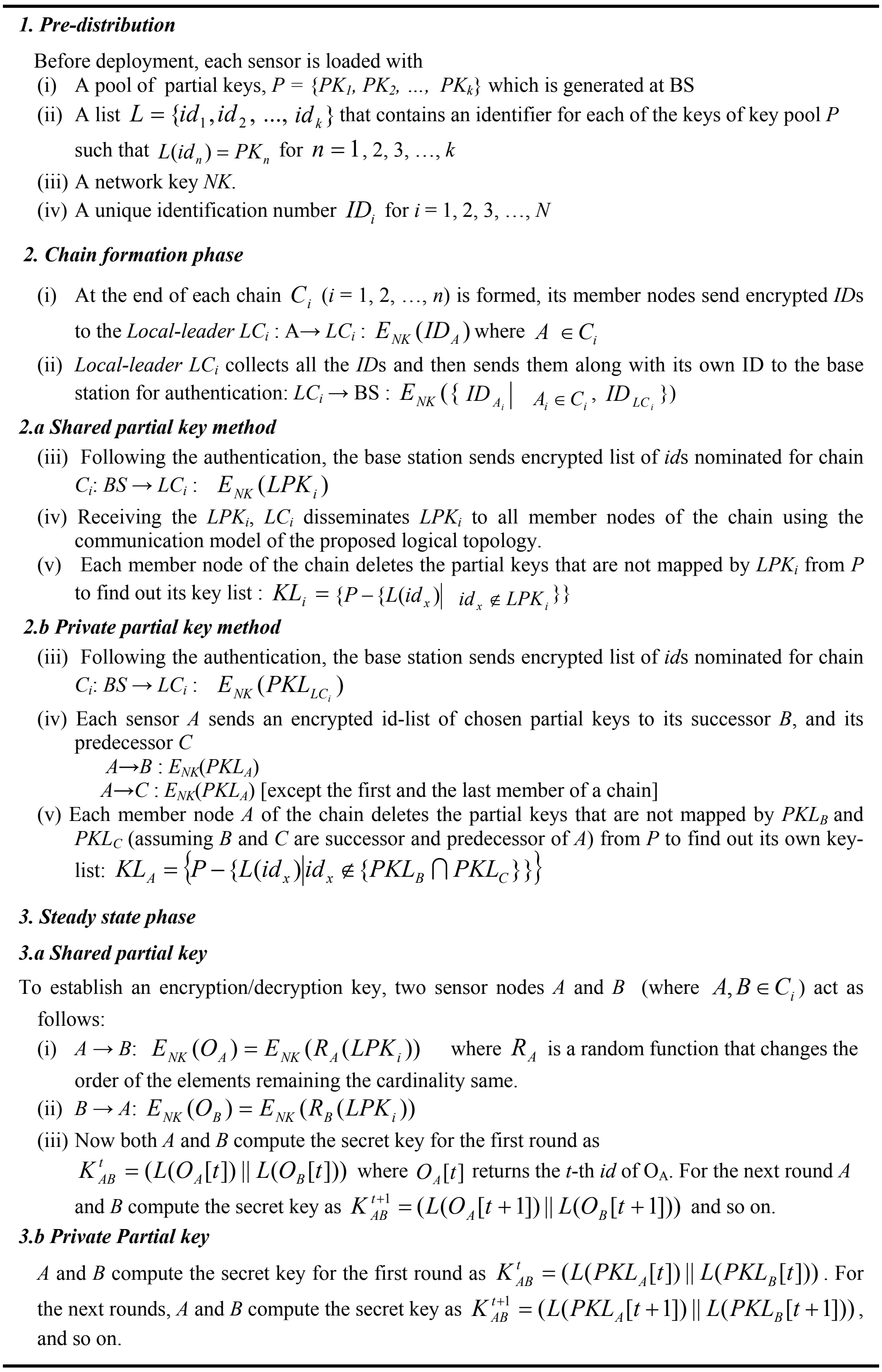
A key pool of partial keys is generated at the BS prior to the deployment of sensors. The size of the key pool is an important factor to determine. Considering that, an encryption / decryption key should be 128 bits long and the key pool contains 1000 partial keys of 64 bits long, the total memory consumption for storing all the partial keys in a sensor is $((128/2) \times 1000)$ bits or 8KB. It should be noted that, the Berkeley Mica Motes (one of the oldest sensors) has 128K bytes of program storage, and 4K bytes of SRAM. Although the proposed protocol forces each sensor node to consume all the partial keys generated at BS, soon after the first chain formation phase a sensor can delete all the partial keys except the nominated keys for it. Thus, if a sensor stores 50 half keys, it requires only 400 bytes to store the partial keys. Note that, using n partial keys, two neighbouring nodes can establish up to $2 \times n^2$ secret session keys. The number of session key candidates is one of the important performance evaluation metrics. Number of session key candidates simply refers to the minimum number of session keys possibly being created for data encryption/decryption. Before deployment, each sensor is loaded with a key pool of partial keys S, a list of identifiers of the partial keys L, a single network key N_K, and a unique identifier ID. Note that, if a key pool contains η number of partial keys, $\log_2 \eta$ bits are required for identifying each of the partial keys. Thus, a sensor node is loaded with around 8.7KB of partial keys and their identifiers. Soon after the chain formation, each node requires 444 bytes (assuming each node keeps 50 partial keys and deletes the rest).

Step 2: Chain formation phase

The single network key and the unique identifier of each node are used for distributing the administrative keys securely and authentication purposes respectively. During the chain formation phase of the proposed multi-chain oriented logical topology, the proposed key management protocol authenticates each sensor in a chain. Most of the sensor networks are used to protect or monitor critical infrastructures. In such structural monitoring applications, it would be a reasonable assumption that the sensor field is under super surveillance only during the deployment phase, which usually does not last too long [Zhang et al., 2005]. Thus, it can be assumed that adversaries do not actively catch or attack individual sensor nodes in this phase because otherwise they would run a high risk of exposing themselves. Therefore, a single network key can be assumed sufficient in the early deployment phase. However, an adversary might send strong signals as *HELLO* messages to tempt a sensor to consider it as its neighbour, and, therefore, becomes a member of a chain. For this reason, in the proposed protocol, a three-tier authentication check is performed after forming a chain. All members of the chain send their *ID*s (encrypted by the network key) to the leader of the chain. After collecting all the *ID*s, the local-leader sends all the *ID*s along with its own *ID* to the BS for authentication.

Steps 2a, and 3a (shared partial keys)

After the BS authenticates all members of a chain, it randomly chooses a predetermined number of partial keys and constructs an LPK using the *ID*s of those partial keys. BS then sends the *LPK* (encrypted by the network key) to the lower-level leader



1. Pre-distribution

Before deployment, each sensor is loaded with

(i) A pool of partial keys, $P = \{PK_1, PK_2, ..., PK_k\}$ which is generated at BS

(ii) A list $L = \{id_1, id_2, ..., id_k\}$ that contains an identifier for each of the keys of key pool P such that $L(id_n) = PK_n$ for $n = 1, 2, 3, ..., k$

(iii) A network key *NK*.

(iv) A unique identification number ID_i for $i = 1, 2, 3, ..., N$

2. Chain formation phase

(i) At the end of each chain C_i ($i = 1, 2, ..., n$) is formed, its member nodes send encrypted *ID*s to the *Local-leader* LC_i : A→ LC_i : $E_{NK}(ID_A)$ where $A \in C_i$

(ii) *Local-leader* LC_i collects all the *ID*s and then sends them along with its own ID to the base station for authentication: $LC_i \rightarrow$ BS : $E_{NK}(\{ ID_{A_i} | A_i \in C_i, ID_{LC_i} \})$

2.a Shared partial key method

(iii) Following the authentication, the base station sends encrypted list of *id*s nominated for chain C_i: $BS \rightarrow LC_i$: $E_{NK}(LPK_i)$

(iv) Receiving the LPK_i, LC_i disseminates LPK_i to all member nodes of the chain using the communication model of the proposed logical topology.

(v) Each member node of the chain deletes the partial keys that are not mapped by LPK_i from P to find out its key list : $KL_i = \{P - \{L(id_x)| \ id_x \notin LPK_i\}\}$

2.b Private partial key method

(iii) Following the authentication, the base station sends encrypted list of *id*s nominated for chain C_i: $BS \rightarrow LC_i$: $E_{NK}(PKL_{LC_i})$

(iv) Each sensor *A* sends an encrypted id-list of chosen partial keys to its successor *B*, and its predecessor *C*

$A \rightarrow B$: $E_{NK}(PKL_A)$

$A \rightarrow C$: $E_{NK}(PKL_A)$ [except the first and the last member of a chain]

(v) Each member node *A* of the chain deletes the partial keys that are not mapped by PKL_B and PKL_C (assuming *B* and *C* are successor and predecessor of *A*) from *P* to find out its own key-list: $KL_A = \{P - \{L(id_x)|id_x \notin \{PKL_B \cap PKL_C\}\}\}$

3. Steady state phase

3.a Shared partial key

To establish an encryption/decryption key, two sensor nodes *A* and *B* (where $A, B \in C_i$) act as follows:

(i) $A \rightarrow B$: $E_{NK}(O_A) = E_{NK}(R_A(LPK_i))$ where R_A is a random function that changes the order of the elements remaining the cardinality same.

(ii) $B \rightarrow A$: $E_{NK}(O_B) = E_{NK}(R_B(LPK_i))$

(iii) Now both *A* and *B* compute the secret key for the first round as $K^t_{AB} = (L(O_A[t]) \,||\, L(O_B[t]))$ where $O_A[t]$ returns the *t*-th *id* of O_A. For the next round *A* and *B* compute the secret key as $K^{t+1}_{AB} = (L(O_A[t+1]) \,||\, L(O_B[t+1]))$ and so on.

3.b Private Partial key

A and *B* compute the secret key for the first round as $K^t_{AB} = (L(PKL_A[t]) \,||\, L(PKL_B[t]))$. For the next rounds, *A* and *B* compute the secret key as $K^{t+1}_{AB} = (L(PKL_A[t+1]) \,||\, L(PKL_B[t+1]))$, and so on.

Figure 7.3: Proposed key management protocol. (Two versions are depicted interleaved).

of the chain. The lower-level leader then disseminates the *LPK* (encrypted by the network key) to all member nodes of the chain using the communication model of the proposed logical topology. Once a sensor node comes to know which partial keys it will use with its neighbour, it deletes the rest of the partial keys and their corresponding *ID*s.

At this stage, the sensor nodes are ready to establish the communication keys. Note that, in the proposed logical topology, a sensor node only communicates with its neighbouring nodes. Furthermore, all the nodes of a chain share the same list of partial keys. To establish communication keys with its neighbour, a node A creates an order list O_A, which maintains the same cardinality with *LPK* but the order of the *ID*s is different, and sends it (encrypted by the network key) to its neighbour. A's neighbour, B, sends a similar order list O_B to A. Sensors A and B can now construct their secret communication keys for each round. In the higher-level chain, the lower-level leaders can construct their session keys in a similar way if they share some common partial keys. This can be done by the BS while assigning *LPK*s and generating some common partial keys in each *LPK* such that the local-level leaders can construct a secure communication key among themselves. When a new chain is reconstructed, the aforementioned procedures take place.

Steps 2b, and 3b (private partial keys)

In this version of the proposed protocol, instead of using shared partial keys, each member sensor node of a chain chooses its own private partial keys. Each sensor then creates the corresponding id list of the partial keys it chooses. This list is called as partial key list, *PKL*. Now, instead of sending the *LPK*s, sensors send *PKL*s. The number of partial keys to be selected is predetermined. When the lower-level leader sends the encrypted *ID* list of the chain, BS verifies all the sensors and sends a partial key list to each lower-level leader. Partial key lists for the lower-level leaders are to be selected by the BS because lower-level leaders should have some common partial keys among themselves to construct a symmetric encryption / decryption key. Thus, each sensor selects a predetermined number of partial keys from the key pool and thus constructs a partial key list *PKL* and sends it to its successor and predecessor. Each sensor can now establish a secured encryption /decryption key with its successor in the same way that a sensor establishes a secured key using shared partial keys.

Shared partial keys vs. private partial keys

One of the implications of using private partial keys is that it increases the resilience at a cost of more buffers being required in each sensor to store triple number of partial keys compared to the shared partial keys approach. This is because a sensor has to save the partial keys of its own as well as the partial keys of its successor and predecessor sensor nodes. However, in this approach, sensors do not need to store the partial key order lists of neighbouring sensors. Using private partial keys instead of using shared partial keys also increases the domain of session key candidates. This is illustrated in Figure 7.4. Thus, private partial keys approach requires lower numbers of partial keys compared to the shared partial keys approach to have the same number of session key candidates. For example, in Figure 7.4, 35 private partial keys can create as many session key candidates, which can be generated using 50 shared partial keys. To store 50 partial keys in the shared keys approach around 400 bytes are required, whereas using 35 private partial keys, around 280 bytes are required.

Another implication of the private partial key method is that, whenever an adversary captures a node, the adversary can at best find out the possible links encryption/decryption keys used by the node and its neighbours. Thus, the probability that an adversary can decrypt a random communication link by knowing another node is greatly reduced.

7.2.5 Analysis and simulation results

While designing a communication protocol and imposing security constraints on it, the first thing to consider is the number of messages to accomplish security measures in that communication protocol. In order to reduce the total number of messages sent, and thus to save energy, sensor readings from multiple nodes may be processed at one of many possible aggregation points. An aggregation point collects sensor readings from surrounding nodes and forwards a single message representing an aggregate of the values. Aggregation points are typically regular sensor nodes, and their selection is not necessarily static. Aggregation points could be chosen dynamically for each query or event, for example. It is also possible that every node in the network functions as an aggregation point, delaying transmission of an outgoing message until a sufficient number of incoming messages has been received and aggregated. Note that the proposed logical topology chooses the later one. Also note that, in the proposed key management protocol, once the partial key list is received by the sensor nodes, two neighbouring nodes do not need to send messages to establish each secret key.

Power management in sensor networks is also critical. At full power, the Berkeley Mica mote can run for only two weeks or so before exhausting its batteries. Consequently, for a sensor network to last for years, it is crucial that they run at around a 1% duty cycle (or less). Similarly, since the power consumption of the radio is at least three orders of magnitude higher when transmitting or listening than when in sleep mode, it is crucial to keep the radio in sleep mode for the overwhelming majority of time. In addition, communication bandwidth is extremely dear: each bit transmitted consumes about as much power as executing 800-1000 instructions [Hill et al., 2000], and therefore, any message expansion caused by security mechanisms comes at significant cost. That is why emphasis is given on calculating the future secret keys by collecting the order list from a sensor's neighbour, rather than sending and receiving partial keys in each round.

Besides data aggression, and the number of messages required for the key management protocol, the following evaluation metrics were also considered: (i) the number of session key candidates, (ii) scalability, (iii) key connectivity, (iv) resilience, (v) storage complexity, (vi) processing complexity, and (vii) communication complexity.

The number of session key candidates simply refers to the minimum number of session keys possibly being created for data encryption/decryption. Figure 7.4 shows the exponential increase of session key domain with respect to the number of partial key assigned to a node. This means that finding out a secret key even knowing the *LPK* or *PKL* is nearly impossible.

Scalability is the ability to support larger networks. Larger networks can be supported if there is enough storage for the required security credentials which is related to storage complexity of the solution. Figure 7.4 also implicitly shows high scalability of the proposed protocol. Every sensor needs only 100 partial keys to store (which requires only 850 bytes) one of the 20,000 session keys that can be generated for a large sensor network.

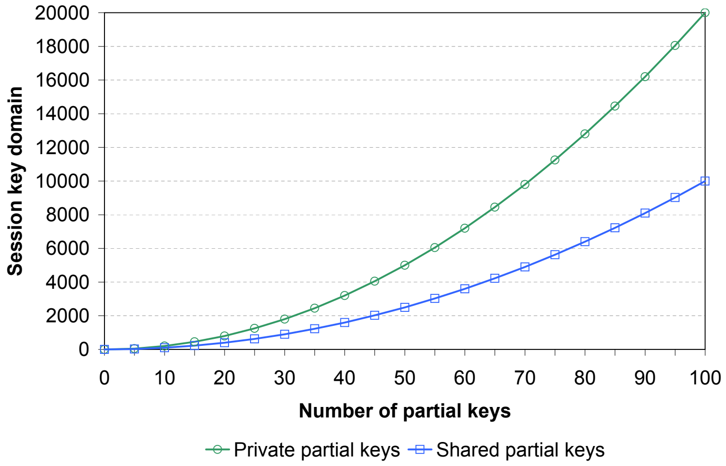


Figure 7.4: Number of session key candidates.

Resilience can be defined in one of the following ways: (i) probability that a link is compromised when an adversary captures a node, (ii) number of nodes whose security credentials are compromised when an adversary captures a node, or (iii) number of sensor nodes required to be captured to compromise a whole WSN. Note that using the proposed protocol, even if an adversary captures a node, it cannot find out the encryption/decryption key, because the keys that remain in a sensor node are partial keys rather than full secret keys. Assume that the adversary can mount a physical attack on a sensor node after it is deployed, and read secret information from the sensor's memory. In this situation, the resilience of a scheme can be evaluated by estimating the fraction of total network communications that are compromised by a capture of x nodes not including the communications in which the compromised nodes are directly involved [Ren et al., 2008].

This section measures the resilience of the proposed protocol by calculating the fraction of links in the network that an attacker is able to eavesdrop as a result of recovering keys from captured nodes. That is, an answer is sought for the question: for any two nodes A and B in the network, where neither A nor B has been captured by the attacker, what is the probability that the attacker can decrypt their communication keys using the subset of the key pool that was recovered from the nodes that were compromised? Using m partial keys, the number of full keys that can be established is ${}^{m}\mathrm{P}_2$ (where P is the permutation function). Let the number of captured nodes be x.

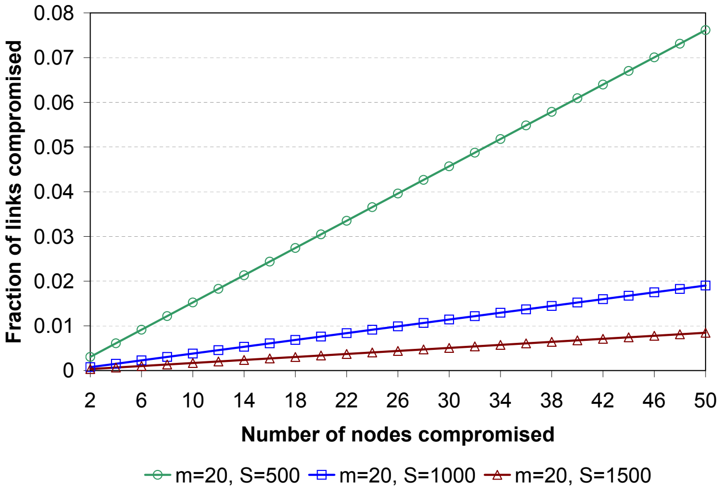


Figure 7.5: Resilience of the proposed key management protocol.

Since each node contains m partial keys, the probability that a given key has not been compromised is $p(not\_compromised) = (1 - x \times {}^{m}\mathrm{P}_2 / {}^{S}\mathrm{P}_2)$. Thus, the expected fraction of total keys compromised is ${}^{m}\mathrm{P}_2/{}^{S}\mathrm{P}_2$. Hence, the probability that any secure link setup in the key-setup phase between two uncompromised nodes is compromised when x number of nodes have been captured is ${}^{m}\mathrm{P}_2/{}^{S}\mathrm{P}_2$. Figure 7.5 shows how it varies with the number of nodes captured by the attacker. Note that, the scale of the x-axis shows absolute numbers of nodes compromised (i.e., independent of the actual total size of the network) while the y-axis is the fraction of the total network communications compromised.

Key connectivity, one of the important factors for random key pre-distribution schemes, considers the probability that two (or more) sensor nodes store the same key or keying material to be able to establish pair-wise, group-wise or network-wise keys [Camtepe and Yener, 2007]. In the proposed protocol, when all sensor nodes use private partial key lists, once a sensor node receives the order list from its neighbours, it is always able to construct the secret communication keys. On the other hand, when a public partial key list is used for a chain, all members of the chain share the same LPK, and therefore, share the same partial keys. Thus, connectivity is always guaranteed in this proposed key management protocol.

Efficiency of the solutions is generally measured using their storage, processing, and communication complexities. Here storage complexity is measured using the

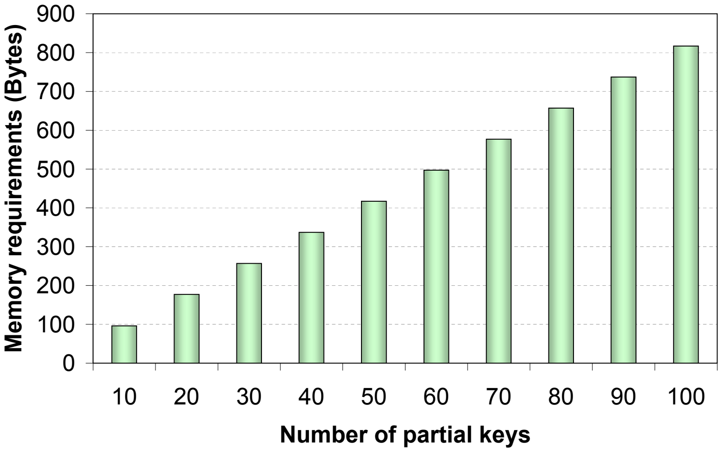


Figure 7.6: Memory requirement of the proposed key management protocol.

amount of memory units required to store security credentials. The proposed key management protocol requires very low storages to keep the partial keys (around 420 bytes to store 50 partial keys, their identifiers, and the network key). Figure 7.6 depicts the memory requirement of the proposed protocol. Communication complexity is measured as the number and size of packets sent and received by a sensor node. In the proposed protocol, a node communicates with its neighbour using only the identifiers rather than partial keys. Thus, the packet size is relatively low. Moreover, to create the communication keys, each sensor employs fundamental calculations, such as concatenation.

In summary, the proposed key management protocol minimizes the constraints of the WSNs, while maintaining very high level of security aspects. The underlying logical topology and its communication abstract make this possible for the key management protocol. The next section describes another application protocol, and further discusses how the underlying logical topology helps the application protocol in different ways.

7.3 Protocol 2: Data Transfer Protocol

In some applications of WSNs, a source node needs to transfer certain instructions or control information to a destination node. Based on these instructions, or control information, a specific action is taken. For example, a WSN, which is placed on a

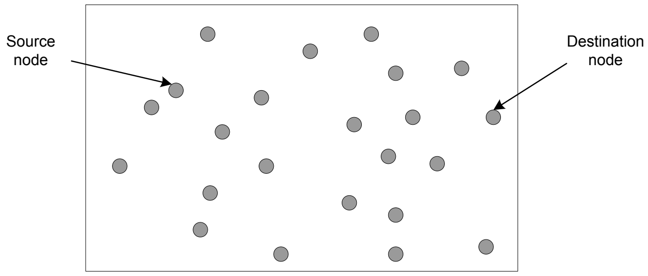


Figure 7.7: Source node needs to send data to the destination node.

riverbed to measure water level of the river, can send control signals to a distant sensor node to access the gates of a dam. Thus, data transfer in WSNs is crucial at times, and this section addresses this issue. In doing so, this section presents a data transfer protocol, the problem of which is depicted in Figure 7.7. In this figure, the source node needs to transfer a piece of data to the destination node. This problem slightly differs from the routing problem. In WSNs, routing problems usually refer to the problem of sending data from a sensor node to the BS or data sink.

The proposed data transfer protocol uses the proposed multi-chain oriented logical topology. The aim of this protocol is to design the routing paths for data to send from a source sensor node to another sensor node in the network. The following section discusses the existing works on data transfer protocols, and identifies the requirements of a data transfer protocol.

7.3.1 Existing protocols for data transfer

Several existing protocols can be used to transfer data from a source node to a destination node, such as flooding, gossiping and SPIN. Comparative discussions of them are provided below.

A. ***Flooding.*** In flooding, a node wishing to send a piece of data to another node across the network starts by sending a copy of this data to all of its neighbours. Whenever a node receives new data, it makes copies of the data and sends the data to all of its neighbours, except the node from which it just received the data. In this fashion, the destination node receives the intended data from the source node. One of the main drawbacks of flooding includes implosion, which is caused by duplicate messages sent to the same node [Heinzelman et al., 1999]. Another major problem is overlap, which occurs when two nodes sensing the same region send similar packets to the same neighbour. Flooding is also responsible for resource blindness by consuming a large amount of energy without considering other energy constraints [Chang and Liu, 2007].

B. ***Gossiping.*** Gossiping [Krishnamachari et al., 2003] is an alternative to the flooding approach, which uses randomization to conserve energy. Gossiping avoids the problem of implosion by just selecting a random node to send the packet, instead of broadcasting the packet blindly. However, this causes delays in propagation of data through the nodes [Boyd et al., 2006]. Since the source sends the packet to only one of its neighbours, and because the neighbour sends the packet to only one of its neighbours, the fastest rate at which gossiping distributes data is one node/round. However, gossiping does not solve the overlap problem.

C. ***SPIN.*** Another well-known protocol that can be used to transfer data, is called Sensor Protocols for Information via Negotiation (SPIN). SPIN is a family of adaptive protocols proposed by Kulik et al. [2002]. The original SPIN protocols disseminate all the information at each node to every node in the network, assuming that all nodes in the network are potential BSs. These protocols make use of the property that nodes in close proximity have similar data, and hence there is a need only to distribute the data that other nodes do not possess. The SPIN family of protocols uses data negotiation and resource adaptive algorithms. Nodes running SPIN assign a high-level name to completely describe their collected data (called meta-data), and perform meta-data negotiations before any data is transmitted. This assures that there is no redundant data sent through the network. The semantics of the meta-data format is application specific, and is not specified in SPIN. For example, sensors might use their unique *ID*s to report meta-data if they cover a certain known region.

To transfer data from a source node to a destination node, SPIN can be used in the following way. SPIN uses three types of messages: *ADV*, *REQ* and *DATA*. The message type *ADV* is used to advertise data, *REQ* to request data, and *DATA* is the actual message itself. The protocol starts with the source node by broadcasting an

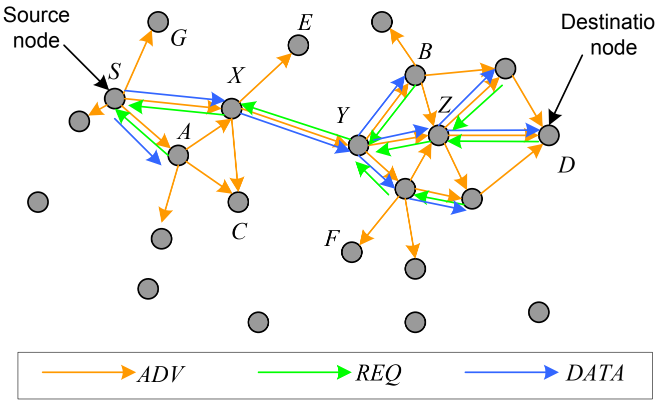


Figure 7.8: Illustration of SPIN protocol.

ADV message containing meta-data. If a neighbour is interested (e.g., if the node is in the direction of the destination node), it sends a *REQ* message for the data, and then *DATA* is sent to this neighbour node. The receiving sensor node further repeats the same process with its neighbouring sensor nodes. As a result, the data propagates towards the destination node, and after some time, the destination node receives the data. Figure 7.8 illustrates the SPIN protocol for sending data from the node S to the node D. This figure shows that the intermediate nodes X, Y, and Z co-operate with the source node S to send data to the destination node D. Note that, if a node receives some data, and at later receives an advertisement for the same data, the receiving node does not respond to the advertisement. For example, the node X receives data from the node S, and after that receives an *ADV* message from the node A for the same data. That is why the node X does not send any *REQ* message to the node A. In addition, there are a number of nodes which are not interested at all in the data. For example, although the nodes C, E, F, and G receive *ADV* messages, they do not respond to the advertisement, because these nodes are not interested in the data. A node might not be interested in receiving data when it receives an *ADV* message for many reasons, such as, the node does not have enough energy to receive and forward the data, or the data is not intended for the node, or the node is not in the direction of the destination node etc.

Although SPIN differs itself from the classic flooding by implementing its mechanisms to control the unwanted flooding, number of messages generated to transfer the data from the source to the destination is still high. In addition, and most importantly, SPIN protocol does not guarantee the delivery of data, because intermediate nodes between the source and the destination nodes may not be interested in advertised data. Therefore, such data may not be forwarded to the destination [Rehena et al., 2010].

From the above discussions, several potential requirements for a data transfer protocol can be identified as follows: i) a lower number of messages required to transfer data, ii) lower total energy to be spent, iii) prevention of implosion, iv) prevention of overlap, v) shorter time required to send data from the source node to the destination node, and v) higher reliability. The next section proposes a data transfer protocol, which uses the proposed multi-chain oriented logical topology, and discusses each of the above mentioned requirements.

7.3.2 Proposed data transfer protocol

The proposed data transfer protocol is based on the hierarchical structure and communication abstraction of the proposed multi-chain oriented logical topology. Assume that all the nodes in the target field are organized as the multi-chain oriented logical topology. Without the loss of generality, it can be assumed that the lower-level leaders know their corresponding chain members. Further, assume that each sensor has a unique *ID*, and that a lower-level leader knows all the *ID*s of its chain members. The proposed data transfer protocol works in the following way:

i) Data transfer is initiated by the source node. The source node sends the data to its neighbouring node in the direction of its lower-level leader.

ii) When the lower-level leader receives the data, it sends a probe message to its next lower-level leader in the higher-level chain. The probe message contains the address of the destination node.

iii) Whenever a lower-level leader receives a probe message along the higher-level chain, the chain leader checks the destination node address in the probe message. If a lower-level leader finds that the destination address contained in the probe message matches with one of its chain member's address, the lower-level leader sends back a probe reply message towards the sender of the probe message. On

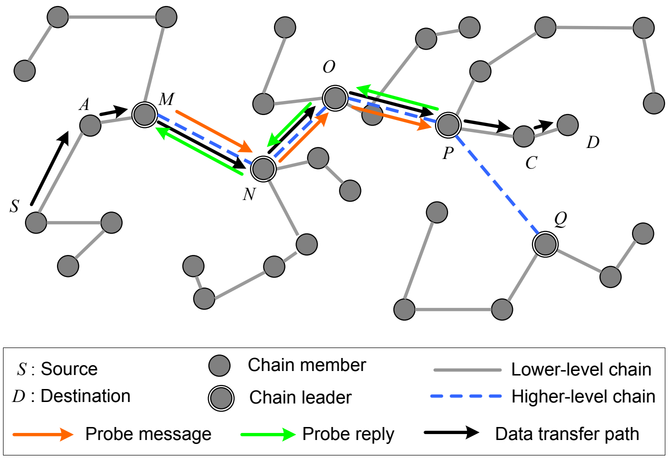


Figure 7.9: Illustration of the proposed data transfer protocol.

the other hand, if the destination address matches with none of its chain member's address, the lower-level leader simply forwards the probe message to the next lower-level leader in the higher-level chain.

iv) After receiving the probe reply message, the lower-level leader of the source node encapsulates the data, and sends it to the lower-level leader, which sent the probe reply message, via the higher-level chain. The data packet contains two addresses, one for the lower-level leader, and another for the destination node.

v) When the lower-level leader of the destination node receives the encapsulated data packet, it removes the header, and sends the data towards the destination node via the lower-level chain.

vi) Finally, the destination node receives the data.

The proposed data transfer protocol is illustrated in Figur 7.9. In this figure, the source node S wants to send data to the destination node D. The node M is the lower-level leader of S's chain, while the node P is the lower-level leader of D's chain. The other local leaders of the deployed sensor nodes are N, O, and P. All the local leaders construct a single higher-level chain $MNOPQ$.

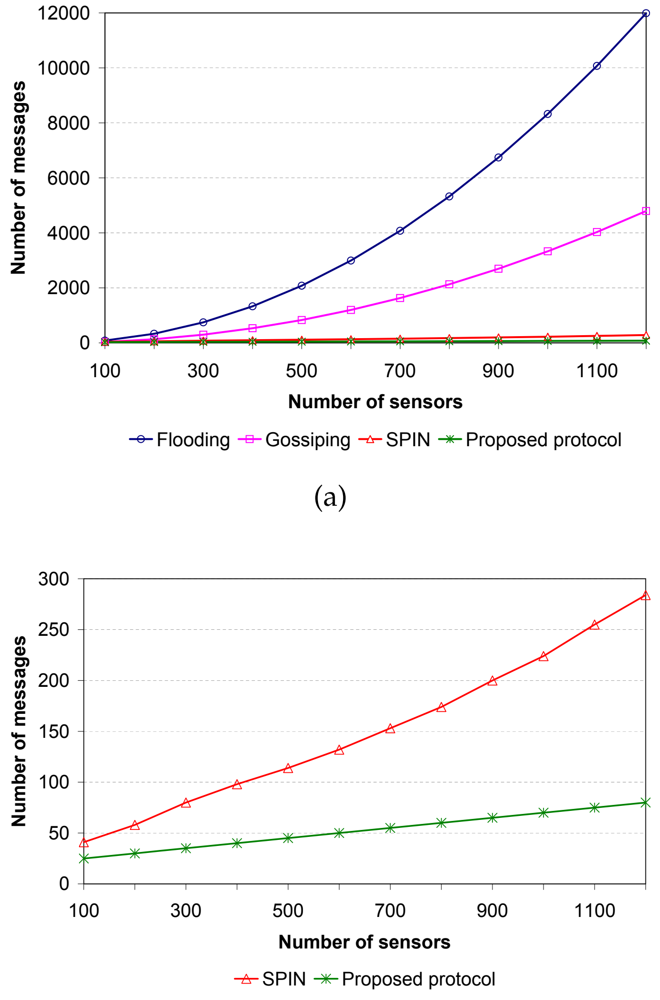
First, the source node S sends the data packet to lower-level leader node M via the node A. The data packet contains the *ID* of the destination node D. The node M needs to send the data to the lower-level leader of the destination node. To find the lower-level leader of the destination node D, the node M then sends a *probe message*, which contains the *ID* of the destination node D through the higher-level chain $MNOPQ$. Both of the nodes N and P forward the *probe message*, because they find the address inside the *probe message* does not match with any of the members of their chains. On the other hand, when the lower-level leader P receives the *probe message*, it finds the address inside the *probe message* matches one of the *ID*s of its member nodes. As a result, it sends a *probe reply message* to the node M via the higher-level chain $MNOPQ$. After receiving the probe reply message from the lower-level leader P, the lower-level leader M puts the address of the node P as a header and sends the packet to the node P via the higher-level chain $MNOPQ$. When node P receives data from M, it removes the header inserted by the node M, and sends it to the destination node D through the node C of the lower-level chain.

7.3.3 Analysis and simulation results

The proposed data transfer protocol is free from implosion and overlap. This is because the proposed protocol uses the communication abstraction of the proposed logical topology, where a node only communicates with its successive nodes in the chain. The proposed data transfer protocol creates a single virtual path from the source node to the destination node. Thus, there is no likelihood of having implosion or overlap using the proposed data transfer protocol.

With the existence of the proposed logical topology, it is always guaranteed that the source node is able to send the data to the destination node. If any intermediate node dies, it is the responsibility of the logical topology to mend this problem. The data transfer protocol does not need to worry about this. On the other hand, neither SPIN nor gossiping can assure that the source node is always able to send the data to the destination node, because the intermediate nodes may not be interested in forwarding the data.

This section further evaluates the performance of the proposed data transfer protocol with respect to the other protocol requirements described in section 7.3.1. In doing so, several simulation experiments were performed. First of all, the proposed protocol required the lowest number of total messages among all the data transfer pro-



(a)

(b)

Figure 7.10: Comparison of the proposed data transfer protocol with flooding, gossiping, and SPIN in respect of number of messages required to transfer data.

tocols discussed in this section. Obviously, flooding requires the highest number of messages, because flooding is a blind process. Gossiping protocol, although creating fewer messages than flooding, required a very high number of messages compared to SPIN or the proposed protocol.

Figure 7.10(a) shows the comparison regarding the number of messages required to tranfer data among flooding, gossiping, SPIN, and the proposed protocol. This figure shows that flooding and gossiping require vast number of messages compared to SPIN and the proposed protocol. The performance comparison between SPIN and

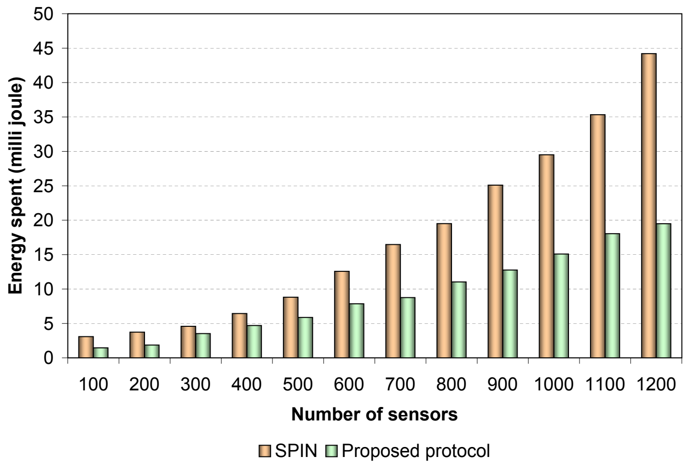


Figure 7.11: Comparison of the proposed data transfer protocol with SPIN with respect to the total energy spent on transferring data from the source node to the destination node.

the proposed protocl is not comprehensible from Figure 7.10(a). That is why Figure 7.10(b) shows the comparison between SPIN and the proposed protocol using different scale. The proposed data transfer protocol required the lowest number of messages to transfer data from source to destination because of the communication abstraction of the logical topology.

From the above discussion, it can be said that the proposed data transfer protocol requires the lowest energy to transfer data from source to destination, because energy consumption is directly related to the number of messages generated in the process of data transfer. Fig 7.11 shows the simulation results regarding the comparison of energy spent by SPIN and the proposed protocol. In the simulation experiment, the source node and the destination nodes were chosen randomly from opposite side of the target field, and the number of sensor nodes was gradually increased. In this experiment, the total energy spent was calculated using the transmission cost, while it was assumed that the data processing cost is negligible. It was found that the total energy spent was not perfectly proportional to the number of messages. This is because the distance between two successive nodes in a higher-level chain is greater than that of a lower-level chain.

Figure 7.12 shows the comparison of the proposed data transfer protocol with flooding, gossiping and the SPIN protocol in respect of time required for the data

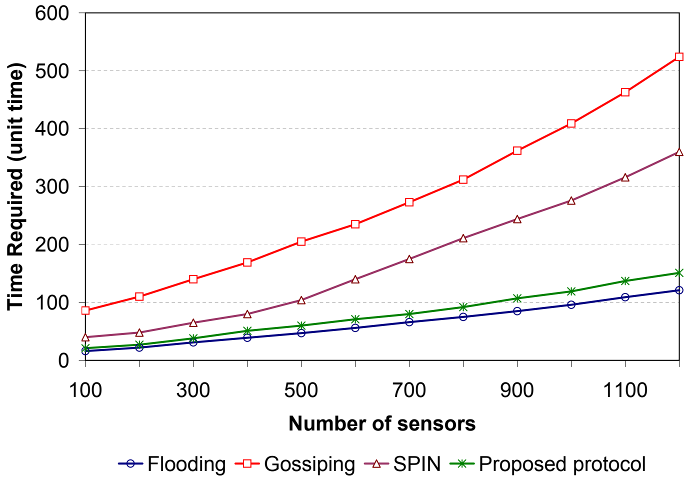


Figure 7.12: Comparison of the proposed data transfer protocol with flooding, gossiping, and SPIN with respect to the time required to transfer data from the source node to the destination node.

transfer. Gossiping took the longest time to transfer data among all the protocols, because the fastest rate at which gossiping distributed data was 1 node/round. Although the SPIN protocol distributed data using more than one node at a time, it spent a significant amount of time in advertising the data to the intermediate nodes. On the other hand, flooding showed the best result with this experiment. To send the data from the source node to the destination node, the proposed protocol took a slightly longer time compared to that of flooding. This was because of sending the *probe message* in the higher-level chain. However, sending a probe message is worthwhile because there is no point in sending a large data packet along the higher-level chain if the destination node is unavailable for some reason.

7.4 Summary

This chapter discusses various data related protocols, which are designed and applied on the top of the proposed logical topology. Each of the protocols demonstrate better performance compared to other existing protocols. For example, in a large WSN of 500 nodes, the proposed data transfer protocol saves around 35% of energy required to transfer a data packet from one end to the other end of a target field of $200m \times 200m$ of dimensions. Using the same establishment, the proposed data transfer protocol requires 42% and 71% less time compared to that of the SPIN and gossiping protocols respectively. Moreover, using the logical structure and the communication abstraction of the proposed topology, the proposed data transfer protocol guarantees no implosion and no overlap. The proposed data dissemination protocol distributes secret keys to establish a secured WSN. The proposed protocol demonstrates high resilience, high key connectivity, lower storage complexity, and lower processing and communication complexities. This is possible because of the underlying logical topology, and its communication abstraction. Using the proposed logical topology, data collection protocols saves more energy, and helps to lengthen the lifetime of the network.

Behind the scene of high performances of these protocols, the underlying logical topology contributed a lot. All the protocols discussed in this chapter use the hierarchical structure and the communication abstraction of the logical topology, and thus performed better. As communication among the sensor nodes is the most important attribute in WSNs, it can be argued that other application protocols, too, will provide better results using the proposed logical topology.

Chapter 8

Conclusion

Recent advances in wireless and MEMS[1]-based sensor technologies, low-power analog and digital electronics, and low-power RF[2] design have enabled the development of relatively inexpensive wireless sensor technology. It is known that, new technologies replace existing technologies or fill new niches when there are economic advantages. Thus, wireless sensors will replace wired technologies, because no wiring incurs lower costs and more flexibility in deployments. In a word, wireless sensor networks have a bright future; many applications have been proposed, and availability of sensors will lead to a large number of new and exciting applications. Hence, a design paradigm is needed for these application protocols of WSNs. This thesis contributes in this area by using logical topology.

8.1 Contributions of the Thesis

This thesis presents a novel notion in application protocols design paradigm of WSNs. The traditional approaches of designing application protocols tend to focus primarily on developing protocols first, and then using them on different topologies for implementation. This thesis, however, argues that logical topology of WSNs should be considered before designing application protocols in WSNs. The argument is made on the basis that the logical topology of WSNs dictates the structure and hierarchy of the network. It governs the communications among sensor nodes providing various decisions, such as routing path establishment, leader selection, successor-predecessor determination etc. Logical topology further covers other issues, such as network

[1] Micro-Electro-Mechanical System

[2] Radio Frequency

management, connectedness, data aggregation, or data fusion. Thus a well-designed topology provides benefits to design various application protocols. For example, the application protocol can use the communication abstraction of the logical topology, or it can use some network management sub-routines to ensure quality of service (QoS). This proposed design paradigm also helps to contend with various constraints of WSNs, such as limited energy, low quality of communication, limited computational resources, and scalability. For this reason, this thesis primarily aims to design an improved logical topology for WSNs.

In doing so, different logical topologies that are used as the underlying structures of different protocols were investigated. By defining a set of evaluation metrics, these topologies were compared with one another from different perspectives. As a result of this comparative evaluation, this thesis argues that chain-oriented topology has the highest potential to minimize different constraints of WSNs in comparison with any other topologies.

Following the results of the comparative analysis of different topologies, a basic multi-chain oriented logical topology was designed. The multi-chain topology demonstrated excellent results in saving energy consumption, lengthening lifetime, and reducing latency. An embedded network management architecture was developed to ensure different network management aspects, such as fault detection, performance management, and security management. Furthermore, to enhance the performance of the proposed topology, three adaptations were proposed, namely node scheduling scheme, localized chain creation scheme, and a mobile data collection scheme.

The first adaptation was designed for coverage based sensor node scheduling. The node scheduling scheme was motivated by the reason that some applications of wireless sensor networks do not require 100% coverage. In addition, in a target field, sensor nodes are usually deployed densely, and this creates redundancy. By exploiting both redundancy of sensor nodes and the requirement of less than 100% of coverage, the scheduling scheme selects a set of sensor nodes to meet the user's requirement of coverage ratio.

The second adaptation was intended to create localized chains for the proposed logical topology. Localized chains mean chains that are restricted in local areas. *Voronoi* *diagram* was used in this scheme to divide a target area into several smaller areas (*Voronoi cell*s). *Voronoi* tessellation technique was chosen, because one of the characteristics of *Voronoi diagram*s is that dense subsets of sites (sensor nodes) give rise to *Voronoi cell*s of small area, and that sparse subsets of sites produce larger *Voronoi cell*s. Thus, a *Voronoi diagram* balances the site density. In each of the created *Voronoi cell*s, a single chain was constructed. The constructed chains exhibited the following characteristics: i) the chains were optimal in chain lengths, iii) all the created chains were of similar length, and iv) there was no long link between any two successive nodes in a chain. These characteristics of chains assured low interference and low energy consumptions.

With the third and last adaptation, a mobile data collection scheme was designed for the logical topology. The proposed scheme employed multiple mobile data collectors, called MDCs, and used the spatial division multiple access (SDMA) technique. MDCs were used to traverse a region of the sensor network, and then to collect data from nearby chain leaders, and to transmit the data to the BS. As a result, lower-level chains did not need to send data to distant nodes/BS. This resulted in saving a large amount of energy. Furthermore, the SDMA technique was used to minimize the time required to gather data from the leader nodes.

After developing the multi-chain oriented logical topology and its adaptations, a number of application protocols were designed on top of the proposed topology. As WSNs are very much data centric, a number of data related protocols, namely data collection protocol, data dissemination protocol, and data transfer protocol were discussed. All the protocols used the hierarchical structure, and the communication abstract of the logical topology. The performances of these protocols demonstrated how a well-designed logical topology influences the designing of other protocols.

In summary, the main contributions of the thesis are:

i) This thesis demonstrates that the issues of WSN constraints should be addressed first by examining the logical topology.

ii) The proposed multi-chain oriented logical topology outperforms other chain-oriented topologies in respect of energy consumptions, network lifetime, and latency. For example, the simulation results showed that the proposed topology saved 10% - 20% more energy than PEGASIS, and 15% - 20% more energy than COSEN. Additionally, the proposed topology spent energy more evenly. Using the proposed protocol, the first node died after 540 rounds, whereas for PEGASIS it was 350. Furthermore, using the proposed topology all sensor nodes died after 648 rounds, whereas PEGASIS could manage up to 575 rounds. Accordingly, the first sensor node of the proposed topology expired around 190 operational rounds later than that of PEGASIS. Thus, if the network lifetime is defined as the time when the first sensor node dies, the proposed topology extended the system lifetime by around 55%. On the other hand, if the system lifetime is defined by the time when all sensor nodes die, the proposed topology offered around 20% extended lifetime. The definitive improvement of the proposed topology over PEGASIS was the latency in executing operational rounds. In the simulations, it was found that the proposed topology required 80% less amount of time for executing 100 operational rounds than PEGASIS.

iii) The coverage-based node scheduling algorithm proposed as the first adaptation saves a significant amount of energy by sacrificing a small amount of coverage area. Using a mathematical model it was shown that, the proposed node scheduling algorithm required the minimal number of nodes to provide the desired coverage ratio. The simulation results also showed that while the proposed scheduling algorithm was applied with the proposed multi-chain oriented topology, the system lifetime was doubled, while sacrificing only 8% coverage ratio. The scheduling protocol was also compared with other similar existing protocols (PECAS and PEAS). It was found that, the proposed algorithm lost only 30% sensor node, while both PECAS and PEAS lost 100% sensor nodes. Moreover, another great advantage of the proposed algorithm was that, the node-scheduling scheme was embedded into the basic logical topology seamlessly without any modification of its original workflow.

iv) The *Voronoi* tessellation technique used for the second adaptation created chains with shortened lengths. Furthermore, the chains produced lower interference, and consumed lower energy. The proposed scheme was compared with other chain construction algorithms, and it was found that the total chain length produced by the proposed scheme was 30% smaller than that of PEGASIS, and 12%

smaller than that of COSEN. The proposed scheme also constructed chains of similar sizes, and this aspect of the chains was useful for even energy distribution. For example, the standard deviation of chain-lengths of different chains created in the proposed scheme was 1.82, whereas it was 4.16, 5.40 and 12.53 for ECR, EBCRP, and CHIRON respectively. Thus, the proposed scheme were able to lengthen the lifetime of the network by 55%, compared to the existing protocols, namely ECR and EBCRP. While the other protocols were facing exponentially increasing interference in large-scale sensor networks, the increment of the interference in the proposed scheme remained very steady. It proved the scalability of the scheme.

v) The mobile data collection scheme, designed as the third adaptation of the proposed logical topology, saved around 17.5% of the total energy in 200 rounds of operation. As a result of the implementation of the mobile data collectors, sensor nodes did not require to send data to distant leader nodes/BS. Therefore, more even energy distribution was possible. As a result, the occurrence of first node's death was delayed by 80% of the operational rounds.

vi) The application protocols, which were designed based on the hierarchical structure and communication abstraction of the proposed logical topology, showed excellent results. The secret key management protocol required less storage, processing, and communication complexities, while providing high resilience and robustness. The data transfer protocol also required a very low number of messages compared to other existing protocols. For example, in a network of 1,000 nodes, to send data from a source node to a destination node, the protocol SPIN required 224 messages, while the proposed protocol used only 65 messages in the worst case. It was found in the simulations that the proposed data transfer protocol saved more than 100% energy compared to the SPIN protocol in a large network of 1,200 nodes. The outperforming performances of the application protocols were possible because of their usage of the proposed multi-chain oriented logical topology. As in a WSN communication among the sensor nodes is the most important attribute, it can be argued that other application protocols, too, would achieve better results using the proposed logical topology.

8.2 Future Research Directions

In this thesis, a novel application protocol design paradigm was presented. This design paradigm can be used to design new application protocols with minimum cost and effort. The use of multi-chain oriented logical topology in designing various application protocols of WSNs can enhance the performance of the protocols. There remain some issues such as testing the presented techniques in real situations that require large-scale WSN deployment. Moreover, further improvements can be incorporated to improve and extend the presented protocols. A list of some possible future directions is provided below:

- *New topology design*: Maintaining the principle of this thesis 'logical topology first, then protocol', other types of logical topologies can be designed for areas which this thesis did not cover, such as i) Body Sensor Networks, ii) Vehicular Sensor Networks, iii) Machine-to-Machine, iv) Acoustic (underwater) Sensor network, and v) Interplanetary Sensor Networks etc.
- *Compliance with real-time constraints*: In real-time applications, data is delay constrained, and has a certain bandwidth requirement. For instance, scheduling messages with deadlines is an important issue in order to take appropriate actions in real time. However, due to the interference and contention on the wireless medium, this is a challenging task. The communication abstraction of the multi-channel oriented topology can help to reduce the delay by increasing the number of parallel transmissions and help the network to achieve real-time guarantees.
- *Multiple applications running on the same network*: With the latest operating systems for WSNs, it is possible to have multiple applications running on the same network. This certainly leads to larger amounts of data to be transmitted in the network and handling the traffic, often with different priority levels, in an energy efficient way while avoiding collisions and interference becomes a major issue. Multi-channel communication can be a topic to be researched along with the proposed multi-chain logical topology for solving the problems that arise with running multiple applications in the network.

- *Different application protocol design*: In this thesis, only three application protocols were discussed. Various other application protocols can be designed using the proposed logical topology, and its communication abstraction. According to Pinto et al. [2006], maximum efficiency can be reached when the communication specification is entered at high levels of abstraction, and the design process optimizes the implementation from this specification. As the proposed topology removes the burden of communication, replacing flooding or multicast by unicast, any protocol that is designed carefully with the proposed logical topology would result in high performances.

References

Abo-El-Fotoh, H., E. ElMallah, and H. Hassanein (2006, June). On the reliability of wireless sensor networks. In *IEEE International Conference on Communications, (ICC 2006)*, Volume 8, pp. 3455–3460.

Akyildiz, I., W. Su, Y. Sankarasubramaniam, and E. Cayirci (2002, August). A survey on sensor networks. *IEEE Communications Magazine 40*(8), 102–114.

Akyildiz, I. F., T. Melodia, and K. R. Chowdury (2007, December). Wireless multimedia sensor networks: a survey. *IEEE Wireless Communications 14*(6), 32–39.

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

Aurenhammer, F. (1991). Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys 23*(3), 345–405.

Bachir, A., D. Barthel, M. Heusse, and A. Duda (2006, January). A synthetic function for energy-delay mapping in energy efficient routing. In *Third Annual Conference on Wireless On-demand Network Systems and Services, (WONS 2006)*, pp. 170–178.

Baeg, S. H., J. H. Park, J. Koh, K. W. Park, and M. H. Baeg (2007, August). RoboMaid-Home: A sensor network-based smart home environment for service robots. In *The 16th IEEE International Symposium on Robot and Human Interactive Communication, (RO-MAN 2007)*, pp. 182–187.

Banimelhem, O. and S. Khasawneh (2009, May). Grid-based multi-path with congestion avoidance routing (GMCAR) protocol for wireless sensor networks. In *International Conference on Telecommunications, (ICT 2009)*, pp. 131–136.

Banka, T., G. Tandon, and A. Jayasumana (2005, April). Zonal rumor routing for wireless sensor networks. In *International Conference on Information Technology: Coding and Computing, (ITCC 2005)*, Volume 2, pp. 562–567.

Behroozi, H., F. Alajaji, and T. Linder (2008, July). On the optimal power-distortion region for asymmetric gaussian sensor networks with fading. In *IEEE International Symposium on Information Theory, (ISIT 2008)*, pp. 1538–1542.

Belding Royer, E. M. (2003, September). Multi-level hierarchies for scalable ad hoc routing. *Wireless Networks 9*(5), 461–478.

Beluch, T., D. Dragomirescu, F. Perget, and R. Plana (2010, April). Cross-layered synchronization protocol for wireless sensor networks. In *2010 Ninth International Conference on Networks, (ICN '10)*, pp. 167–172.

Benliang, L., H. Wang, B. Yan, and C. Zhang (2006, June). The research of applying wireless sensor networks to intelligent transportation system (ITS) based on IEEE802.15.4. In *6th International Conference on ITS Telecommunications Proceedings, 2006*, pp. 939–942.

Blough, D. M. and P. Santi (2002, September). Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks. In *8th ACM International Conference on Mobile Computing and Networking, (MobiCom 2002)*, pp. 183–192.

Boyd, S., A. Ghosh, B. Prabhakar, and D. Shah (2006, June). Randomized gossip algorithms. *IEEE Transactions on Information Theory 52*(6), 2508–2530.

Buratti, C., A. Conti, D. Dardari, and R. Verdone (2009, August). An overview on wireless sensor networks technology and evolution. *Sensors 2009 9*(9), 6869–6896.

Burkhart, M., P. von Rickenbach, R. Wattenhofer, and A. Zollinger (2004). Does topology control reduce interference? In *The 5th ACM International Symposium on Mobile Ad-hoc Networking and Computing, (MobiHoc 2004)*, pp. 9–19.

Cai, W., X. Jin, Y. Zhang, K. Chen, and J. Tang (2006, September). Research on reliability model of large-scale wireless sensor networks. In *International Conference on Wireless Communications, Networking and Mobile Computing, (WiCOM 2006)*, pp. 1–4.

Camtepe, S. and B. Yener (2007, April). Combinatorial design of key distribution mechanisms for wireless sensor networks. *IEEE/ACM Transactions on Networking 15*(2), 346–358.

Capkun, S., M. Hamdi, and J. P. Hubaux (2001, January). GPS-free positioning in mobile ad-hoc networks. In *The 34th Annual Hawaii International Conference on System Sciences*, pp. 10–19.

CC2420 (2004). [Online] http://focus.ti.com/docs/prod/folders/print/cc2420.htm, (cited: October 25, 2010).

Cens (2002). [Online] http://research.cens.ucla.edu/, (cited: October 25, 2010).

Cerpa, A. and D. Estrin (2004, July). ASCENT: Adaptive self-configuring sensor networks topologies. *IEEE Transactions on Mobile Computing 3*(3), 272–285.

Chan, H., A. Perrig, and D. Song (2003, May). Random key predistribution schemes for sensor networks. In *2003 Symposium on Security and Privacy*, pp. 197–213.

Chandrakasan, A., R. Min, M. Bhardwaj, S. Cho, and A. Wang (2002, September). Power aware wireless microsensor systems. In *The 28th European Solid-State Circuits Conference, (ESSCIRC 2002)*, pp. 47–54.

Chang, N. and M. Liu (2007, April). Controlled flooding search in a large network. *IEEE/ACM Transactions on Networking 15*(2), 436–449.

Chen, B. J., K. Jamieson, H. Balakrishnan, and R. Morris (2002, September). SPAN: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *International Journal of Wireless Networks 8*, 481–494.

Chen, K. H., J. M. Huang, and C. C. Hsiao (2009, April). CHIRON: An energy-efficient chain-based hierarchical routing protocol in wireless sensor networks. In *Wireless Telecommunications Symposium, (WTS 2009)*, pp. 1–5.

Chen, M. X. and Y. D. Wang (2009, August). An efficient location tracking structure for wireless sensor networks. *Computer Communication 32*(13-14), 1495–1504.

Chen, W. P., J. Hou, and L. Sha (2004, July). Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing 3*(3), 258–271.

Chen, W. Q. and B. J. Hu (2010, September). A data fusion algorithm in LEACH protocol using gauss membership function. In *The 6th International Conference on Wireless Communications Networking and Mobile Computing, (WiCOM '10)*, pp. 1–4.

Chen, X., W. Qu, H. Ma, and K. Li (2008, September). A geography based heterogeneous hierarchy routing protocol for wireless sensor networks. In *10th IEEE International Conference on High Performance Computing and Communications, (HPCC 2008)*, pp. 767–774.

Chen, Y., A. Liestman, and J. Liu (2006, May). A hierarchical energy-efficient framework for data aggregation in wireless sensor networks. *IEEE Transactions on Vehicular Technology 55*(3), 789–796.

Chen, Y. and Q. Zhao (2005, November). On the lifetime of wireless sensor networks. *IEEE Communications Letters 9*(11), 976–978.

Cheng, Z., M. Perillo, and W. Heinzelman (2008a, April). General network lifetime and cost models for evaluating sensor network deployment strategies. *IEEE Transactions on Mobile Computing 7*(4), 484–497.

Cheng, Z., M. Perillo, and W. Heinzelman (2008b, April). General network lifetime and cost models for evaluating sensor network deployment strategies. *IEEE Transactions on Mobile Computing 7*(4), 484–497.

Cho, J., G. Kim, T. Kwon, and Y. Choi (2007, September). A distributed node scheduling protocol considering sensing coverage in wireless sensor networks. In *The IEEE 66th Vehicular Technology Conference, (VTC-2007)*, pp. 352–356.

Chong, C. Y. and S. Kumar (2003, August). Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE 91*(8), 1247–1256.

Cody Kenny, B., D. Guerin, D. Ennis, R. Simon Carbajo, M. Huggard, and C. Mc Goldrick (2009). Performance evaluation of the 6LoWPAN protocol on MICAz and TelosB motes. In *The 4th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, pp. 25–30.

Cormen, T. H., C. Leiserson, R. Rivest, and C. Stein (2001). *Introduction to algorithms*. The MIT Press.

Dardari, D., A. Conti, C. Buratti, and R. Verdone (2007, July). Mathematical evaluation of environmental monitoring estimation error through energy-efficient wireless sensor networks. *IEEE Transactions on Mobile Computing 6*(7), 790–802.

Datta, D. and S. Kundu (2007, April). Reliable and efficient data transfer in wireless sensor networks via out-of-sequence forwarding and delayed request for missing packets. In *Fourth International Conference on Information Technology, (ITNG 2007)*, pp. 128–133.

Dong, Q. (2005, April). Maximizing system lifetime in wireless sensor networks. In *Fourth International Symposium on Information Processing in Sensor Networks, (IPSN 2005)*, pp. 13–19.

Du, K., J. Wu, and D. Zhou (2003, April). Chain-based protocols for data broadcasting and gathering in the sensor networks. In *International Parallel and Distributed Processing Symposium, 2003*, pp. 8–13.

Du, W., J. Deng, Y. Han, S. Chen, and P. Varshney (2004, March). A key management scheme for wireless sensor networks using deployment knowledge. In *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, (INFOCOM 2004)*, Volume 1, pp. 586–597.

Du, X. and H. H. Chen (2008, August). Security in wireless sensor networks. *IEEE Wireless Communications 15*(4), 60–66.

Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of Mathematics 17*, 449–467.

Elson, J. and D. Estrin (2001). Random, ephemeral transaction identifiers in dynamic sensor networks. Technical Report [Online] http://www.circlemud.org/ jelson/writings/retri/ (cited: October 30, 2010).

Eltoweissy, M., M. Moharrum, and R. Mukkamala (2006, April). Dynamic key management in sensor networks. *IEEE Communications Magazine 44*(4), 122–130.

Eltoweissy, M., A. Wadaa, S. Olariu, and L. Wilson (2005). Group key management scheme for large-scale sensor networks. *Ad Hoc Networks 3*(5), 668–688.

Eschenauer, L. and V. D. Gligor (2002). A key-management scheme for distributed sensor networks. In *The 9th ACM Cconference on Computer and Communications Security, (CCS 2002)*, pp. 41–47.

Estrin, D., L. Girod, G. Pottie, and M. Srivastava (2001). Instrumenting the world with wireless sensor networks. In *Acoustics, Speech, and Signal Processing, (ICASSP 2001)*, Volume 4, pp. 2033–2036.

Evans, J. (2005, October). Wireless sensor networks in electrical manufacturing. In *Proceedings of Electrical Insulation Conference and Electrical Manufacturing Expo, 2005.*, pp. 460–465.

Fan, K. W., S. Liu, and P. Sinha (2007, August). Structure-free data aggregation in sensor networks. *IEEE Transactions on Mobile Computing 6*(8), 929–942.

Fan, K. W., S. Liu, and P. Sinha (2008, October). Dynamic forwarding over tree-on-dag for scalable data aggregation in sensor networks. *IEEE Transactions on Mobile Computing 7*(10), 1271–1284.

Fariborzi, H. and M. Moghavvemi (2009, May). EAMTR: Energy aware multi-tree routing for wireless sensor networks. *IET Communications 3*(5), 733–739.

Feeney, L. and M. Nilsson (2001). Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *20th Annual Joint Conference of the IEEE Computer and Communications Societies, (INFOCOM 2001)*, Volume 3, pp. 1548–1557.

Fei, H., J. Ziobro, J. Tillet, and N. K. Sharma (2005). Secure wireless sensor networks: Problems and solutions. *Systemics, Cybernetics and Informatics 1*(4), 90–100.

Fortune, S. (1986). A sweepline algorithm for Voronoi diagrams. In *The Second Annual Symposium on Computational Geometry*, SCG '86, pp. 313–322.

Girod, L. and D. Estrin (2001). Robust range estimation using acoustic and multimodal sensing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Volume 3, pp. 1312–1320.

Gui, C. and P. Mohapatra (2004). Power conservation and quality of surveillance in target tracking sensor networks. In *ACM Mobicom 2004*, pp. 129–143.

Gupta, P. and P. R. Kumar (2000, March). The capacity of wireless sentworks. *IEEE Transactions on Information Theory 46*(2), 388–404.

Hamida, E. and G. Chelius (2008, December). Strategies for data dissemination to mobile sinks in wireless sensor networks. *IEEE Wireless Communications 15*(6), 31–37.

Haupt, S., K. Long, G. Young, and A. Beyer (2007, August). Data requirements from evolvable sensor networks for homeland security problems. In *Second NASA/ESA Conference on Adaptive Hardware and Systems, (AHS 2007)*, pp. 58–66.

He, L.-M. (2008, October). Time synchronization based on spanning tree for wireless sensor networks. In *4th International Conference on Wireless Communications, Networking and Mobile Computing, (WiCOM 2008)*, pp. 1–4.

Heinzelman, W., A. Chandrakasan, and H. Balakrishnan (2000, January). Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000*, Volume 2, pp. 10–20.

Heinzelman, W. R., J. Kulik, and H. Balakrishnan (1999). Adaptive protocols for information dissemination in wireless sensor networks. In *The 5th Aannual ACM/IEEE International Conference on Mobile Computing and Networking, (MobiCom 1999)*, pp. 174–185.

Hill, J., R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister (2000, November). System architecture directions for networked sensors. *SIGARCH Computer Architecture News 28*(5), 93–104.

Huang, D., M. Mehta, D. Medhi, and L. Harn (2004). Location-aware key management scheme for wireless sensor networks. In *The 2nd ACM Workshop on Security of Aad hoc and Sensor Networks, (SASN 2004)*, pp. 29–42.

Huang, Y., W. He, and K. Nahrstedt (2009, October). ChainFarm: A novel authentication protocol for high-rate any source probabilistic broadcast. In *IEEE 6th International Conference on Mobile Adhoc and Sensor Systems, (MASS 2009)*, pp. 264–273.

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

Intanagonwiwat, C. (2002). *Directed diffusion: an application-specific and data-centric communication paradigm for wireless sensor networks*. Ph. D. thesis, University of Southern California, Los Angeles, CA, USA.

J. Pan, Y.T. Hou, L. C. (2003, September). Topology control for wireless sensor networks. In *9th ACM Annual International Conference on Mobile Computing and Networking, (MobiCom 2003)*, pp. 286–299.

Jin, Y., F. Z. Chen, G. F. Che, and W. Hu (2010, April). Energy-efficient data collection protocol for wireless sensor network based on tree. In *2010 Asia-Pacific Conference on Wearable Computing Systems, (APWCS)*, pp. 82–85.

Johansson, T., E. Osipov, and L. Carr Motyčkovà (2008). Interference aware construction of multi- and convergecast trees in wireless sensor networks. In *The 8th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking, (NEW2AN) and 1st Russian Conference on Smart Spaces, (ruSMART)*, pp. 72–87.

Jung, W. S., K. W. Lim, Y. B. Ko, and S. J. Park (2009, February). A hybrid approach for clustering-based data aggregation in wireless sensor networks. In *Third International Conference on Digital Society, (ICDS 2009)*, pp. 112–117.

Kahn, J. M., R. H. Katz, and K. S. J. Pester (2000, September). Emerging challenges: Mobile networking for smart dust. *International Journal of Communications and Networks 2*, 188–196.

Kang, H., X. Li, and P. Moran (2007, March). Power-aware markov chain based tracking approach for wireless sensor networks. In *IEEE Wireless Communications and Networking Conference, (WCNC 2007)*, pp. 4209–4214.

Kang, I. and R. Poovendran (2005, December). Maximizing network lifetime of broadcasting over wireless stationary ad hoc networks. *Mobile Networks and Applications (MONET), Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks 10*(6), 879–896.

Kaplan, E. D. (1996). *Understanding GPS: Principles and Applications*. Artech House Publishers.

Karthickraja, N. and V. Sumathy (2010, Januay). A study of routing protocols and a hybrid routing protocol based on rapid spanning tree and cluster head routing in wireless sensor networks. In *International Conference on Wireless Communication and Sensor Computing, (ICWCSC 2010)*, pp. 1–6.

Kausar, F. and A. Masood (2006, December). A random key distribution scheme for securing wireless sensor network. In *IEEE Multitopic Conference, (INMIC 2006)*, pp. 32–36.

Khan, S., E. N. Huh, and I. Rao (2008, February). Reliable data dissemination with diversity multi-hop protocol for wireless sensors network. In *10th International Conference on Advanced Communication Technology, (ICACT 2008)*, Volume 1, pp. 9–13.

Kim, D. S., S. Y. Lee, K. H. Won, D. J. Chung, and J. H. Kim (2007, November). Time-synchronized forwarding protocol for remote control of home appliances based on wireless sensor network. *IEEE Transactions on Consumer Electronics 53*(4), 1427–1433.

Kim, H. S. and K. J. Han (2005, February). A power efficient routing protocol based on balanced tree in wireless sensor networks. In *First International Conference on Distributed Frameworks for Multimedia Applications, (DFMA 2005)*, pp. 138–143.

Kim, K., J. Jun, S. Kim, and B. Sung (2008, August). Medical asset tracking application with wireless sensor networks. In *Second International Conference on Sensor Technologies and Applications, (SENSORCOMM 2008)*, pp. 531–536.

Kim, K. T., C. H. Lyu, S. S. Moon, and H. Y. Youn (2010, April). Tree-based clustering(tbc) for energy efficient wireless sensor networks. In *The IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, (WAINA '10)*, pp. 680–685.

Krishnamachari, B., Y. Mourtada, and S. Wicker (2003, May). The energy-robustness tradeoff for routing in wireless sensor networks. In *IEEE International Conference on Communications, (ICC 2003)*, Volume 3, pp. 1833–1837.

Kulik, J., W. Heinzelman, and H. Balakrishnan (2002, March). Negotiation-based protocols for disseminating information in wireless sensor networks. *Journal of Wireless Networks 8*(2/3), 169–185.

Kumar, S., A. Arora, and T. Lai (2005, November). On the lifetime analysis of always-on wireless sensor network applications. In *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005*, pp. 183–188.

Kuo, J. C., W. Liao, and T. C. Hou (2009, October). Impact of node density on throughput and delay scaling in multi-hop wireless networks. *IEEE Transactions on Wireless Communications 8*(10), 5103–5111.

Lai, W. K., C. S. Shieh, and Y. T. Lee (2009, August). A cluster-based routing protocol for wireless sensor networks with adjustable cluster size. In *Fourth International Conference on Communications and Networking in China, (ChinaCOM 2009)*, pp. 1–5.

Lee, J. C., V. Leung, K. Wong, J. Cao, and H. Chan (2007, October). Key management issues in wireless sensor networks: current proposals and future developments. *IEEE Wireless Communications 14*(5), 76–84.

Lee, K. and M. Reichardt (2005, December). Open standards for homeland security sensor networks. *IEEE Instrumentation Measurement Magazine 8*(5), 14–21.

Lee, S. B., K. J. Kwak, and A. Campbell (2006, September). Solicitation-based forwarding for sensor networks. In *3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, (SECON 2006)*, Volume 1, pp. 90–99.

Li, H., H. Yu, and A. Liu (2006, April). A tree based data collection scheme for wireless sensor network. In *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, (ICN/ICONS/MCL 2006)*, pp. 119–125.

Li, L. and J. Halpern (2001, June). Minimum-energy mobile wireless networks revisited. In *IEEE International Conference on Communications, (ICC 2001)*, Volume 1, pp. 278–283.

Li, N., J. Hou, and L. Sha (2005, May). Design and analysis of an mst-based topology control algorithm. *IEEE Transactions on Wireless Communications 4*(3), 1195–1206.

Li, X. Y., P. J. Wan, Y. Wang, and C. W. Yi (2004, February). Fault tolerant deployment and topology control in wireless ad hoc networks: Research articles. *Wireless Communicatios and Mobile Computing 4*(1), 109–125.

Lima, C. and G. de Abreu (2008, April). Clusterization for robust geographic routing in wireless sensor networks. In *IEEE Wireless Communications and Networking Conference, (WCNC 2008)*, pp. 2385–2390.

Lin, C. Y., W. C. Peng, and Y. C. Tseng (2006, August). Efficient in-network moving object tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing 5*(8), 1044–1056.

Lindsey, S. and C. Raghavendra (2002). PEGASIS: Power-efficient gathering in sensor information systems. In *IEEE Aerospace Conference*, Volume 3, pp. 1125–1130.

Lindsey, S., C. Raghavendra, and K. Sivalingam (2001, April). Data gathering in sensor networks using the energy*delay metric. In *The 15th International Symposium on Parallel and Distributed Processing*, pp. 2001–2008.

Lindsey, S., C. Raghavendra, and K. Sivalingam (2002a, September). Data gathering algorithms in sensor networks using energy metrics. *IEEE Transactions on Parallel and Distributed Systems 13*(9), 924–935.

Lindsey, S., C. Raghavendra, and K. Sivalingam (2002b, September). Data gathering algorithms in sensor networks using energy metrics. *IEEE Transactions on Parallel and Distributed Systems 13*(9), 924–935.

Liu, C., K. Wu, and J. Pei (2005, September). A dynamic clustering and scheduling approach to energy saving in data collection from wireless sensor networks. In *Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, (IEEE SECON 2005)*, pp. 374–385.

Liu, C., K. Wu, Y. Xiao, and B. Sun (2006, June). Random coverage with guaranteed connectivity: joint scheduling for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems 17*(6), 562–575.

Liu, D. and P. Ning (2005, November). Improving key predistribution with deployment knowledge in static sensor networks. *ACM Transactions on Sensor Network 1*(2), 204–239.

Liu, D., P. Ning, and R. Li (2005, February). Establishing pairwise keys in distributed sensor networks. *ACM Transactions on Information and System Security 8*(1), 41–77.

Liu, J. and B. Li (2003, December). Distributed topology control in wireless sensor networks with asymmetric links. In *IEEE Global Telecommunications Conference, (GLOBECOM 2003)*, Volume 3, pp. 1257–1262.

Liu, R., Z. Rosberg, I. Collings, C. Wilson, A. Dong, and S. Jha (2008, September). Overcoming radio link asymmetry in wireless sensor networks. In *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, (PIMRC 2008)*, pp. 1–5.

Long, D. S. and X. Tao (2006, May). Cluster-based power efficient time synchronization in wireless sensor networks. In *IEEE International Conference on Electro/information Technology*, pp. 147–151.

Lopez, J. (2006, September). Unleashing public-key cryptography in wireless sensor networks. *Journal of Computer Security 14*(5), 469–482.

Luo, H., Y. Liu, and S. Das (2006, November). Routing correlated data with fusion cost in wireless sensor networks. *IEEE Transactions on Mobile Computing 5*(11), 1620–1632.

Luo, H., H. Tao, H. Ma, and S. K. Das (2011, March). Data fusion with desired reliability in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems 22*(3), 501–513.

Luo, J. and J. P. Hubaux (2005, March). Joint mobility and routing for lifetime elongation in wireless sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Volume 3, pp. 1735–1746.

Ma, C., M. Ma, and Y. Yang (2004, June). Data-centric energy efficient scheduling for densely deployed sensor networks. In *IEEE International Conference on Communications, (ICC 2004)*, Volume 6, pp. 3652–3656.

Ma, M., Z. Zhang, and Y. Yang (2005, November). Multi-channel polling in multi-hop clusters of hybrid sensor networks. In *IEEE Global Telecommunications Conference, (GLOBECOM 2005)*, Volume 1, pp. 6–10.

Macedo, M. (2009, April). Are there so many sons per node in a wireless sensor network data aggregation tree? *IEEE Communications Letters 13*(4), 245–247.

Mahdy, A. (2008, April). Marine wireless sensor networks: Challenges and applications. In *Seventh International Conference on Networking, (ICN 2008)*, pp. 530–535.

Malan, D., M. Welsh, and M. Smith (2004, October). A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, (IEEE SECON 2004)*, pp. 71–80.

Mamun, Q. and S. Ramakrishnan (2008, May). SecCOSEN: A key management scheme for securing chain oriented sensor networks. In *The 6th Annual Communication Networks and Services Research Conference, (CNSR 2008)*, pp. 584–592.

Mamun, Q., S. Ramakrishnan, and B. Srinivasan (2008, November). An efficient partial key pre-distribution scheme for chain oriented sensor networks. In *IEEE Region 10 Conference, (TENCON 2008)*, pp. 1 –6.

Mamun, Q., S. Ramakrishnan, and B. Srinivasan (2010a, April). Energy-preserving node scheduling scheme for chain oriented wireless sensor networks. In *2nd International Conference on Computer Engineering and Technology, (ICCET 2010)*, Volume 2, pp. 373–377.

Mamun, Q., S. Ramakrishnan, and B. Srinivasan (2010b, April). Multi-chain oriented logical topology for wireless sensor networks. In *2nd International Conference on Computer Engineering and Technology, (ICCET 2010)*, Volume 2, pp. 367–372.

Mamun, Q., S. Ramakrishnan, and B. Srinivasan (2010c, April). Selecting member nodes in a chain oriented WSN. In *IEEE Wireless Communications and Networking Conference, (WCNC 2010)*, pp. 1 –6.

Mamun, Q., S. Ramakrishnan, and B. Srinivasan (2010d, October). voronoi diagram based chain construction in chain oriented sensor networks. In *Addressing Research Challenges: Emerging Research Conference, (ARCHER 2010)*.

Mamun, Q., K. Zahid, and H. Nakazato (2006, January). Using wireless sensor networks with mobile networks for acquiring road traffic information. In *IEEE International Conference on Next-Generation Wireless Systems, (ICNEWS 2006)*.

Manjeshwar, A. and D. Agrawal (2001, April). TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. In *15th International Parallel and Distributed Processing Symposium*, pp. 2009–2015.

Manolopoulos, Y., D. Katsaros, and A. Papadimitriou (2010). Topology control algorithms for wireless sensor networks: a critical survey. In *The 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies*, (CompSysTech 2010), pp. 1–10.

Mao, S. and Y. Hou (2007, November). BeamStar: An edge-based approach to routing in wireless sensor networks. *IEEE Transactions on Mobile Computing 6*(11), 1284–1296.

Mao, Y., X. Zhou, and Y. Zhu (2008, June). An energy-aware coverage control protocol for wireless sensor networks. In *International Conference on Information and Automation, (ICIA 2008)*, pp. 200–205.

Martinez, K., J. Hart, and R. Ong (2004, August). Environmental sensor networks. *Computer 37*(8), 50–56.

Mascarenas, D., E. Flynn, C. Farrar, G. Park, and M. Todd (2009, December). A mobile host approach for wireless powering and interrogation of structural health monitoring sensor networks. *IEEE Sensors Journal 9*(12), 1719–1726.

Megerian, B., F. Koushanfar, M. Potkonjak, and M. Srivastava (2005, February). Worst and best-case coverage in sensor networks. *IEEE Transactions on Mobile Computing 4*(1), 84–9.

Messina, D., M. Ortolani, and G. Lo Re (2007, October). A network protocol to enhance robustness in tree-based WSNs using data aggregation. In *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems, (MASS 2007)*, pp. 1–4.

Miao, P., Y. Xiao, and P. Wang (2009, November). Error analysis and kernel density approach of scheduling sleeping nodes in cluster-based wireless sensor networks. *IEEE Transactions on Vehicular Technology 58*(9), 5105–5114.

Ming, X. S., W. L. Min, W. X. Sheng, and Z. Y. Zhao (2009, August). Application of wireless sensor networks to remote medical treatment system. In *IEEE International Symposium on IT in Medicine Education, (ITIME 2009)*, Volume 1, pp. 96–99.

Moraes, R., C. Ribeiro, and C. Duhamel (2009, December). Optimal solutions for fault-tolerant topology control in wireless ad hoc networks. *IEEE Transactions on Wireless Communications 8*(12), 5970–5981.

Muruganathan, S., D. Ma, R. Bhasin, and A. Fapojuwo (2005, March). A centralized energy-efficient routing protocol for wireless sensor networks. *IEEE Communications Magazine 43*(3), S8–S13.

Nest (2001). [Online] http://nest.cs.berkeley.edu/nest-index.html, (cited: October 25, 2010).

Nordio, A., C. F. Chiasserini, and E. Viterbo (2008, March). Signal reconstruction in multidimensional sensor fields. In *IEEE International Zurich Seminar on Communications*, pp. 56–59.

Olule, E., G. Wang, M. Guo, and M. Dong (2007, September). RARE: An energy-efficient target tracking protocol for wireless sensor networks. In *International Conference on Parallel Processing Workshops, (ICPPW 2007)*, pp. 76–81.

Pandana, C. (2005). *Resource and Environment Aware Sensor Communications: Framework, Optimization, and Applications*. Ph. D. thesis, University of Maryland, College Park.

Park, Y. and E. S. Jung (2007). Plus-Tree: A routing protocol for wireless sensor networks. In *The 1st International conference on Advances in hybrid information technology*, pp. 638–646.

Paschalidis, I. C., W. Lai, and D. Starobinski (2007, February). Asymptotically optimal transmission policies for large-scale low-power wireless sensor networks. *IEEE/ACM Transactions on Networking, 15*(1), 105–118.

Peng, L., F. Xiao, and Z. Ni (2009, August). Design for wireless sensor network-based intelligent public transportation system. In *6th International Conference on Anti-counterfeiting, Security, and Identification in Communication, (ASID 2009)*, pp. 351–354.

Peng, T. G., J. Zhang, and N. Z. Bian (2007, September). Grid-based routing algorithm for sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pp. 2242–2245.

Perrig, A., R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler (2002, September). SPINS: Security protocols for sensor networks. *Wireless Networks 8*(5), 521–534.

Pham, M. L., D. Kim, Y. Doh, and S. E. Yoo (2004, December). Power aware chain routing protocol for data gathering in sensor networks. In *Intelligent Sensors, Sensor Networks and Information Processing Conference*, pp. 107–112.

Pinto, A., A. Bonivento, and A. L. Sangiovanni Vincentelliacm (2006, July). System level design paradigms: Platform-based design and communication synthesis. *Transactions on Design Automation of Electronic Systems 11*(3), 537–563.

Pister, K. (1998). Smart dust. [Online] http://robotics.eecs.berkeley.edu/ pister/SmartDust/, (cited: October 25, 2010).

Poojary, S. and M. Pai (2010, November). Multipath data transfer in wireless multimedia sensor network. In *International Conference on Broadband, Wireless Computing, Communication and Applications, (BWCCA 2010)*, pp. 379–383.

Quan, Z., A. Subramanian, and A. Sayed (2007, October). REACA: An efficient protocol architecture for large scale sensor networks (corrected)*. *IEEE Transactions on Wireless Communications 6*(10), 3846–3855.

Quek, T. Q., D. Dardari, and M. Z. Win (2007, February). Energy efficiency of dense wireless sensor networks: to cooperate or not to cooperate. *IEEE Journal on Selected Areas in Communications 25*(2), 459–470.

Raicu, I., L. Schwiebert, S. Fowler, and S. K. S. Gupta (2005). Local load balancing for globally efficient routing in wireless sensor networks. *International Journal of Distributed Sensor Networks 1*(1), 163–185.

Rappaport, T. S. (2002). *Wireless Communications: Principles and Practice*. Prentice Hall.

Rausand, M. and A. Hoyland (2004). *System Reliability Theory: Models and Statistical Methods*. Wiley Series in Probability and Mathematical Statistics. John Wiley and Sons.

Rehena, Z., K. Kumar, S. Roy, and N. Mukherjee (2010, July). SPIN implementation in tinyos environment using nesc. In *International Conference on Computing Communication and Networking Technologies, (ICCCNT 2010)*, pp. 1–6.

Ren, K., K. Zeng, and W. Lou (2005, October). On efficient key pre-distribution in large scale wireless sensor networks. In *IEEE Military Communications Conference, (MILCOM 2005)*, Volume 1, pp. 20–26.

Ren, K., K. Zeng, and W. Lou (2008, January). Secure and fault-tolerant event boundary detection in wireless sensor networks. *IEEE Transactions on Wireless Communications 7*(1), 354–363.

Rodoplu, V. and T. Meng (1999, August). Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications 17*(8), 1333–1344.

Rong, B. X., Z. Shi, X. D. Yu, and Q. Z. Ttao (2010, April). An energy-balanced chain-cluster routing protocol for wireless sensor networks. In *Second International Conference on Networks Security Wireless Communications and Trusted Computing, (NSWCTC)*, Volume 2, pp. 79–84.

Rossi, M., N. Bui, G. Zanca, L. Stabellini, R. Crepaldi, and M. Zorzi (2010, December). Synapse++: Code dissemination in wireless sensor networks using fountain codes. *IEEE Transactions on Mobile Computing 9*(12), 1749–1765.

Rothery, S., W. Hu, and P. Corke (2008). An empirical study of data collection protocols for wireless sensor networks. In *Proceedings of the workshop on Real-world wireless sensor networks, (REALWSN 2008)*, pp. 16–20.

Satapathy, S. and N. Sarma (2006). TREEPSI: Tree based energy efficient protocol for sensor information. In *IFIP International Conference on Wireless and Optical Communications Networks*, pp. 4–14.

Sausen, P., M. Spohny, and A. Perkusichy (2008, November). Energy efficient blind flooding in wireless sensors networks. In *34th IEEE Annual Conference on Industrial Electronics, (IECON 2008)*, pp. 1736–1741.

Schurgers, C., G. Kulkarni, and M. Srivastava (2002, October). Distributed on-demand address assignment in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems 13*(10), 1056–1065.

SenSAR (2010). Sensor application research. [Online] http://www.cse.unsw.edu.au/˜sensar/hardware/hardware_survey.html, (cited: October 26, 2010).

Senses (2005). [Online] http://senses.cs.ucdavis.edu/resources.html, (cited: October 26, 2010).

Shah, R., S. Roy, S. Jain, and W. Brunette (2003, May). Data MULEs: Modeling a three-tier architecture for sparse sensor networks. In *The First IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 30–41.

Shakshuki, E., S. Hussain, A. R. Matin, and A. W. Matin (2006, October). P2P multi-agent data transfer and aggregation in wireless sensor networks. In *IEEE International Conference on Mobile Adhoc and Sensor Systems, (MASS 2006)*, pp. 645–649.

Sharifzadeh, M. and C. Shahabi (2004). Supporting spatial aggregation in sensor network databases. In *The 12th Annual ACM International Workshop on Geographic Information Systems, (GIS 2004)*, pp. 166–175.

Shastry, N., J. Bhatia, and R. Adve (2005, December). Theoretical analysis of cooperative diversity in wireless sensor networks. In *IEEE Global Telecommunications Conference, (GLOBECOM 2005)*, Volume 6, pp. 3269–3273.

Shen, L., H. Wang, X. Duan, and X. Li (2008, October). Application of wireless sensor networks in the prediction of wind power generation. In *4th International Conference on Wireless Communications, Networking and Mobile Computing, (WiCOM 2008)*, pp. 1–4.

Shi, Y., Y. Hou, and A. Efrat (2006, August). Algorithm design for base station placement problems in sensor networks. In *3rd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks*.

Shih, E., S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A.Chandrakasan (2001, July). Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In *7th ACM Annual International Conference on Mobile Computing and Networking, (MobiCom 2001)*, pp. 272–287.

Shin, J. and C. Suh (2008, February). Energy-efficient chain topology in ubiquitous sensor network. In *The 10th International Conference on Advanced Communication Technology, (ICACT 2008)*, Volume 3, pp. 1688–1693.

Shokrzadeh, H., A. Rahmani, A. Haghighat, and N. Forouzideh (2010, June). SDRR: Serial directional rumor routing in wireless sensor networks. In *International Conference on Networking and Information Technology, (ICNIT 2010)*, pp. 75–79.

Shu, H., Q. Liang, and J. Gao (2008, April). Wireless sensor network lifetime analysis using interval type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems 16*(2), 416–427.

Simic, S. N. and S. Sastry (2003). Distributed environmental monitoring using random sensor networks. In *2nd International Workshop on Information Processing in Sensor Networks*, pp. 582–592.

Sivrikaya, F. and B. Yener (2004, July). Time synchronization in sensor networks: a survey. *IEEE Network 18*(4), 45–50.

Smith, C. U. and L. G. Williams (2003). *Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software* (Revised 2003 ed.). Addison-Wesley Professional.

Somasundara, A., A. Kansal, D. Jea, D. Estrin, and M. Srivastava (2006, August). Controllably mobile infrastructure for low energy embedded networks. *IEEE Transactions on Mobile Computing 5*(8), 958–973.

Stallings, W. (1999). *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2* (3rd ed.). Addison-Wesley Professional.

Su, W. and T. Bougiouklis (2008, October). Modeling of data fusion algorithms in cluster-based wireless sensor networks. In *42nd Asilomar Conference on Signals, Systems and Computers, 2008*, pp. 868–872.

Suh, C. and Y. B. Ko (2008, August). Design and implementation of intelligent home control systems based on active sensor networks. *IEEE Transactions on Consumer Electronics 54*(3), 1177–1184.

Sun, H. M., Y. H. Lin, C. T. Yang, and M. E. Wu (2009). A pair-wise key establishment for wireless sensor networks. In *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, (IIH-MSP 2009)*, pp. 1152–1155.

Tabassum, N., Q. Mamun, and Y. Urano (2006, April). COSEN: A chain oriented sensor network for efficient data collection. In *Third International Conference on Information Technology: New Generations, (ITNG 2006)*, pp. 262–267.

Tang, X. and J. Xu (2008, June). Adaptive data collection strategies for lifetime-constrained wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems 19*(6), 721–734.

TI (1999). Texas Instruments, [Online] http://www.ti.com/, (cited: October 25, 2010).

Tian, D. and N. D. Georganas (2002). A coverage-preserving node scheduling scheme for large wireless sensor networks. In *The 1st ACM International Wworkshop on Wireless Sensor Networks and Applications*, pp. 32–41.

Tian, D. and N. D. Georganas (2004, January). Location and calculation-free node scheduling schemes in large wireless sensor networks. *Ad Hoc Network 2*(1), 65–85.

Tian, Y., Y. Wang, and S. F. Zhang (2007, September). A novel chain-cluster based routing protocol for wireless sensor networks. In *International Conference on Wireless Communications, Networking and Mobile Computing, (WiCom 2007)*, pp. 2456–2459.

TIK (2008). The Sensor Network Museum, [Online] http://www.snm.ethz.ch/Main/HomePage, (cited: October 26, 2010).

Tilak, S., N. B. Abu Ghazaleh, and W. Heinzelman (2002, April). A taxonomy of wireless micro-sensor network models. *SIGMOBILE Mob. Comput. Commun. Rev. 6*(2), 28–36.

Toriumi, S., Y. Sei, and H. Shinichi (2008, November). Energy-efficient event detection in 3D wireless sensor networks. In *1st IFIP Wireless Days, (WD 2008)*, pp. 1–5.

Tsai, H. W., C. P. Chu, and T. S. Chen (2007, June). Mobile object tracking in wireless sensor networks. *Computer Communication Review 30*(8), 1811–1825.

Tsai, Y. R. (2007, April). Coverage-preserving routing protocols for randomly distributed wireless sensor networks. *IEEE Transactions on Wireless Communications 6*(4), 1240–1245.

Tse, D. and P. Viswanath (2009, February). Fundamentals of wireless communication. *IEEE Transactions on Information Theory 55*(2), 919–920.

Vass, D., V. Zoltn, V. Rolland, and V. Attila (2005, July). Energy efficiency in wireless sensor networks using mobile base station. In *11th Open European Summer School and IFIP WG6.6, WG6.4, WG6.9 Workshop, (EUNICE 2005)*, pp. 173–186.

Verdone, R., D. Dardari, G. Mazzini, and A. Conti (2008). *Wireless Sensor and Actuator Networks*. Academic Press/Elsevier.

Verdone, R., F. Fabbri, and C. Buratti (2010, September). Maximizing area throughput in clustered wireless sensor networks. *IEEE Journal on Selected Areas in Communications 28*(7), 1200–1210.

Von Rickenbach, P., S. Schmid, R. Wattenhofer, and A. Zollinger (2005, April). A robust interference model for wireless ad-hoc networks. In *19th IEEE International Symposium on Parallel and Distributed Processing, (IPDPS 2005)*, pp. 239–246.

Walsh, M., S. Alavi, and M. Hayes (2008, December). On the effect of communication constraints on robust performance for a practical 802.15.4 wireless sensor network benchmark problem. In *47th IEEE Conference on Decision and Control, (CDC 2008)*, pp. 447–452.

Wang, D., B. Xie, and D. Agrawal (2008, December). Coverage and lifetime optimization of wireless sensor networks with gaussian distribution. *IEEE Transactions on Mobile Computing 7*(12), 1444–1458.

Wang, K., S. Ayyash, T. Little, and P. Basu (2005, September). Attribute-based clustering for information dissemination in wireless sensor networks. In *Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, (IEEE SECON 2005)*, pp. 498–509.

Wang, L. and S. Kulkarni (2006). Sacrificing a little coverage can substantially increase network lifetime. In *Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, (IEEE SECON 2006)*, pp. 326–335.

Wang, L. and Z. B. Wang (2007, December). A survey of time synchronization of wireless sensor networks. In *IET Conference on Wireless, Mobile and Sensor Networks, (CCWMSN07)*, pp. 241–244.

Wang, L. and Y. Xiao (2005, October). Energy saving mechanisms in sensor networks. In *2nd International Conference on Broadband Networks, (BroadNets 2005)*, Volume 1, pp. 724–732.

Wang, L. and Y. Xiao (2006). A survey of energy-efficient scheduling mechanisms in sensor networks. *Mobile Networks and Application 11*(5), 723–740.

Wang, Q. F., S. Zhang, Y. Yang, and L. Tang (2009, December). The application of wireless sensor networks in coal mine. In *7th International Conference on Information, Communications and Signal Processing, (ICICS 2009)*, pp. 1–4.

Wang, Y., S. Goddard, and L. Perez (2007, May). A study on the cricket location-support system communication protocols. In *IEEE International Conference on Electro/Information Technology*, pp. 257–262.

Warneke, B. and K. Pister (2002). MEMS for distributed wireless sensor networks. In *9th International Conference on Electronics, Circuits and Systems, 2002*, Volume 1, pp. 291–294.

Weng, H. C., H. Chan, E. K. Wu, and G. H. Chen (2007, November). A 2-approximation double-tree algorithm for correlated data gathering in wireless sensor networks. In *IEEE Global Telecommunications Conference, (GLOBECOM 2007)*, pp. 898–902.

West, D. B. (2000). *Introduction to Graph Theory* (2nd ed.). Prentice Hall.

Woo, A., T. Tong, and D. Culler (2003). Taming the underlying challenges of reliable multihop routing in sensor networks. In *The 1st International Conference on Embedded Networked Sensor Systems*, pp. 14–27.

WSWG (2002, November). Wireless Sensor Working Group. Wireless communication protocols and transducer electronic data sheets (TEDS) formats. [Online] http://grouper.ieee.org/groups/1451/5/, (cited: October 27, 2010).

Wu, F. J. and Y. C. Tseng (2009, November). Distributed wake-up scheduling for data collection in tree-based wireless sensor networks. *IEEE Communications Letters 13*(11), 850–852.

Wu, H., Y. M. Ding, and Z. Zhong (2009, December). A chain-based fast data aggregation algorithm based on suppositional cells for wireless sensor networks. In *International Conference on Power Electronics and Intelligent Transportation System, (PEITS 2009)*, Volume 1, pp. 106–109.

Wu, K., Y. Gao, F. Li, and Y. Xiao (2005, December). Lightweight deployment-aware scheduling for wireless sensor networks. *Mobile Networks and Application 10*(6), 837–852.

Wu, Y., L. Zhang, Y. Wu, and Z. Niu (2009, February). Motion-indicated interest dissemination with directional antennas for wireless sensor networks with mobile sinks. *IEEE Transactions on Vehicular Technology 58*(2), 977–989.

Xiao, Y., H. Chen, K. Wu, B. Sun, Y. Zhang, X. Sun, and C. Liu (2010, April). Coverage and detection of a randomized scheduling algorithm in wireless sensor networks. *IEEE Transactions on Computers 59*(4), 507–521.

Xiao, Y., H. Li, Y. Pan, K. Wu, and J. Li (2004, November). On optimizing energy consumption for mobile handsets. *IEEE Transactions on Vehicular Technology 53*(6), 1927–1941.

Xibei, J., Z. Huazhong, and Z. Jingchen (2010, July). Research of data aggregation routing protocol in WSN data-related applications. In *The 3rd IEEE International Conference on Computer Science and Information Technology, (ICCSIT 2010)*, Volume 1, pp. 647–651.

Xin-lian, Z. and G. Bo (2008, December). Intra-cluster nodes scheduling algorithm satisfying expected coverage degree of application in distributed clustering WSNs. In *2008 International Conference on Computer Science and Software Engineering*, Volume 3, pp. 332–3335.

Xing, G., M. Li, H. Luo, and X. Jia (2009, September). Dynamic multiresolution data dissemination in wireless sensor networks. *IEEE Transactions on Mobile Computing 8*(9), 1205–1220.

Xing, G., X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill (2005). Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Sensor Networks 1*(1), 36–72.

Xu, X., X. Y. Li, X. Mao, S. Tang, and S. Wang (2011, January). A delay-efficient algorithm for data aggregation in multihop wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems 22*(1), 163–175.

Xu, X., S. Sahni, and N. Rao (2008, July). Minimum-cost sensor coverage of planar regions. In *11th International Conference on Information Fusion*, pp. 1–8.

Xu, Y., J. Heidemann, and D. Estrin (2000, October). Adaptive energy-conserving routing for multihop ad hoc networks. Research Report 527, USC/Information Sciences Institute, [Online] http://www.isi.edu/˜johnh/PAPERS/Xu00a.pdf, (cited: November 27, 2010).

Xu, Y., J. Heidemann, and D. Estrin (2001, July). Geography-informed energy conservation for ad hoc routing. In *7th ACM Annual International Conference on Mobile Computing and Networking, (MobiCom 2001)*, pp. 70–84.

Xue, W. and Z. Chi (2007). An immune algorithm based node scheduling scheme of minimum power consumption and no collision for wireless sensor networks. In *IFIP International Conference on Network and Parallel Computing Workshops, (NPC 2007)*, pp. 630–635.

Yang, J., D. Zhang, and Y. Zhang (2009, June). An energy-efficient data gathering protocol for wireless sensor networks. In *Eighth IEEE/ACIS International Conference on Computer and Information Science, (ICIS 2009)*, pp. 780–785.

Ye, D., D. Gong, and W. Wang (2009, December). Application of wireless sensor networks in environmental monitoring. In *The 2nd International Conference on Power Electronics and Intelligent Transportation System, (PEITS)*, Volume 1, pp. 205–208.

Ye, F., G. Zhong, S. Lu, and L. Zhang (2003, May). PEAS: A robust energy conserving algorithm for long-lived sensor networks. In *IEEE 23rd International Conference on Distributed Computing Systems, (ICDCS 2003)*, pp. 28–37.

Ye, M., C. Li, G. Chen, and J. Wu (2005, April). EECS: An energy efficient clustering scheme in wireless sensor networks. In *24th IEEE International Conference on Performance, Computing, and Communications Conference, (IPCCC 2005)*, pp. 535–540.

Yebari, M., T. Addali, A. Z. Sadouq, and M. Essaaidi (2008). Energy conservation challenges in wireless sensor networks: A state-of-the-art study. *International Journal on Information and Communication Technologies 1*(2), 29–35.

Yoo, J.-Y. and J. Kim (2007, March). Maximum end-to-end throughput of chain-topology wireless multi-hop networks. In *IEEE Wireless Communications and Networking Conference, (WCNC 2007)*, pp. 4279–4283.

Younis, M., K. Ghumman, and M. Eltoweissy (2006, August). Location-aware combinatorial key management scheme for clustered sensor networks. *IEEE Transactions on Parallel and Distributed Systems 17*(8), 865–882.

Younis, O. and S. Fahmy (2004, March). Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach. In *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, (INFOCOM 2004)*, Volume 1, pp. 629–640.

Yu, Y. and G. Wei (2007, September). Energy aware routing algorithm based on layered chain in wireless sensor network. In *International Conference on Wireless Communications, Networking and Mobile Computing, (WiCom 2007)*, pp. 2701–2704.

Yuan, L., Y. Zhu, and T. Xu (2008, October). A multi-layered energy-efficient and delay-reduced chain-based data gathering protocol for wireless sensor network. In *IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications, (MESA 2008)*, pp. 13–18.

Zhang, H. and J. Hou (2005, January). Maintaining sensing coverage and connectivity in large sensor networks. *An International Journal on Wireless Ad Hoc and Sensor Networks 1*(2), 89–124.

Zhang, Y., W. Liu, W. Lou, and Y. Fang (2005, March). Securing sensor networks with location-based keys. In *IEEE Wireless Communications and Networking Conference, (WCNC 2005)*, Volume 4, pp. 1909–1914.

Zhang, Y. and L. Wang (2008, June). A comparative performance analysis of data dissemination protocols in wireless sensor networks. In *7th World Congress on Intelligent Control and Automation, (WCICA 2008)*, pp. 6669–6674.

Zhang, Y., Z. Zhou, and M. Huang (2009, July). A comprehensive study of data gathering system in wireless sensor networks. In *International Conference on Communications, Circuits and Systems, (ICCCAS 2009)*, pp. 191–195.

Zhang, Z., M. Ma, and Y. Yang (2008, February). Energy efficient multi-hop polling in clusters of two-layered heterogeneous sensor networks. *IEEE Tansactions on Computer 57*(2), 231–245.

Zhang, Z. and F. Yu (2010, April). Performance analysis of cluster-based and tree-based routing protocols for wireless sensor networks. In *International Conference on Communications and Mobile Computing, (CMC 2010)*, Volume 1, pp. 418–422.

Zhao, M., M. Ma, and Y. Yang (2008, April). Mobile data gathering with space-division multiple access in wireless sensor networks. In *The 27th IEEE Conference on Computer Communications, (INFOCOM 2008)*, pp. 1283–1291.

Zhao, Q. and M. Gurusamy (2005, February). Lifetime maximization for connected target coverage in wireless sensor networks. *IEEE/ACM Trnasactions of Netwoking 16*(6), 84–92.

Zhao, S. and D. Raychaudhuri (2009, October). Scalability and performance evaluation of hierarchical hybrid wireless networks. *IEEE/ACM Transactions on Networking 17*(5), 1536–1549.

Zhao, W., M. Ammar, and E. Zegura (2004a, October). The energy-limited capacity of wireless networks. In *IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, (IEEE SECON 2004)*, pp. 279–288.

Zhao, W., M. Ammar, and E. Zegura (2004b). A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *The 5th ACM international symposium on Mobile ad hoc networking and computing, (MobiHoc 2004)*, pp. 187–198.

Zhaohua, L., L. Jianfeng, and J. Guiquan (2010, July). Survey of routing protocols in wireless sensor networks based on applications. In *2nd International Conference on Industrial and Information Systems, (IIS 2010)*, Volume 2, pp. 381–385.

Zhigang, H. and C. C. Hui (2009, July). The application of ZigBee based wireless sensor network and gis in the air pollution monitoring. In *International Conference on Environmental Science and Information Application Technology, (ESIAT 2009)*, Volume 2, pp. 546–549.

Zhou, Z., S. Das, and H. Gupta (2004, October). Variable radii connected sensor cover in sensor networks. In *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, (IEEE SECON 2004)*, pp. 387–396.

Zurawski, R. (2009, May). Keynote: Wireless sensor network in industrial automation. In *International Conference on Embedded Software and Systems, (ICESS 2009)*, pp. xxii–xxii.

Publications on PhD Research

i) Q. Mamun, S. Ramakrishnan and B. Srinivasan, "An efficient localized chain construction scheme for chain oriented wireless sensor networks," Accepted in *IEEE International Symposium on Autonomous Decentralized System (ISADS 2011)*, Tokyo, Japan, March, 2011.

ii) Q. Mamun, S. Ramakrishnan and B. Srinivasan, "Voronoi diagram based chain construction in chain oriented sensor networks," In *Addressing Research Challenges: Emerging Research Conference (ARCHER 2010)*, Melbourne, Australia, October, 2010.

iii) Q. Mamun, S. Ramakrishnan and B. Srinivasan, "Energy preserving node scheduling scheme for chain oriented wireless sensor networks," In *IEEE International Conference on Computer Engineering and Technology (ICCET 2010)*, Chengdu, China, pp. 373-377, April, 2010.

iv) Q. Mamun, S. Ramakrishnan and B. Srinivasan, "Multi-chain oriented logical topology for wireless sensor networks," In *IEEE International Conference on Computer Engineering and Technology (ICCET 2010)*, Chengdu, China, pp. 367-372, April, 2010.

v) Q. Mamun, S. Ramakrishnan and B. Srinivasan, "Selecting member nodes in a chain oriented WSN," In *IEEE Wireless Communications and networking Conference (WCNC 2010)*, Sydney, Australia, pp. 1-6, July, 2010.

vi) Q. Mamun, S. Ramakrishnan and B. Srinivasan, "An efficient partial key predistribution scheme for chain oriented sensor networks," In *IEEE Tencon (TENCON 2008)*, India, pp. 1-6, November, 2008.

vii) Q. Mamun and S. Ramakrishnan, "SecCOSEN - A key management scheme for securing chain oriented sensor networks," In *Communication Networks and Services Research Conference (CNSR 2008)*, Halifax, Canada, pp. 584-594, May, 2008.

Index