uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Sonia Ali Hashish
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Computer Science)
GRADE / DEGREE

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

Efficient Communication Protocols for Sensor Network Architecture with Multiple Mobile Sinks

TITRE DE LA THÈSE / TITLE OF THESIS

Ahmed Karmouch
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

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

Michel Barbeau

Nancy Samaan

Amiya Nayak

Abdelhakim Hafid (University of Montreal)

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

Efficient Communication Protocols for Sensor Network Architecture with Multiple Mobile Sinks

Thesis Submitted to
School of Information Technology and Engineering (SITE)
Faculty of Engineering
University of Ottawa

In Partial Fulfillment of the Requirement for the Degree of Doctor of Philosophy

By

Sonia Ali Hashish

Ottawa 2010

Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-66258-8
Our file Notre référence
ISBN: 978-0-494-66258-8

NOTICE:

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

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

AVIS:

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

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

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

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

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

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

Canada

To my wonderful parents, Ali and Sawsan.

Abstract

Maximizing sensor network lifetime, efficiently gathering data from sensor nodes, and leveraging the reusability of sensor network protocols are crucial to make the emerging sensor network technology succeed in real world deployments. In this thesis, our aim is to develop novel communication protocols that will achieve this goal.

First, a deployment-based solution of multiple mobile sinks is developed to maximize the lifetime of sensor networks. We introduced the concept of *peeling phenomenon and show how sinks could move* following the direction of the progressive peeling to retain the sink-to-network connectivity. We also proposed a novel data forwarding strategy, namely the "away-from-centroid" data forwarding, which increases the load balancing and leverages the utilization of the network resources. We then introduced the concept of a "guard region" that protects the core area of interest from being peeled-off and we show how this could contribute to further maximizing the network lifetime.

In the second part of this thesis, we turn our attention to leveraging our solution to support a wide class of applications with different data delivery requirements. We start by analyzing the limitation of the existing rendezvous data dissemination solutions. We then developed a novel adaptive rendezvous mechanism that efficiently adapts to the network irregularity as well as to the application query-requirements. We show how existing sinks cooperate as a single tier to build a reliable rendezvous structure. We emphasize synchronization of the underlying pull and push components to prevent resource wasting. The developed protocol works with equal efficiency in both regular as well as irregular sensor networks.

Finally, we propose a novel mobility-based protocol for structuring a generic infrastructure that could be efficiently leveraged by different upper layer protocols. The proposed protocol allows mobile robots/sinks to dynamically organize the arbitrary network topology into a *physical* co-centric circular layered infrastructure. The resulting infrastructure supports both multi-hop and data-mules regimes of communication and requires only local updates for maintenance. We prove the correctness and efficiency of the proposed scheme. We also provide a rough cost model that predicts the cost of communication over the resulting infrastructure.

Acknowledgment

I would like to express my gratitude to my supervisor, Professor A. Karmouch, for all the help and support he gave me during my five years of study. He gave me the chance to overcome all the difficulties that I have experienced. I appreciate so much his patience and his advices, which helped me to adjust my attitude towards research. I believe that I will never forget the last five years of study under his supervision, including all my positive experiences and even my negative ones.

I would also like to thank the members of my committee, Professor M. Barbeau and Professor A. Nayak, for their valuable comments, which helped me to improve the quality of my thesis in both the technical and presentation parts.

I would also like to thank all my laboratory colleagues, especially Mr. I. AlOqily, who were all very supportive during the times when I was absent from the laboratory. In addition, I thank all the anonymous reviewers who participated in reviewing any of my published work. Their reviews were a great resource to improve the quality of my work.

Finally, I would like to express my sincere thanks to my parents, who are there for me all the time. My work would not be finished without their support and love. Remembering that my success in obtaining my Ph.D. would draw happiness on their lovely faces, motivated me to do my best to finish my studies. Moreover, I cannot forget to thank my little wonderful children Amgad and Ramez. In spite of the difficulties of giving birth to them during the period of my study and being physically live in a city far away from the University, they provided me with the courage to continue my studies and to finish my job. Looking into their eyes, I believed that I would be able to do it. I would also like to thank my husband Mohamed for being with me when I experienced hard times, going from up to down and visa versa. And as a final word, I cannot forget to thank my sincere friends, who shared with me the moments of joy and sadness during all these years away from my home country and especially my best friend forever, my sister Sally.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Wireless sensor networks are a subset of wireless ad hoc networks where a large quantity of small inexpensive devices (sensor nodes) is deployed in the field of interest to create a densely distributed network. Each sensor node incorporates communication, computation, and sensing capabilities. The role that these networks could play in our lives puts them amongst the top of the research topics that attract many research communities. Sensor networks applications are widespread. They include environmental and habitat monitoring, health care, traffic control, emergency scenarios, control and structure monitoring.

Sensor nodes have constraints in their power, communication, and processing capabilities, as well as in their storage capacity. These constraints have drawn the attention of researchers for many years. Many solutions that accommodate such limitations in most of the networking aspects (e.g. data gathering, routing, and aggregations) have been developed. Yet the existence of multiple tradeoffs and the wide spectrum of system-related application requirements call for more optimization in the design of the communication protocols.

Recent advances in wireless communications and mechanical evolutions are enabling mobility and multiple radio capabilities. These powerful facilities have made possible the new trend towards heterogeneous sensor networks, and are encouraging protocol designers to take advantage of their power. Heterogeneity could include the equipped power supply, the means of communication, the sensing capabilities, and the mobility of nodes. While the new features provide strength to the old-fashioned homogenous sensor networks, they also complicate the task of designing the communication protocols. They raise new problems that require a new paradigm for communications and present new challenges for the protocol designers.

Among these features, this thesis aims to translate the cons of incorporating sink multiplicity and mobility to the pros by exploiting them to provide efficient communications protocols that enhance the network performance while avoiding the problems that may arise. First, a deployment-based solution of multiple mobile sinks was developed to maximize the lifetime of sensor networks. We aimed to exploit sink mobility to maintain the network connectivity as much as the application requires. A novel data dissemination strategy is proposed as part of our suggested solution. Then we naturally leverage our solution to support a wide class of network applications. Namely, a rendezvous mechanism based on our solution for event-driven data delivery is developed. The proposed mechanism efficiently adapts to network irregularity as well as to application query-requirements. Finally, we present an attempt to exploit sink mobility in decoupling the communication infrastructure from the upper data communication model. We propose a mobility-based protocol for structuring a generic infrastructure that could be efficiently leveraged by different upper layer applications, such as routing, broadcasting, and multiple-resolution data collection protocols.

The rest of this chapter is organized as follows: Section 1.2 describes the problem of maximizing the lifetime of sensor networks. This section provides the motivation and contributes towards solving the problem. Section 1.3 describes the problem statement and motivation and our contributions towards developing an adaptive rendezvous data delivery mechanism; Section 1.4 describes the problem of building generic infrastructure for sensor networks. Our motivation and contributions towards achieving this goal are described in this section. Section 1.5 summarizes the underlying research objectives; Section 1.6 describes the methodology of evaluation; and Section 1.7 provides the outline of the thesis and lists our publications.

1.2 Maximizing Sensor Network Lifetime

Maximizing the lifetime of sensor networks would convert the theoretical achievements in sensor network research into successful real world deployed networks. In this section, we describe the problem statement, our motivation, and our contribution towards maximizing the lifetime of sensor networks.

1.2.1 Problem Statement and Motivation

The most important reason the lifetime of sensor networks is threatened is the energy depletion of nodes located around the data collector nodes (usually referred to as sinks) [1]. In early work, sensor networks were assumed to be homogenous. All sensor devices in the network have the same communication, computation, and sensing capabilities. A typical scenario is the sensor network that consists of fixed sensors and fixed sink(s). The main task of the network is to gather the data of interest and disseminate it to the end user through the sink for further processing. Data travels to the sink using the multi-hop communication model, whereby nodes located near the collectors are exposed to a heavier load. Such nodes lose their energy resources faster than others and an early sink-network partition occurs. This kind of partition makes the network inoperable even though the network resources are still available. Such energy depletion forms what is called an energy hole around the data collector. The energy hole is most likely unavoidable in networks dependent on stationary-based sinks. This is definitely true for networks with uniform node distribution and uniform data collection [2].

Maximizing sensor network lifetime is considered an optimization problem in linear programming. The problem has been proved to be NP-hard [3]. Approximation algorithms are developed to provide efficient solutions under certain settings. The dynamicity and the frequent updates in the network topology are usually unpredictable, which reduces the efficiency of such solutions in many cases.

Many other practical solutions have been considered to minimize the tendency of the nodes located near sinks to form an early partition. Examples of these solutions are (1) considering the dynamic clustering of the network [4, 5], and (2) intentionally performing non-uniform (controlled) node distribution over the coverage area [6]. Each of these mechanisms has its own achievements that contribute to enhancing sensor network lifetime. Clustering results in better load balancing that could contribute to enhancing the lifetime. Non-uniform node distribution could result in the assurance that nodes around sinks are able to carry the data generated from other far nodes. Although effective solutions, they have limitations and constraints in real world deployments.

Using different clustering techniques to enhance the network performance has been known for long time in the area of ad hoc networks as well as sensor networks. However, clustering algorithms are expensive in terms of their communication cost. They require extensive overhead that could contribute to draining the resources of sensor networks. On the other hand, intentionally performing controlled non-uniform node distribution is problematic and difficult to achieve in real world deployments. This is due to the expected large number of sensor nodes that usually need to be randomly deployed within the coverage area.

Recent advances in wireless communications and mechanical evolutions are enabling mobility and multiple radio capabilities to be incorporated for moderate prices [7]. This availability encourages us to take advantage of their power to overcome the aforementioned problem. Mobility could reduce hotspots and balance the energy consumption and, in turn, could contribute to enhancing the lifetime [8]. However, sink multiplicity and mobility complicate the design of the networking protocols. They raise many new problems. Some examples include controlling the mobility to achieve the required energy efficiency of the data collection task; solving the problem that results from the difference between the speed of moving the sink to collect the data versus the multi-hop transmissions of the data; and coordinating the mobility trajectories (in the case of multiple mobile sinks) to keep the positive effect of the interference of the mobility trajectories while collecting data.

Existing mechanisms that opt for exploiting mobility to maximize the network lifetime provide only partial solutions. In most cases, not all of the above issues are considered. Thus, a solution is still required that could result in composite benefits for all the above mechanisms while avoiding their limitations.

1.2.2 Contribution

In Chapter 3, we explain our contribution towards maximizing the sensor network lifetime by answering questions regarding how sinks could be deployed while moving and collecting data in a way that maintains the network connectivity and maximizes the network lifetime. We present a solution that combines the benefits of using multiple-mobile sinks and the non-uniform node distribution (without performing non-uniform node distribution). Moreover, we provide a strategy for data propagation segmentation that we call “away-from-centroid”.

This strategy leverages the load balancing without introducing extra communication overhead. We build our solution upon the following key goals:

a. *Maintaining the network connectivity*
b. *Utilizing the network resources*
c. *Leveraging the energy balancing*

To achieve these goals, we initially deploy the mobile sinks at the network peripheral. Then the virtual center of the network is used as a pulling force for mobile sinks at the same time as a disparate force for internal data propagation. We provide a dissemination strategy that allows the data traffic to be directed away from the network center (where the number of nodes is relatively small) toward the network peripheral (where the number of nodes is intuitively large as we go from a narrow to a wider area). The directed data propagations increase the load balancing through the network as it allows the traffic to spread over the whole network. Then the data reports will travel over the network perimeter towards the sinks. Eventually, perimeter nodes at the network peripheral will be exposed to a *peeling phenomenon or shrinking* (will have their energy depleted exposing other inner nodes to the perimeter), which results in partitioning one or more sinks from their one-hop neighbors. The partitioned sinks move discrete steps following the direction of the progressive peeling (i.e. from outer to inner layers towards the network center) and the connectivity is re-established.

In contrast to existing solutions, the protocol is totally dynamic. It is a localized protocol that does not require any global topological information. The overhead of exchanging topology updates and messages associated with sink mobility is not required. We show that our solution leads to a sub-optimal energy balancing through the network. The ability to dynamically maintain the sink-to-network connectivity, the balance of both load and energy consumption and the short distance that data need to travel leads to a significant increase in the network lifetime. We also investigate the "guard region" concept where a subset of the coverage area at the outer layer of the network could participate in protecting the core area of interest from shrinking. This concept raises the notion of coverage tolerance ratio (CTR) that depends on the application. The performance of the protocol is shown through intensive simulations that consider realistic conditions of the underlying network settings. Many experiments are conducted to compare the performance of our protocol with other protocols from the literature, such as MobiRoute [9]. The results show the efficiency of the proposed model in expanding the network lifetime, minimizing overhead associated with sink-mobility, and achieving a high degree of reliability. Our work presented above is published in [161],[162].

1.3 Adaptive Data Delivery

In this section, we describe the problem of developing a data delivery solution that maximizes the lifetime and adapts to both the irregularity and the application query-requirements.

1.3.1 Problem Statement and Motivation

Data delivery is the crucial task of deployed sensor networks. Different applications of sensor networks require different data delivery models [10]. Data delivery in sensor networks is usually classified into three categories: query-driven, event-driven, and hybrid [11]. Hybrid approaches mix both event and query based data delivery models. They aim at overcoming the limitations of the basic models when standalone so they are an important class of data delivery approaches in sensor networks. Yet they are not widely utilized in real world deployments such as the wireless vineyard [12], the botanical garden's mother redwood grove [13], the great duck island [14] and the remote ecological micro-sensor network [15]. The reason is the limitations of the existing hybrid approaches. One of the main limitations is that existing hybrid schemes are optimized for flat architecture in which each node could be a data consumer as well as a data producer [16]. Sink multiplicity and mobility complicate the design of these schemes. Coordination and load balancing become essential components of the devised protocols. Most of the existing solutions do not consider the nature of cooperation between the existing sinks. When sink mobility is considered, the design of efficient rendezvous data dissemination protocol for such architecture is more complicated. Existing solutions consider only random sink mobility where the movement of sinks is uncontrolled (e.g. mobile users). Another important limitation is that they are tightly coupled to the underlying topologies. Irregular network topologies present a challenge for these schemes and usually result in the performance degradation of such schemes [17]. Moreover, they cannot avoid wasting resources due to the "uncontrolled events propagation," the problem that originates in their underlying event-driven component. Current hybrid approaches (referred to as rendezvous approaches) adapt both event-driven (known as push) and query driven (known as pull) components that are working independently. With a long time-interval between successive queries, resources are wasted in processing unimportant events that are still propagated even though they are not of current interest. Resources are also wasted when the events are not queried before being discarded by the temporary destination nodes. These limitations narrow the spectrum where they can be applied. So developing an energy efficient rendezvous data dissemination mechanism that enhances sensor network lifetime is considered a challenging task that requires more attention.

1.3.2 Contribution

To widen the applicability of the current rendezvous mechanisms, we develop a novel efficient rendezvous mechanism with underlying localized algorithms that adapts to both the irregularity and the application query-requirements. Our work in Chapter 4 extends our solution for the event-driven data collection model in Chapter 3. The proposed solution is a completely different approach to the traditional rendezvous schemes. We propose a novel topology-adaptive, content-independent hybrid protocol for irregular, tiered, sensor network architecture. Our contribution towards developing this solution starts by considering the key limitations discussed above. We emphasize synchronization of underlying pull and push phases in the sense of making them aware of each other. This would prevent resource wasting and leverage on-demand data communication. Then we approach the topological adaptation by smoothly integrating a topology detection algorithm from the literature so that the devised protocol would be topology-adaptive. This allows the protocol to work with equal efficiency in both regular as well as irregular sensor networks. To the best of our knowledge, most academic work considers topology detection and data dissemination separately, even though data dissemination at the upper layer is greatly affected by the underlying protocols. We show that, by integrating them, we can provide robust rendezvous data dissemination. Moreover, the proposed protocol is operable and optimized for sensor networks tiered architecture. The existing sinks cooperate as a single tier to build a reliable rendezvous infrastructure that maintains efficient communications. When these sinks are mobile, the mobility could be controlled according to our proposed strategy presented in

Chapter 3. The ability to adapt the dissemination behavior to the query types and the dynamicity of the underlying phases produce a robust operational network. This increases the interesting features that could make it the model of choice, especially for remote sensing applications.

To illustrate the efficiency of the proposed solution, we provide a generic analytical model. Moreover, we also give a comparison based on a grid analysis model to compare our solution to two benchmarks protocols that exemplify both query-driven and hybrid schemes. Simulation experiments are also conducted to compare the performance of the proposed solution to the benchmark Two Tier Data Dissemination (TTDD) protocol [18]. Both analysis and simulation results show the robustness and efficiency of the proposed solution in any network topology whether regular or irregular. Results also show the reduction in the communication cost and the even distribution of events propagation through the network. Our research work presented in this section is published in [163],[164], [165].

1.4 Generic Sensor Network Infrastructure

Sensor networks are known to be ad hoc with no static infrastructure [19]. In scenarios where all nodes are static with no ability to move, nodes build logical infrastructure by self-organizing. In this section, we describe the problem of building generic sensor network infrastructure and our contribution towards achieving it.

1.4.1 Problem statement and Motivation

To date, the main regime that is assumed by the protocol designers is the development of data communication protocols that are coupled with an underlying logical infrastructure. This is the reason that most of the current proposed protocols for WSN are described as cross-layered protocols [20]. These protocols are in most cases tied up to specific communication algorithms. The underlying communication algorithm in turn optimizes for a specific system architecture.

Clustering and spanning trees based protocols are forms of self-organization approaches that assume this regime [21]. These approaches result in unstructured overlays. The operations over such overlays are usually based on flooding. Failure handling and maintenance require cascade updates through the network. In addition, the optimization of such approaches is usually towards a specific process, such as routing.

On the other hand, logical overlays such as DHTs are not suitable in the context of sensor networks [22]. The main reason is the belief that DHTs overlays would produce extra overhead compared to the benefits they provide to the upper layer applications. When mobility is considered, the movement of the nodes may quickly change the topology. This results in the increase of the overhead messages for topology maintenance and movement management.

Increasing the reusability of the designed protocols requires decoupling the underlying communication primitives from the upper layer protocols primitives [23],[24]. To achieve this goal, recent research builds generic infrastructure at the level of physical links The problem with building such infrastructure is that it is challengeable and is still open, especially considering that sensor networks are application-oriented. The infrastructure should be: (1) generic enough to be leveraged by upper layer protocols with equal efficiency; and (2) flexible enough to support both multi-hop and data mule-like [25] communication regimes. As a result, a solution is required that attempts to build a structure to fill in the space between the physical and logical structures.

1.4.2 Contribution

In Chapter 5, we present a localized mobility-based protocol that organizes the physical links of the network in generic infrastructure that could be leveraged by multiple upper layer protocols. The protocol builds a structure that fills in the space between the physical and logical structures. This kind of overlaying was difficult in networks such as the sensor network. We focus on the deployment scenarios that involve both mobile and static nodes. We consider the case where most nodes are static and only a few powerful nodes are mobile robots/sinks. Our vision is that these robots could take the burden of being the network organizers in addition to being the data collectors and management infusers. They would act as *moving probes* that would move to access the data at its position in addition to accessing the data that hops to them through the multi-hop regime.

Our contribution in Chapter 5 could be summarized as follows: (i) designing a protocol (layered infrastructure protocol LIP) that constructs an infrastructure (Circular layered infrastructure CLI) that adheres to the design objectives; (ii) designing generic communication primitives that provide multiple communication configurations to the upper layer protocols; (iii) providing a customized cost model that gives insight into the cost of communication over CLI; and (iv) validating the performance of CLI in supporting upper layer protocols.

The resulting CLI infrastructure keeps the proximity of the nodes. Nodes that are neighbours in the infrastructure are physically neighbours. Each layer in CLI is assigned a mobile robot that acts as a probe to access the data and monitor the layer. Access positions are selected dynamically at each layer to provide anchors for the probes to visit at their associated layers. CLI supports both multi-hop and data-mules regimes of communication.

The operations of the LIP protocol to build the infrastructure are localized (i.e. based only on one-hop neighbor information). In addition, topological changes trigger only local updates. This provides CLI with the outstanding ability to cope with failures and requires only local updates for maintenance. We prove both the correctness and efficiency of our solution. A rough cost model that gives insight into the cost of the communication over the CLI is provided. The performance of the resulting CLI infrastructure is evaluated by the implementation and the simulation of some of the upper layer processes. We also provide simulation-based comparisons to a multi-scale communication approach [26]. The results show the robustness and efficiency of CLI in supporting the upper layer processes. The research work presented in this section is published in [166].

1.5 Summary of the Research Objectives

As mentioned in the previous section, our thesis aims to investigate the impact of enabling sinks multiplicity and mobility on the design of efficient communication protocols for large scale sensor networks. More specifically, the key research objectives of this thesis can be summarized as follows:

- To provide a novel solution for maximizing the sensor network lifetime based on leveraging the positive impacts of enabling sinks multiplicity and mobility while avoiding the implications that they raise in the protocol design.
- To develop a novel data delivery protocol that better utilizes the network resources and overcomes the limitations that exist in the current solutions.
- To highlight the current limitations towards increasing the reusability and modularity of the existing communication protocols for sensor networks, and to develop a novel protocol for arranging the arbitrary sensor network topology into a

generic infrastructure that could be leveraged by multiple upper layer communication protocols.

- To investigate and study the requirements that could ease the development of different protocol categories over the generic sensor network infrastructure and to provide the basic primitives for facilitating such development.
- To provide an empirical as well as theoretical evaluation of the proposed protocols/algorithms in terms of the performance metrics, such as energy consumption and success rates and reliability, and to compare against well-established research work.

The proposed protocols adhere to the design criteria shown in Figure 1. These design criteria go along with the requirements of efficient protocol development for sensor networks [27]. Considering the constrained environment of wireless sensor networks, the proposed protocols aim to achieve the following main design objectives:

- **Energy efficiency** In wireless sensor networks (WSN) the nodes should work unattended and unwired. They highly depend on their equipped power supply, which is too limited. Power limitation and transmission range limitation are exhibited by sensor nodes and result in high energy-constrained networks. So energy consumption is a very critical constraint in wireless sensor networks. This characteristic of sensor networks motivates the need for developing new protocols that compromise and adapt to this constraint. The proposed protocols achieve the goal of energy-efficiency by balancing the load among nodes and reducing the number of transmissions required to perform a given task.
- **Scalability** The constrained resources in wireless sensor networks turn the scalability into a challenging objective. The proposed protocols achieve this design criterion by

being *localized*, which means that no global information is required by any node to act or take a required decision. Nodes depend only on knowledge about their one-hop neighbours to achieve the global behaviour. The proposed protocols are able to adapt to the increasing network size, in terms of the number of sensor nodes and sinks, while maintaining a high degree of load balancing.

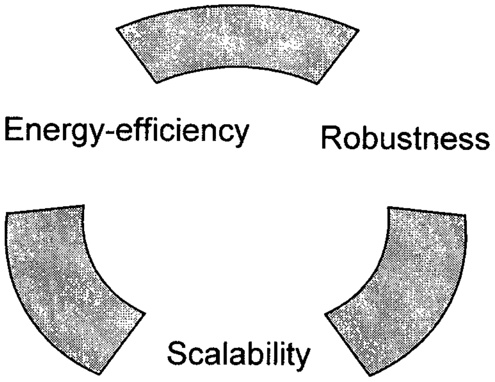


Figure 1.1 Design criteria

- **Robustness** WSN is known to be vulnerable to increasing node failures and topology updates. The proposed protocols involve mechanisms to overcome failures and adapt to topological changes. A localized failure handling mechanism is involved in the design of the proposed protocols. Moreover, global updates are completely avoided to keep the efficiency of the proposed protocols.

1.6 Methodology of Evaluation

The evaluation of the proposed protocols is done through intensive simulations. The wireless sensor network simulator (WSNS) from the literature is used during simulation. The details of the simulator are described in [28]. The simulator uses a multithreading mechanism to allow accurate network configuration in addition to attractive visualization ability. The proposed protocols are built above the infrastructure layer that defines the coverage-area dimensions, power model, model of nodes' distribution, and the transmission/sensing ranges. The behavior of the protocols is tested in both dense and sparse network settings. We used fixed network terrain dimensions and changed the sensor densities within the terrain. The distribution of nodes is uniformly random based on the built-in Random[1] class of the underlying C# programming language. Different radio ranges are used to maintain the connectivity of the network with the changed densities. Experiments are conducted with different numbers of deployed sinks that are assumed to have no restriction in their power supply. In all experiments, the initial energy of the sensor nodes is equal. We study the energy efficiency of the proposed protocols by considering the limited energy capacity of sensor nodes and measure the number of transmissions required for achieving the given tasks as a reflection of the energy consumption. We also investigate the unreliability of sensor networks by modeling the frequent temporary node failures and the ability of the given protocols to survive under different failure models. Protocol specific measurements that reflect a certain aspect of a given protocol are also considered and will be explained in detail in the context of the associated protocol. To ensure the consistency of the results, all of the conducted experiments are repeated 5 to 10 times and the results are obtained as the average of the resulting outputs. Then standard error bars that represent {+,-} 2 σ_m (the standard variance of the mean) are utilized to show the confidence level of 95%.

---

[1] According to Microsoft.net® "This class generates Pseudo-random numbers that are chosen with equal probability from a finite set of numbers. The chosen numbers are not completely random because a definite mathematical algorithm is used to select them, but they are sufficiently random for practical purposes. The current implementation of the **Random** class is based on Donald E. Knuth's subtractive random number generator algorithm. For more information, see D. E. Knuth. "The Art of Computer Programming, volume 2: Semi numerical Algorithms". Addison-Wesley, Reading, MA, second edition, 1981".

1.7 Thesis Outline

The rest of this thesis is organized as follows: Chapter 2 provides the literature review and the related work; Chapter 3 describes our deployment-based solution for prolonging sensor network lifetime; Chapter 4 describes the proposed scalable rendezvous data dissemination approach for irregular, tiered, sensor network architecture; Chapter 5 presents the proposed mobility-based protocol for building generic infrastructure for sensor networks; Chapter 6 concludes the thesis with a summary and a discussion about the conducted research and the future research related to this thesis.

Chapter 2

Literature Review and Related Work

This chapter introduces state-of-the-art research directions in the wireless sensor network that are considered the current hot issues in the area. In particular, data gathering and mobility and sensor network infrastructure are investigated. Current solutions are explored and discussed. This chapter also addresses the contributions of this thesis towards solving the problems specified in Chapter 1 relating to these issues. Relevant and related works are highlighted and comparisons are given in the context. This chapter is organized as follows: Section 2.1 provides a generic background to introduce the wireless sensor network (WSN); Section 2.2 shows the significance of the data gathering process in WSN and explores the different mechanisms in the literature; Section 2.3 shows the impact of enabling mobility in sensor network architecture; Section 2.4 shows the impact of the infrastructure on the protocol development process and discusses the current direction towards increasing reusability and modularity of the devised protocols for the sensor network; and Section 2.5 gives a discussion and a brief summary.

2.1 Background

The Distributed Sensor network has emerged as a sub-class of the ad hoc network. It is defined as a large collection of small inexpensive devices used to collect data from the physical world [29]. The network acts in a similar way to a network of processors with smart input peripherals that have the ability to sense. The network nodes are usually deployed in the field of interest either deterministically or randomly. These sensor nodes sense the physical world and process their measurements by their local processors. The communication among them is usually over a restricted wireless medium. Nodes cooperate to perform the assigned tasks by partially processing their own data. Application of such technology is widespread and includes traffic control, monitoring, health applications, tracking, manufacturing, earth activities and many others [30]. It is not a surprise that wireless sensor networks attract researchers from different disciplines, such as hardware design, communication, embedded systems, distributed systems, real time systems, multiprocessing, and parallel processing.

Wireless sensor networks show some similarities to Mobile Ad hoc NETwork MANET. The differences between them hinder the direct applicability of the solutions devised for MANET to the problems associated with the current WSN design [19],[27]. One of the main differences is the **services** exchanged between nodes. In WSN, nodes do not require sophisticated services from each other and the core task is the data gathering. The direction of the data transfer is usually toward a central unit, which is almost always called the sink. So the communication in WSN is **point-to-multipoint** rather than point-to-point. Another important difference is the **degree of mobility** and its impact on the whole behavior of the WSN. While mobility in MANET highly contributes to the failure rate, it seems to have less impact on the kind of failures exhibited by the sensor network. In WSN, sensor nodes are assumed to be stationary after their deployment. Mobility is usually assumed by only a few nodes (mobile sinks). In general, failures in WSN are mostly due to channel error or topological changes that result from malfunction nodes, obstacles, or physical damage [31]. Moreover, the number of deployed nodes in WSN ranges from hundreds to thousands, while in MANET it usually ranges from tens to hundreds. One of the most important features of wireless sensor network WSN is that nodes work unattended and unwired. They highly depend on their equipped power supply, which is too limited. Power limitation and transmission range limitation exhibited by sensor nodes result in a high **energy-constrained network**. So energy consumption is a very critical constraint in wireless sensor networks [32]. This characteristic of the sensor network motivates the development of new protocols that compromise and adapt to this constraint.

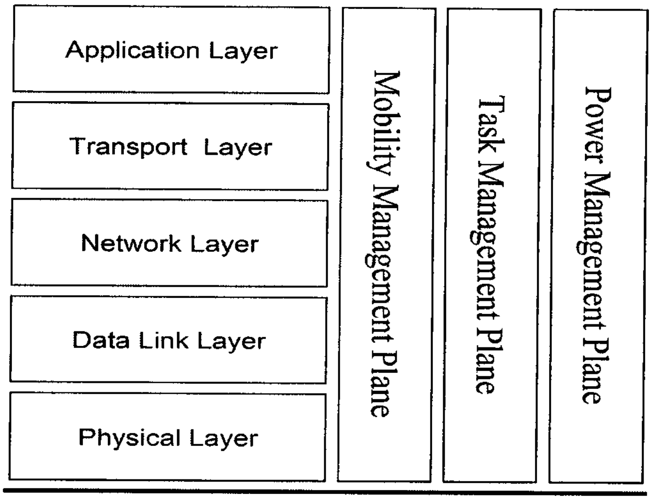


Figure 2.1. Sensor network architecture layers

WSN is **application-oriented** in the first place. The variety of WSN design choices has a great impact on designing the networking protocols. For example, the **data delivery model** required by the application could be: event-driven, query-driven or periodic updates [11]. The underlying network could involve mobile sinks or stationary ones. Sensor nodes could be homogenous or heterogeneous. The network could rely on one sink or on multiple sinks. The **deployment** of sensor networks guided by its underlying application could be random or structured. This variety of choices implies that there is no one general solution that could adapt to all the dynamics involved in the design. So implementing the right networking protocol for the sensor network requires a deep understanding of the nature of the application [10]. Consequently, researchers are trying to achieve the best design regarding architecture, topology, routing, tasking, and querying. There are many issues that are currently being addressed, but standard solutions are not yet being maintained. Solutions are still required that offer the quality of services needed by the application and, at the same time, adapt to the constraints related to the current sensor nodes hardware design. Currently, researchers have agreed upon the architecture layers that reflect the requirements of efficient protocol designs for the sensor network. As depicted in figure 2.1, those layers are the physical, data link, network, transport, and application layers. Three basic management planes are also considered in the network stack to allow management process to be distributed over the whole layers. Management process is concerned with power, mobility, and tasking.

At each layer of the stack, many efforts have been taken in order to enhance the performance of the network. For example, the use of a more robust power source or a self-supplying source, such as solar cells, is investigated at the physical layer [33]. The concept of scheduling is investigated at the data link layer so that nodes that are not currently interested in processing just need to turn to the sleep modes in order to conserve their power [34]. Above the data link layer are many power aware routing mechanisms, which are invited to work at the network layer. Also, techniques such as data aggregation [35], which is the concept of combining data packets into a smaller number of packets, are utilized with the goal of minimizing power consumption. However, there is a wide belief in the WSN research community that the restrictive obeying of this layered architecture could increase the overhead of the proposed protocols and could contribute to draining the WSN of limited resources. Therefore, most of the current proposed protocols in WSN are cross-layers approaches that span multi-layers of the network stack [20]. One important research issue that is still open is whether the application-independent techniques will be more robust than those which assume application-specific techniques. To the best of our knowledge, the mechanisms that demonstrate good performance for certain kinds of applications perform very poorly when the assumptions about their virtual environment are changed.

2.2 Data Gathering in Sensor Network

Data gathering is the crucial task of the sensor network [19]. The data processing scenario within sensor networks starts with nodes acquiring data from their physical environment. This data acquisition process could be either a response to a query issued by the sink or based on a particular schedule.

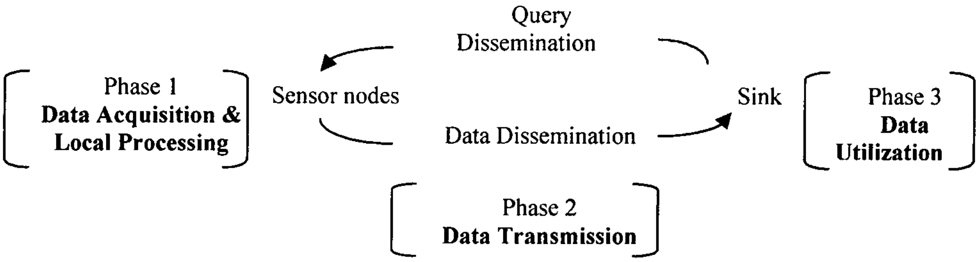


Figure 2.2: Data manipulation phases within sensor network

Figure 1 shows the different data manipulation phases following the acquisition phase. The local processing phase usually involves data storage and data aggregation. These concepts will be explained in the following text. At the data transmission phase, data is transmitted according to the dissemination protocol and the underlying routing protocol. Once the data is received at the final destination, further processing may be applied before it can be utilized by the application. Although nodes consume their energy along all the above phases, the transmission phase is the most expensive one [32]. In order to reduce the amount of energy dissipation along this phase, all data gathering protocols that are used for sensor networks must take this constraint into account.

Data gathering in WSN is sub-divided into three main topics: **data storage, data aggregation, and data reporting** [36]. Data reporting refers to the tasks of routing and dissemination. Many mechanisms have been introduced to cover the above issues in the constrained wireless sensor network. The main notice is the variety of assumptions about the underlying environment. Researchers are still working towards standardized techniques that could adapt to the high degree of dynamicity offered by the sensor network. This section investigates some of the already existing data gathering protocols with their underlying related issues.

2.2.1 Data Storage

Storage mechanisms have a high impact on the availability, accessing, and accuracy of the queries' results. They also influence the scalability of the network as well as the communication protocols since they affect the rate of energy consumption due to data transmission. Each node within the network has its own local storage, which is a limited resource. Three storage mechanisms are explored within the literature [37]. The acquired data is assumed to be locally stored and waiting to be queried. This is known as a local data storage mechanism. Another alternative is that data could be sent directly to the sink nodes on the fly once acquired. This is referred to as external data storage. The third model is the hybrid storage model where the data can be stored within the network, but at nodes other than those acquired for them. These mechanisms take advantage of the local storage at the nodes and aim to reduce the total energy consumption.

Another way to enhance the storage capability of sensor networks is to augment the network with particular storage nodes. This results in reducing the load by transmitting all the data to the sink and hence reduces the communication cost of the query process. For example, in [38] authors examine the deterministic placement of storage nodes. They provide dynamic programming based algorithms aiming to minimize the total energy cost for data gathering. Another issue related to the data storage problem is the storage overhead required by the different protocols. Most of the protocols for sensor networks may require meta-data to be stored at each node. Essential concerns might then include whether each node maintains its own copy of the meta-data, what is the lifetime of stored data, and how frequent is the need for replacement mechanisms.

2.2.2 Data Aggregation

Aggregation is introduced within sensor network data dissemination protocols in order to reduce the energy consumption due to exchanging large numbers of packets between nodes. The simplest mechanism is called direct delivery [39] in which nodes are required to report their data individually to the sink. While simple, it is problematic. Communication in WSN is multi-hop-based so data will be sent to the sink through intermediate nodes. While each node individually sends its data to the sink, large numbers of packets will be exchanged between nodes increasing the transmission cost and the redundancy.

The key behind aggregation is that the local processing of data consumes less energy than the data transmission [40]. Moreover, nodes are equipped with local processors, which could participate efficiently in the global processing goal of the network. Most of the data dissemination protocols allow intermediate nodes to perform partial processing of the data in many forms. For example, nodes could filter their own data and integrate them with the data received by their neighbor nodes [41].

Data aggregation protocols are affected by the underlying communication protocols as these protocols will determine the data flow paths within the network. Aggregation could be performed implicitly by the dissemination protocols (i.e. the way of disseminating the data determines how it will be aggregated). Also, it may be independent and general so that many routing protocols could be applied [42].

The timing pattern [43] also has a significant impact on the design of the aggregation mechanism for periodic update applications. There is a trade-off between data accuracy and freshness and energy efficiency. Timing pattern allows data synchronization and determination regarding when data is to be sent to the other nodes. Timing models could be classified to simple periodic, periodic per hop, and periodic per hop adjusted. In simple periodic, a node waits a predefined amount of time before aggregating all the data it has received and then sends the data as a single packet. In periodic per hop, each node waits until it receives data for all its children, aggregates the data, and then forwards the data. Timeout must be used in the case of existing failed children. Periodic per hop adjusted is similar to the periodic per hop approach but the timeout is adjusted based on the position of the node in the routing tree. Nodes that are lower in the routing tree experience a timeout before those nearer to the host (sink).

Another important factor that affects the design of the aggregation protocols is the nature of the aggregation itself. There are many ways in which aggregation could be performed. For example, aggregation could take the form of compression, concatenation, encryption, or coding [44]. Most of the aggregation protocols can be seen as complementary approaches that could be integrated into other dissemination and communication protocols without hindering their operations.

2.2.3 Data Dissemination and Communications

Dissemination protocols determine the way data will be disseminated through the network. Communications in sensor networks are characterized in a multi-hop fashion. Nodes are scattered in the investigated area in a large number. A node within the radio range of other nodes is described as a neighbor node and can directly communicate. Non-neighboring nodes require multi-hop communication protocols to communicate. Communication protocols for sensor networks must be energy-aware and must optimize the transmission cost of data packets. So the number of transmissions required by each node and the number of states maintained at each node should both be minimized [19].

There are many important factors that affect the design of dissemination and communication protocols for sensor networks [45]. These factors can be summarized as follows:

- Non existence of specific routers; All nodes are working as small routers from the source to the destination.
- Nodes are equipped with limited energy, computation, and communication resources.
- The identities of nodes are less important than the data they carry.
- Nodes are location-aware.
- The flow of the data within the network is mostly directed toward single sink nodes and the collaboration between nodes is usually in the form of data aggregation.
- Nodes do not usually require sophisticated service from each other.

- The efforts of data dissemination protocols are focused by eliminating data redundancy to allow only information of high quality to be received by the sink nodes.

Data dissemination and communication protocols in WSN are described as **data-centric** protocols [16]. This is due to the fact that the data carried by the node is more important than the identity of the node. In most cases, these protocols are application-dependent. The protocol usually starts with an interest (which describes the attributes of the phenomena) being sent through the network, either initiated by the sink or advertised by the nodes. All the high level layers of the communication stack must be aware and must agree upon the formatting structure of these interests. Thus, data centric protocols require naming mechanisms for specifying the interested events or attributes that describe the interested data [46]. All of the data-centric approaches use such application specific semantics so they cannot work as general approaches. Many researchers are currently trying to achieve a general approach that could work independently from the applications and, at the same time, could highly contribute to minimizing the energy consumption.

Data centric approaches could be classified according to the underlying dissemination structure to be either ***flat*** or ***hierarchical*** protocols [47]. In flat approaches, the transfer of data is usually performed in a flat space as all nodes are working at the same level. In hierarchical protocols, nodes are organized in multi-levels. Nodes could be homogenous, having the same capabilities, or heterogeneous, having certain nodes with more computation facilities than the other nodes and working as cluster heads. Data aggregation could be utilized at the cluster heads. The cluster heads could be elected autonomously or they may be pre-selected. Hierarchical protocols contribute toward the scalability of the network and reduce the power consumption.

Whether flat or hierarchical, the protocols could be further classified according to the data delivery model they assume. The data delivery model describes the data reporting mechanism, which could be: *event-driven*, *query-driven,* or *hybrid* [11]. In event-driven, data reporting occurs upon detecting pre-specified events. The nodes themselves initiate the reporting action. In query-driven, data reporting occurs upon receiving a query (request) from the user through the sink. In hybrid models, a mix between the two strategies is applied. We will discuss such approaches in a separate section for their importance to our research work. Apart from the dissemination structure, protocols that utilize the location information of the nodes are known as geographic protocols. Most of the data-centric protocols assume stationary nodes. Sink mobility is considered only as a side effect that should be mitigated. Flat, hierarchical, geographic and mobile sink protocols are discussed in the following subsections.

2.2.3.1 Flat Approaches

To overcome the problems exhibited by flooding-based mechanisms in WSN, Sensor Protocols for Information via Negotiation SPIN [48] protocol introduces the idea of using meta-data to allow only useful data to be transmitted based on negotiations between nodes. The basic idea is that the transmission of meta-data, which is less in size than the actual data, would consume fewer resources than those required for transmitting the actual data.

The actual data are assigned high level names that describe them. These names are used to advertise the data by sending advertisement packets. The protocol uses the flooding mechanism to send these advertisements before the actual data being transmitted to the nodes is requested.

The protocol assumes homogenous sensor networks where all nodes could be potential sinks. So any node could be a source, as well as a destination, of the data. The protocol works well when only a few nodes request the advertised data. If all the nodes are interested in getting the data, the performance of SPIN would be worse than that of flooding. In such cases, the advertisement packets and the data request packets would create extra overhead and more energy consumption.

Another famous protocol is the Directed Diffusion (DD) data-centric dissemination [49]. The protocol uses a distributed data centric algorithm for routing data based on localized information. Nodes need to know only about their one-hop neighbors. The protocol assumes a query-driven data delivery model in which an interest (task description) must be sent from the sink to the nodes. The interest carries with it some gradient fields that help construct the delivery path. Each node receives the interest, stores an entry into local cache, and forwards the interest to its neighbors. Once matched data is discovered, nodes start to transmit the data through the constructed paths to the sink, which selects the best delivery path by reinforcement. Reinforcement requires the sink to re-send the same interest with a request for a higher delivery rate. During data delivery, the intermediate nodes can perform data aggregation to enhance the performance. Many paths will be constructed between the source and the sink and the data will be delivered along all of them with different rates. As in [48] the algorithm does not involve sink identification so it does not distinguish between different sinks. An improvement to DD is suggested in [50] to enable quick path repairing. The idea is to maintain backup paths in addition to the primary data forwarding path. In case of node failures, these backup paths are used to cope with the failure.

Another protocol that assumes a data-centric approach is the Temporal coherency-aware in-Network Aggregation TiNA [51]. The protocol focuses on the energy saving in sensor networks, while trying to keep the data accuracy as high as specified by the user. The protocol uses temporal coherency tolerances in addition to in-network aggregation mechanisms in order to achieve its goal. The user should specify a tolerance value that describes the amount of diversity between consecutive readings in order to allow nodes to report their data. The nodes store the previous readings and compare them to the current new readings. The data would be reported only if the difference between the two readings satisfies the tolerance threshold. The protocol allows parent nodes to use the last reading received from any child node in place of a new reading required by that child in case of failure. The authors assume that the temporal coherence tolerance value reflects the user preferences about the required data and determines the quality of data QoD submitted to the user.

The idea of using declarative queries as an abstraction is introduced in COUGAR [52]. Queries are then processed through a query layer above the network layer. The sink is responsible for creating a query plan. This plan should be self-contained and able to specify the data flow and the in-network computation. For each query plan, nodes should select a leader node that is responsible for data aggregation and data reporting to the sink node. Some of the drawbacks of this approach are the dynamic selection of leader nodes, the overhead due to the introduction of the additional query layer on each sensor node, and the requirement of synchronization between nodes

ACtive QUery Forwarding In sensoR nEtwork ACQUIRE [53] is proposed for complex quires (queries that consist of several sub-queries). First, nodes receive the query from the sink and try to respond partially to the submitted query by using their pre-cached information. If such information is unsuitable for resolving the query, nodes look up the relevant information at their neighbors within d hops. The query results should then be sent back to the sink through the reverse path or the shortest path. The parameter d is adjustable such that if d is equal to the network size, the mechanism behaves in a way similar to the flooding mechanism. The optimal value of d is calculated for the grid deployment scenario of sensor networks.

The data reporting phase in all the above approaches flows in the reverse direction to the paths constructed during the query dissemination phase. Nodes learn the path to the sink by memorizing and keeping information about the neighbors from which they receive the query. Such link memorization adds extra overhead in case of topological changes due to link failures. In most cases, per sink information should be kept at each node. Sink mobility requires reconstruction of the data flow paths so it is not well addressed by these approaches.

Other flat-based data centric approaches adapt other techniques to report the data to the sink. A technique called cost-field based reporting that does not require the nodes to memorize the reverse paths to the sink is presented in [54], [55], [56]. A cost value is calculated during the query flooding phase. Each node keeps a cost value of reporting to the sink. The selection of next forwarding neighbor could be receiver based or sender based. In receiver based schemes, a node transmits the data to its neighbors without specifying the next forwarder. The neighbors that have cost values less than the sender transmit to their neighbors and so on until data arrives at the sink node. In sender based schemes, the sender transmits the data to a specific neighbor for further transmission until data arrives at the sink. The cost filed could be calculated based on hop-count, energy level, or distance to the sink.

2.2.3.2 Hierarchical approaches

LEACH [57] is one of the early and most popular hierarchical protocols for sensor networks. The system runs a periodically randomized distributed algorithm to elect a number of cluster heads among the nodes. This number is determined a priori. The CHs then advertise themselves to the nodes. Nodes join certain clusters based on the strength of the received signals. The sensor nodes then report themselves to the CHs to join their clusters. Each CH could create a TDMA schedule to control the transmission within its cluster. Data aggregation within each cluster is performed at the CHs to save energy. Leach assumes that each sensor node could adjust the transmission power so that it can reach the sink by using the maximum transmission power. Using this assumption, nodes directly report to the CHs and the CHs are able to report to the sink directly using one-hop transmission.

Passive Clustering (PC) [58] is a clustering mechanism that has been introduced for MANET. PC aims to eliminate the problems exhibited by the ordinary flooding mechanism when high rate of mobility is considered. PC utilizes the IP option field to piggyback the clustering related information (e.g., cluster status of the node and the node ID) to the ongoing traffic. This reduces the rate of exchanging cluster-related control packets. Its applicability to sensor network is similar to that of dominating set. They aim to provide efficient broadcasting. PC has been integrated to several MANET routing protocols such as ADOV [59]. It has also been utilized by Directed Diffusion (DD) in sensor network [60]. In both cases PC is utilized to reduce the communication overhead incurred in their flooding phase (i.e. during the search for route phase).

In TTDD [18], each source node proactively builds up a virtual grid infrastructure over the two dimensional plane. The grid should consist of cells of dimensions X * X where X is a global parameter. The source node should be at a crossing-point in such a grid. The crossing points of the grid are called dissemination points. The sensor node closest to the dissemination point on the grid serves as a dissemination node. When the source node needs to report data (e.g. upon detecting an event) it propagates the data announcement to all the dissemination points. Dissemination nodes store information about the data announcement and the dissemination points they serve in addition to the upstream dissemination nodes locations. Then they further propagate the data announcement to the other dissemination points except those from which the data is received. Eventually data announcement covers all the dissemination points in the grid. When the sink node has a query to send, it floods the query locally within the cell where it is located. The nearest dissemination nodes for the requested data forward the query to the source node through the upstream dissemination nodes. The source then replies to the sink using the reverse path to the query path.

Siphon [61] suggests augmenting the sensing field with additional powerful nodes. It introduces the idea of virtual sinks (VSs), which are powerful nodes distributed randomly within the field and are able to communicate directly with each other. These VSs form a secondary network by themselves. Nodes divide themselves among the VSs in the form of clustering. Data reporting is done in two phases: nodes report to their associated VS, and the VS reports within the secondary network to the sink. As a result, Siphon works in a similar way to LEACH with the difference that CHs in LEACH are ordinary nodes that are elected randomly.

The concept of virtual sink architecture, in which sensor nodes forward data to one or more spatially diverse sinks, is also proposed for underwater sensor networks in [62]. The authors proposed a multi-path data delivery method for a grid network with multiple sinks. The main goal of the protocol was to provide a routing operation that could cope with contention.

Both flat and hierarchical approaches allow for data aggregation. In hierarchical approaches, cluster heads could perform the in-network aggregation. Hierarchical approaches contribute better to the scalability of the network. In comparison with the flat-based approaches, these approaches exhibit trade-offs between overhead and low energy consumption, as well as complexity verses performance. Hierarchical approaches require extra overhead to maintain the hierarchical structure. They also require global and local synchronization, which add to their complexities. Whether cluster heads are elected nodes or specific powerful nodes, other ordinary nodes need to agree upon their cluster heads. As in flat-based approaches, mobility impacts the robust operation of these approaches.

2.2.3.3 Geographic protocols

In wireless sensor networks, nodes need to know their location within the sensing field in order to provide meaningful information [19]. So there is always an assumption that sensor nodes know their locations and could tag their data by such information. Geographic approaches efficiently utilize the location information to perform the dissemination task [63]. They usually build over geographic routing schemes, and can be classified as weak or strong geographic protocols [37]. In weak geographic protocols, data can be directed and sent to a certain region. In strong geographic protocols, nodes can send data to a particular location. All of the geographic routing protocols require each node to know only its location plus the locations of their neighbor nodes. These approaches work as complementary approaches when the location of the source nodes that are required to report their data are known in advance [64], [65], [66]. In deployment scenarios where the sink is static, the location of the sink could be used to allow nodes to report their data ("push data") to the sink using geographic routing. We describe Greedy Perimeter stateless Routing GPSR [67] and GEAR [68] as representative of these protocols.

In GPSR only local information about the position of the nodes is utilized. Each node needs to know only its position and the positions of its immediate neighbor nodes. The packets to be transmitted should include the positions of the destination nodes. Upon receiving the packet, the node enters a greedy forwarding mode in which it tries to send the packet in a greedy way to a node that is closer to the destination than itself. When greedy mode fails (i.e. there is no closer node to the destination than itself), the node converts to the perimeter mode to try to cope with the failure. GPSR uses the right hand rule in which packets tour enclosed faces on planner sub-graphs. Once a failure is solved, and a node in the way of the destination is found, the node returns back to its greedy forwarding mode. The algorithm continues the same way until the packets are received by their final destination.

The use of geographic information while disseminating queries is studied in [68]. The main goal is to enhance the performance of DD under the assumption of known data sources positions. Instead of transmitting the interests to the whole network, GEAR considers transmitting it to a certain region so conserving energy. The protocol allows nodes to determine two costs: an estimated cost based on residual energy and the distance to the destination; and a learned cost based on the consideration of routing around holes. Packets are forwarded to the target region using ordinary greedy forwarding. If holes exist, the next forwarder node is selected based on the learned cost. Within the target region, packets are flooded or recursively geographically forwarded.

2.2.3.4 Mobile Sinks Protocols

All of the above representative protocols assume stationary nodes. There are some other data dissemination protocols that consider mobility issues. In [69] the authors proposed a Scalable Energy-efficient Asynchronous Dissemination SEAD protocol. In this protocol, sinks will be able to move around the network. It does not use mobile sinks as intermediate members of the dissemination tree so frequent changes to the dissemination paths will be reduced. Also, those sinks will be joining the dissemination tree by sending a join query to selected sensor nodes that will be considered as their access points. The access points are stationary nodes that are taking the place of sinks in constructing the dissemination paths. Those access points on one side will keep track of the mobile sinks and, on the other side, will replace the sinks from the point of view of the tree. So data are delivered to fixed points along the network, yet the sinks are mobile. Sinks request the data from the sources, periodically using refresh rates assigned to each access point and the data will be replicated at selected nodes between the source and the sinks.

TTDD [18] described under the hierarchical class is another protocol that considers mobility. TTDD assumes another form of sink mobility where sinks are mobile users. SEAD and TTDD belong to the early work that considers sink mobility as a negative property that should be mitigated. The main concern was how to route the data to the mobile sink/user rather than exploiting such mobility to enhance the network performance. This recent research direction is discussed in section 3.

2.3.4 Hybrid Data Dissemination Protocols

Hybrid data dissemination approaches are an important class of data dissemination approaches in sensor networks. Hybrid refers to the fact that such mechanisms mix different basic strategies of data dissemination solutions. From the point of view of the data delivery model, they mix the query based strategy and the event based strategy. From the point of view of storage, they involve a storage mechanism that is in between local and external storage. The hybrid approaches are usually motivated by the trail to achieve the balance and to overcome the drawbacks of the basic strategies. This is referred to as rendezvous mechanisms. The hybrid approaches could be classified as either content based or content independent. Figure 2.5 shows the classification of such approaches with some examples of each class. The literature contains many of these hybrid information-processing systems. Rumor Routing [70], Geographic Hash Table (GHT) [71], DIM [72], KDDSC [73], Comb-Needle (CN) [74], PathDC [75] and the approach in [76] are among them.

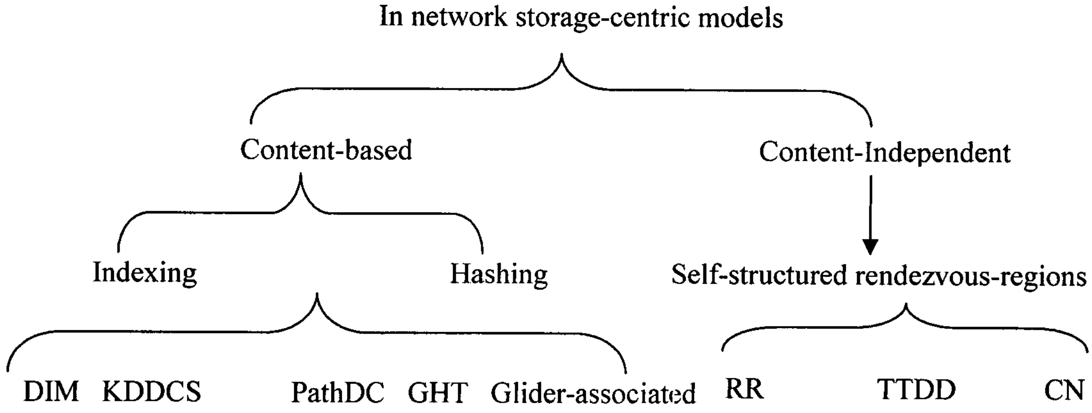


Figure 2.3 classification of hybrid data dissemination

In content-based models, [71], [72], [73], [76], the owner sensors are usually defined by a mapping function linking the content of the data with a location or ID within the network. In content-independent models, [18], [70], [74], events and queries are sent to self-structured sub-network regions. Querying the network in hybrid models relies on the concept of "rendezvous". Events travel to the rendezvous regions where they will be stored; queries are directed to the rendezvous regions on demand.

Data-Centric Storage/Geographic Hash Table (GHT) [71] is a popular pull-push mechanism based on Greedy Perimeter Stateless Routing (GPSR) [67]. Events are hashed (by their names) using predefined hash functions to locations inside the network topology. The events then travel to those locations and queries are similarly directed to the same locations. GHT requires periodic beacon signals so that nodes can update their neighbor table. The beacons are periodically transmitted through the network independent of the arrival of the events and queries. Network irregularities increase the failure rate of the underlying GPSR greedy forwarding strategy. In turn, the need to apply the planner algorithm is increased, which results in the formation of hot-spots.

The solution proposed in [76] is similar to the one in [70] but provides better locality awareness. While GHT builds its dissemination structure on top of the GPSR [67] routing protocol, it builds its dissemination structure on the Glider [77] protocol. Glider builds Voronoi complex cells with a landmark associated with each cell: sensor nodes are assigned to tiles by their graph distance to a set of landmarks. A distributed hash table is built from the combined tile adjacency graph. While a hash function in GHT takes the data as input and provides a node location or ID as an output, the function in [76] provides a landmark node ID as an output.

In DIM [72] and KDDSC [73], the mapping is done using a K-D tree. Generally, the use of hashing techniques requires global coordination and makes the system insufficiently flexible to cope with the dynamic nature of sensor networks. If sensor locations are not uniformly distributed in the coverage area, or if sensor readings are not uniform over the range of possible reading values, the performance of these techniques is greatly affected. According to [17], in an irregular sensor field, a standard geometric quad-tree in DIM and KDDSC may become unbalanced. This results in an imperfect partition hierarchy that subsequently affects the performance. Also Irregularities result in Voronoi cells that have vastly varying areas, which results in degradation of the performance of the protocol in [76].

PathDCS [75] exhibits another form of data mapping. Spanning tree paths are constructed through the network and data is mapped onto these paths. Some nodes work as beacons and become roots for the constructed paths. The paths are divided into segments that define a certain number of hops towards a particular network beacon. Data is then stored at the end point of the concatenated path segments. Mapping data storage locations onto terminating nodes of existing paths eliminates the need to know the network boundaries as this ensures that a node always exists at that location. Topological changes increase the energy consumption through the network. The overhead becomes noticeable with nodes failure as nodes are trying to reconstruct the tree paths.

Models using content-independent rendezvous regions such as [70] and [74] are known to show more flexibility over content-based models due to the lack of the mapping function (usually in terms of hashing/indexing) that requires global coordination and maintenance. In addition, the dynamic way in which their dissemination structures are constructed contributes to their great flexibility. Even so, topology irregularities pose a great challenge. Rumor Routing [70] is a pioneering content-independent hybrid scheme. It seeks to propagate both events and queries randomly through the network. The propagation is controlled by a time to live (TTL) parameter, with no guarantee that query paths will meet event paths. Events and queries are assumed to happen simultaneously, so the problem of inter-query events-propagation is not explored.

Authors in [74] have introduced a combing approach in which queries are propagated horizontally while events are propagated vertically. The protocol introduces two strategies: combing and reverse combing. The ratio of the event arrival frequency E to the query arrival frequency Q is used to determine the length of the needle (event propagation) and the inter-spike of the comb (query propagation). If $Q < E$, ordinary combing strategy implies that the comb carries the query and the needle implements the travel of the events. If $Q > E$, reverse combing is used, with needles implementing the queries while events travel in the comb. Although the protocol achieves an asymptotic communication cost of $O(\sqrt{n})$ similar to GHT, its efficiency relies heavily on knowing Q and E in advance. In addition, irregular network topologies could result in performance degradation of the CN. In such topologies, combs are no longer well-structured and are incomplete. Events that propagate in needle structures cannot meet the comb-teeth.

Recently, other content-independent rendezvous-based data dissemination schemes have been developed as alternatives to the existing solutions. Among these are the LBDD [78] and Railroad [79]. In [78], a vertical line with width w is defined to act as a rendezvous region. The line is placed in the centre of the network and divides the network into two parts. In [79], a virtual infrastructure called rail is placed in the middle area of the network. Sources transmit the Meta data of the discovered events to the closest node within the rail. Sinks transmit their quires to the rail where they travel to reach the node with the interested data. The limitations described in section 2 apply to these approaches.

2.3.4.1 Location-Service Rendezvous Protocols

Many Location-service protocols apply the concept of rendezvous to location-updates and location search queries. A survey on location service protocols and their different classifications appears in [80]. Many of these schemes have been developed for MANET. However, approaches such as quorum–based [81], [82], [83] and grid-based quorum approaches [84], [85] have been developed for WSN.

Grid-based schemes build the dissemination structure, “the grid,” using geographic routing such as GPSR. Some of these schemes decouple the function of location service from the routing function [84],[85]. Geographic routing is then used for data routing to destinations with known locations. Decoupling routing functions from location service functions is believed to provide performance enhancement. This is different from the approach taken by other grid-based schemes such as TTDD [18], which utilizes the existing grids for both data query and retrieval. This makes the delivery paths longer, especially when the source and destination are close to each other. However, grid-based schemes have some drawbacks. As the grid could eventually disconnect the network, the performance of the protocol is degraded once the network is disconnected. Also, extra overhead is required for the purpose of grid maintenance. Moreover, grid schemes rely on a static parameter known as (cell size a) to construct the virtual grid. This parameter should be carefully selected according to the network size and node density. Both small and large values of this parameter are shown to reflect some problems on the performance of the grid-based schemes [81].

Variants of quorum-based schemes have been introduced [81], [83], [86] and a comparison with the grid based quorum is given in [80]. Some of the proposed strategies to build the quorum require the data (location of the mobile sinks) to travel on orthogonal directions; others require data to continue traveling over the perimeter of the network and voiding era (in addition to the orthogonal directions). The variant where location updates travel the perimeter are shown to be more efficient in their success rate and hence their reliability as the intersection at the perimeter is a guarantee. However, they require extra cost as they mainly use face routing [87] to travel the perimeter. Face routing requires the network to be a planner graph (e.g. Gabriel graph). Heavy beaconing is a vital requirement to cope with topological changes and allows planarization information locally stored at nodes to be updated, which adds to the overall cost.

From the point of view of location service, the above approaches are complementary to our approach. Their main objective is to enable geo-routing (location based routing) as the source nodes need to be aware of the location of the destination nodes. They are best used in scenarios where destination nodes move uncontrolled (such as in the case of MANET and in WSN with mobile sinks that apply random mobility). The movement of the sink itself is thought of as an "event" that the source nodes should be aware of. These approaches are less efficient in scenarios where sinks are static or sinks mobility is controlled. In the first case, there is no need for location updates/retrieval. In the second case, Mobility trajectory is controlled and guided by an application goal and thus is no longer random.

Looking at these approaches as integrated data dissemination protocols (location service + geo-routing), we would argue that these approaches belong to the "push" data dissemination model. Once the mobile sink location is acquired, sources utilize the location information and transmit their data directly to the sink through location-based routing (e.g. GPSR). Communication traffic is no longer on-demand. When the network exhibits a large number of sources/events the significant overhead overwhelms the constrained resources of WSN, especially the constrained power supply.

To summarize, most of the rendezvous data dissemination approaches cannot cope with the shape irregularities of the underlying network topology. They build the dissemination structures upon global topological knowledge. In most cases, cooperation of the existing multiple sinks is not considered. Moreover, the independent actions of the underlying components (the push and the pull) leverage resource wasting. Our proposed work aims at developing an efficient hybrid scheme that takes the above drawbacks into consideration. We consider integrating a topology detection process into our proposed scheme so that the proposed protocol is topology adaptive. While data dissemination at the upper layer is greatly affected by the underlying protocols, most academic work considers topology detection and data dissemination separately. The work in this thesis shows that, by integrating them, we can provide a robust localized data dissemination scheme. We opt for modifying and integrating the BoundHole algorithm [88],[89] into our protocol as it has proven to be a lightweight, localized algorithm that scales well and relies only on one-hop neighbor information. This algorithm was originally developed to extract network topology and detect existing holes. Many other algorithms have been developed to achieve the same goal [90], [91]. We believe that all these protocols could be alternatively used.

As in all hybrid schemes, our proposed protocol has two phases: the pull phase and the push phase. The proposed protocol synchronizes these phases to handle the uncontrolled events propagation problem, resulting in totally on-demand data dissemination. In such a way the protocol does not exhibit any resource wasting.

The proposed protocol is a lightweight protocol that scales well as it depends only on localized information. It does not need a specific routing strategy. Both query dissemination and event propagation phases adapt to topological updates in an efficient localized manner.

To the best of our knowledge, our protocol is the first to introduce the concept of hybrid data dissemination for tiered architecture with irregular topology. These unique features of the proposed solution dramatically reduce the transmission costs and result in efficient operations in large-scale sensor networks.

2.3 Mobility in Sensor Network

In early work, mobility in WSN was considered an incidental side effect that should be mitigated. Mobility could result in frequent topology changes, high packet loss rate, and poor energy efficiency [92]. The work described in [18] and [69] represents this point of view. Recently, researchers show that mobility could be a desired property [93]. In both cases, mobility has a significant impact on the dynamics of the WSN. It increases the complexity of the design of networking protocols and the distributed algorithms.

Mobility in sensor networks is different from its counterpart in systems like MANET [94]. In WSN mobility is usually termed as *controlled mobility* [95]. This notation refers to the fact that mobility in WSN is guided by an upper level goal. The node could be attached to or carried by a mobile entity. The movement is controlled either directly by online commands from the monitoring site or by pre-programmed algorithms [7].

The degree of mobility in WSN could be varied. It is possible that all the nodes have movement capabilities. In this case, the network is referred to as a *mobile sensor network.* In other scenarios only a few nodes have such capability and usually these nodes work as *mobile relays or sinks* [96]. For each of these scenarios, the main concern is the movement pattern or trajectory that describes the movement scenario. In particular, how to arrive at a movement pattern that allows for the achievement of the upper goal? Another concern is how to coordinate several mobile nodes in a way that boosts the performance of the network operations? Answers to these questions are always protocol specific. As in all aspects of the sensor network, mobility cannot be considered separately. It should be studied in the context of some other protocols.

As mentioned previously, sensor nodes have a constrained power supply. These nodes are required to work unattended and unwired depending mainly on their power supply with no chance for replenishment. The network lifetime is associated with the nodes lifetime, which should be efficiently utilized. For many years, developing energy-efficient protocols was the only solution that could participate in the efficient utilization of sensor nodes power capabilities. Mobility is first believed to add complexity and hinder the achievement of efficient communication protocols. Most of the research work regarding mobility is concerned with the mitigation of the negative effects that could be introduced by mobile nodes.

Literature then shows that mobility could be incorporated into the WSN with the aim to increase the coverage degree, teach static sensor nodes their positional information, assist in deploying static sensor nodes, and enhance network performance. In fact, all of these goals inherently work towards increasing the network capacity and enhancing its performance.

Recently, much work considers mobility as increasing the coverage ratio [97],[98], [99],[100]. Coverage refers to the overall area that the sensor network can monitor. These approaches target the mobile sensor networks where all sensor nodes are assumed to have a movement facility that can be controlled to allow nodes relocation. The idea is based on deployment then relocation. Moderate numbers of redundant sensor nodes are first deployed then relocated to fill the position of the failed nodes.

Mobility has also been utilized to perform the deployment task [101-105]. For example, in [103] three distributed protocols -- VEC (VECtorbased), VOR (VORonoi-based), and Minimax -- are introduced for mobile sensor networks. The main principle is that moving sensors from densely deployed areas to sparsely deployed areas provide high coverage. The protocols run iteratively until they terminate or reach the specified maximum round. In each round, sensors first flood their locations and construct their local Voronoi polygons (VP). These VPs are used to discover the coverage holes. Then one of the three proposed protocols can be adapted to calculate the target positions of the sensors and determine where they should move. The VEC pushes sensors away from a densely covered area. The VOR pulls sensors to the sparsely covered area. The Minimax moves sensors to their local center area.

Enhancing network connectivity is one of the issues that mobility can leverage [106],[107],[108],[109]. While the work in [109] does not target wireless sensor networks, it shares many concepts that are worth exploiting in WSN. The authors consider a network of robots. They assume that every robot can reach every other robot over at least 2 vertex-disjoint paths. This is known as a 2-connected network. They address the problem of retaining this level of connectivity under link or node failure. The solution is based on the relocation of some robots. They provide an algorithm that is based on graph theory to identify cut vertices. Nodes that have a degree of 1 would move towards a cut vertex and establish new links. Such links provide alternatives to the critical links between cut vertices. Another algorithm is also provided and called contraction. The robots move inward to the centroid of the deployed robots in order to increase the connectivity.

Exploiting mobility to extend network lifetime has been introduced recently into wireless sensor networks. More and more work has been devoted to addressing the effect of sink mobility on enhancing the performance of the network. Two main regimes of sink mobility are considered [96]. The first one is referred to as *data mules* where mobile sinks move to visit sensor nodes and pick up the data over a single hop wireless transmission.

The main philosophy of the data mule regime is that nodes are not required to forward the data on behalf of others. They only transfer their own data on a single hop, which significantly reduces the energy consumption. The other regime of sink mobility considers the mobility of the base station as an assisted facility to improve the network performance. This kind of movement is referred to as *sink repositioning* as a mobile sink changes its position within the field and data are forwarded to it at the new position. In such assisted mobility protocols, the main regime of data transfer is the wireless multi-hop transmissions; mobility helps reduce hot spots and increases the load balancing through the nodes. These approaches are always associated with data collection mechanisms that maximize the benefits of the mobile sinks.

2.3.1 Sinks as Data Mules

The main concern in the data mule regime is the latency problem [110]. This problem refers to the delay exhibited in the data transfer process as nodes have to wait to be visited by the mobile sink. A mobile sink could move randomly within the field, according to a known pattern, or could be actively controlled in real time [7]. Wireless data transmission speed is faster than the mechanical speed of the moving sink. This difference in speed makes these approaches suitable only for applications that can sacrifice latency. Another problem exhibited when the data mule mechanism is considered for data collection is the buffering problem. As nodes wait to be visited by the mule, the buffer, which is usually size-constrained, could be exposed to overflow. This overflow increases the data loss rate and adds to the drawbacks of this mechanism.

In [111], authors propose the idea of moving the data collector/sink towards the source node with the highest traffic. Such a mechanism enhances the performance for high traffic paths. The mechanism requires mentoring the network operation and determining the best location of the sink. The authors assume that the data collector knows the positions of all the deployed nodes. They also assume that the nodes are within the communication range of the data collector and the radio's transmission power is assumed to be programmable for the required range. The main limitation is that the performance could be worsened for paths with lower traffic and for paths topologically opposite to the movement direction of the data collector. To solve these problems, the authors opt for validating the overall impact of the move on the transmission energy before movement is confirmed.

Data Mule mechanisms require extensive data buffering at each source node. The authors in [25] investigate the relationship of the data transfer rate and the buffer size at each node. Also, the effect of changing the number of mules and access points is studied. The random walk model is assumed for the data mules.

In [112], vehicles that pass near sensor nodes are utilized in data collection. Data transmission is initiated by sensor nodes that are assumed to know the trajectory of the mobile vehicles. The time of transmission is determined or predicted according to knowledge of the trajectory. Sensor nodes stay in sleep mode until wakened up for transmission at the estimated time. The data collection process is modeled by a queuing model. Success rate and energy consumption are used as suggested metrics.

To solve the problem of buffer overflow that could be exhibited when the data mule collects the data, authors in [113] suggests a heuristic solution that allows the data mule to first visit the nodes that are expected to have buffer overflow. The main objective is to allow the data mule to visit sensor nodes in a way that prevents the overflow. A solution based on scheduling the mobile data mule in real time is suggested. Authors provide a heuristic solution called Earliest Deadline First (EDF). In this solution, the data mule first visits the node that has the earliest buffer overflow deadline. This solution tends to increase the data loss rate as nodes with consecutive deadlines are suffering from buffer overflow before being visited by the data mule. This is due to the fact that such nodes may be located far from each other. To enhance the behaviour of the EDF algorithm, the authors present a variation that includes buffer overflow deadlines and the distances between nodes in determining the visiting schedule. However, the problem of back-and-forth movements between far away nodes cannot be avoided.

A best-effort based solution is proposed in [114] for data collection. A mobile sink travels guided by a linear path. When it comes within the transmission range of the nodes in the vicinity of that path it collects their data. Sensor nodes that will never be in the vicinity of the travel path of the mobile sink transmit their data to nodes that are expected to be visited by the sink using multi-hop wireless transmission. Nodes determine the intermediate nodes to transmit their data at an initialization phase by building a shortest path tree. The data travels over the shortest path before being buffered at the intermediate node, which will be visited by the mobile sink. An extension to this algorithm is suggested in [115] to consider multiple mobile sinks to increase the scalability in terms of deployment area. The multiple sinks travel in parallel linear paths and divide the workload among them.

Best effort adaptive algorithms that adjust the speed of MDCs while collecting the data are also proposed in [114]. The authors introduce algorithms to allow the mobile data collector to change the speed of movement based on the nodes existence in its transmission range. To increase the amount of data transfer, an MDC should slow down or stop when there are nodes in the transmission range. When there is no node in range, the MDC could travel faster. The algorithms are a best-effort based service where there is no guarantee that data would not get lost during the transfer.

In [116], a PBS solution is introduced with the main objective to ensure that data collection occurs before buffer overflows. The solution computes the minimum speed and the trajectory of the ME that guarantee no buffer overflow occurs. To achieve this goal, the PBS calculates a mobile sink trajectory path so that no overflow occurs in between every two consecutive visits to a node. The PBS algorithm computes such trajectories based on knowledge of the data generation rate of sensors and their locations. The algorithm assigns a buffer overflow time to each node and follows two phases: *partitioning phase and scheduling phase.* In the *partitioning phase*, nodes that have similar buffer overflow times and are closely located form a group. In the *scheduling phase*, the trajectory segments (inside groups) are determined and concatenated to form a complete trajectory. The complete trajectory length determines the minimum speed of ME.

2.3.2 Sink Repositioning

While latency is the main concern in data mules based approaches, the main concern in the sink repositioning based schemes is the movement pattern or trajectory that describes the movement scenario of the mobile sink. Latency in this case is comparable to that of fixed sink scenario. In particular, the main problem is how to come up with a movement pattern that contributes to boosting the performance of the network operations? When multiple mobile sinks are involved in the deployment scenario, coordination among them becomes an essential component of the communication algorithm. Coordination among several mobile nodes in a way that keeps the positive effect of the mobility is a challenging task and is still an open problem. Usually the problem is ignored by considering only the case of a single mobile sink [117]. As a sub-problem of the coordination among multiple mobile sinks, the authors in [117] discuss how to lessen the trajectories interference of multiple-mobile sinks. They provide multiple algorithms to solve this problem.

Sink repositioning is proposed in [118] by aiming to balance the energy consumption among nodes. The authors suggest that the mobile sinks change their positions along the network peripheral. Time is divided into rounds, sinks stay stationary during the rounds, at the end of each round the new positions of the sinks are computed and the sinks move towards these positions. Inductive logic programming methods are used to determine the new positions of the mobile sinks based on minimizing two objective functions. The first function is the total energy consumption of all sensors. The second is the maximum energy consumption of any node in the next round of sink repositioning. The simulation results show a longer lifetime (defined as the time until the first node dies).

Exploiting sink mobility to prolong network lifetime is explored in [119]. The authors consider the case of a grid sensor network where sensor nodes are placed at the grid points. They provide linear programming (LP) solutions to maximize the lifetime by calculating the sojourn times of the mobile sinks. The assumption is that mobile sinks should be placed at one of the grid points and that delay results from the movement are negligible. The authors observed the impact of the shape of the deployment area on determining the sojourn times of the mobile sinks.

The most relevant work for our purposes is that suggested in [120]. The authors study the load balancing among nodes within the network. They analytically prove that the optimal mobility trajectory, in the case of circular networks, occurs when the mobile sink moves around the network peripheral. They suggest a joint mobility and routing solution to extend the network lifetime. In their proposal, a mobile sink constructs a global routing tree based on the shortest path while moving. The peripheral nodes forward their data using a trajectory forwarding routing that follows the mobility trajectory (in that case, a circle). Nodes outside the trajectory reach the mobile sink by using paths that consist of circular arcs followed by straight lines directed toward the trajectory center. The solution results in considerable energy savings. However, the scheme tends to be less adaptable to the topology irregularities. Both the shortest path tree and the trajectory forwarding are proactive mechanisms that require extra communication overhead to keep their performance high.

Joint mobility strategy is also suggested in [9] where a routing scheme is proposed to route the data to a single mobile sink. The sink moves around the network peripheral. The authors propose that each data collection round is preceded by a sampling round. In a sampling round, the sink stops at certain anchor positions and determines the optimal visiting time by sampling the global power consumption. The visiting time is then applied to control the movement of the sink while in the data collection round.

Both [9] and [120] consider a single mobile sink. As mentioned above, multiple-mobile sinks raise the fact that coordination is required to keep the "positive effect" of the mobility on the network performance. Both [9] and [120] did not consider this case.

In the context of enhancing the sensor network lifetime, a dynamic multi-node positioning is proposed in [121]. A clustering mechanism is used where cluster membership is based on proximity of CHs nodes to the individual sensor. The movement of CHs is restricted to maintain the connectivity of the inter-CH network. The movement of such a CH could partition the network. So CH first checks the links with its neighbours before the movement. If changing the position of a CH will partition the network, its neighbours are to move to restore the broken links. A CH will cease all motion until a token is granted. To prevent simultaneous relocations, a mutual-exclusion based mechanism is used.

The approach in [122] also addresses the same problem: maintain the connectivity among CH nodes when some of them move. While [121] provides a centralized based solution, the authors in [122] provide a distributed mutual exclusion based algorithm. The coordination among immediate neighbours is based on a predefined priority system. The assumption is that nodes have distinct priorities. The priority can be static (e.g. ID-based) or dynamic.

In this thesis, we introduce a solution that considers the scenario of multi-mobile sinks. In Chapter 3, we describe a simple and non-interfereable mobility pattern that inherently solves the coordination problem. The proposed protocol efficiently adapts to the topological changes as it does not require any global knowledge that should be collected in a centralized manner. It also works for any network topology whether regular or irregular.

2.4 Communication Infrastructure in Sensor Network

In WSN, a large quantity of sensor nodes are deployed (in most cases randomly) in the area of interest. This deployment results in an arbitrary network graph that is referred to as a network topology [19]. The communication over such a graph is multiple hops. This means that nodes can communicate directly only with those in their vicinity (transmission range).

Nodes communicate with others outside of their transmission range over multiple hops communication. This kind of communication implies that each node could play the role of a router to forward the messages over the multiple hops on behalf of other nodes. The arbitrary topology graph is structured by the communication algorithm [123]. For example, spanning tree based algorithms opt for building the tree structure over the arbitrary graph. Then the communication process takes place using the tree primitives.

One of the research concerns that is not fully explored in the literature is the degree of independency between various network protocols. To date, little work has been done towards unifying the design of WSN protocols. WSN is application-specific in the first place. It poses many challenges that motivate the production of hundreds of protocols at each networking level. The different hardware characteristics and the different requirements of the upper layer applications boost the protocol productivity of WSN [23].

As has been mentioned previously, numerous protocols that target aggregation, routing, dissemination, medium access and topology control have been developed. The main notice regarding the wide scope of WSN protocol development is that the performance of a given protocol is tied to the underlying assumptions about the rest of the system. When such assumptions are varied, degradation in performance is noticed. The variety of possible assumptions about the system decreases the reusability of the developed protocols.

A new research direction towards a unifying WSN software architecture that increases the modularity and reusability of the designed protocols is established in [124], [125]. The argument against this direction is that the unifying process, which aims at the clear cut between various levels of networking, would significantly increase the overhead and affect the resulting performance [20]. Most of the current proposed protocols merge functions from different networking levels.

In [124], the authors discuss the narrow waist architecture where sensor protocol (SP) resides between the network layer and the data link layer. They describe the rules by which the network services could be arranged over the layered architecture. They also discuss the neighbour's management issue.

Leveraging the SNA in [124], the authors in [125] suggest a modular network-layer for sensor networks that sits atop SP. Their main concern is to ease the implementation of new protocols by increasing code reuse and runtime sharing. Code reuse provides a rapid protocol and application development. On the other hand, run-time sharing reduces code and resources consumed. The authors discuss the trade-off between functionality decomposition and complexity. They find that finding the right granularity at which to break up the functionality at the network layer is challenging. Unnecessary runtime overhead could result from a very fine-grained decomposition while a too coarse decomposition reduces the level of sharing, which in turn increases the reimplementation.

The authors in [126] propose Tenet architecture, which is complementary to the narrow waist architecture in [124]. Tenet architecture does not address the modularity of the software. It restricts the placement of the application functionality in a multi-tire system. In Tenant, the sensor level tier can be implemented on SP. Tenet shares some similarities with the Internet's end-to-end principle [127], yet it is based on specific tiered network technology.

The authors in [124] are trying to achieve complete standard software stack architecture for sensor networks. This is still far away from being a reality. There is a conviction in the literature that this unifying process would significantly increase the overhead and affect the resulting performance [20]. It is still unknown whether this huge unifying process will provide the required benefits to the protocol designers.

Building a generic infrastructure at the level of physical links is a promising step toward increasing the reusability of upper layer protocols. To build such infrastructure over sensor networks, the literature explores two approaches. One of them is to construct the infrastructure that supports specific processes, such as routing and data aggregation. This model is usually optimized to efficiently achieve an upper goal such as minimizing congestion. Clustering and tree-based approaches are the most utilized techniques in building such infrastructure [21]. They provide a way for nodes to self organize with the main goal of maximizing the performance of specific upper layer operations.

The other way is to build infrastructure that is not tied to the upper protocol. This is known to be general purpose infrastructure. In this case, the resulting infrastructure should be able to support different upper layer processes with equal efficiency [24]. This task is considered a challenging one in networks that have scarce resources, such as sensor networks.

In this thesis, we envision mobility as an assisted-facility that could provide a great flexibility to static sensor networks. In scenarios where all nodes are static, with no ability to move, nodes build logical infrastructure by self-organizing. Clustering and spanning trees are forms of self-organization approaches that result in unstructured overlays [21]. The operations over such overlays are usually based on the flooding mechanism. Failure handling and maintenance require cascade updates through the network. In addition, the optimization of such approaches is usually towards a specific process, such as routing.

Sensor nodes have scarce resources in terms of energy, bandwidth and communication. This constrained environment makes logical overlays such as DHTs not suitable. The main reason is the belief that DHTs overlays would produce extra overhead compared to the benefits they provide to the upper layer applications [22]. When mobility is considered, the movement of the nodes may quickly change the topology. This results in the increase of the overhead messages for topology maintenance and movement management.

Building generic infrastructure over sensor networks is studied in [128]. The authors develop a virtual infrastructure in terms of coronas and wedges. They consider the case of a static sensor network where all nodes are static. The sink named as Training Agent (TA) is considered at the centre of the network. The assumption is that the TA has a multiple levels transmission range. The TA takes the burden of training the nodes to gather knowledge about their position with respect to the centre. This position will be represented by the wedge, and the corona where the node is located. However, the protocol is centralized and is based on global information. The number of coronas to be created should be known to the TA before it creates them. In addition, the mechanism requires synchronization of the wakeup time of all sensor nodes within the network, and the level of the transmission range of the sink at that time.

A multi-scale communication overlay is developed in [26] to support upper layer protocols. The protocol belongs to the clustering based approaches. The nodes are organized into cells, super-cells and so on. A self-election mechanism based on sending periodic beacons is used to form the hierarchical overlay. As in most clustering based approaches, maintaining the whole structure requires topological updates to be broadcasted to all nodes and re-clustering is to be performed to adapt to the changes. This introduces extra overhead that could participate in draining the resources of sensor nodes.

An attempt to fill in the space between logical and physical infrastructure is proposed in [129]. Authors proposed a Virtual Ring Routing (VRR) protocol where logical rings are constructed over the link layer. The protocol is inspired by DHT mechanisms and provides both point-to-point and DHT like operations. The protocol creates logical rings that do not keep the node proximity. Moreover, all the nodes within the network should have unique logical addresses (identifiers) that are globally ordered. In addition, the protocol is optimized only for routing processes.

Abstract regions [130] and [131] are two approaches described in the literature to abstract the underlying communication primitives. They provide a way to ease the development of upper layer applications.

Our work in Chapter 5 shows that augmenting SSN with a few powerful mobile nodes would simplify organizing a generic network infrastructure that could be leveraged by multiple upper level protocols with efficiency and robustness. We focus on the deployment scenarios that involve both mobile and static nodes. We consider the case where most nodes are static and only a few powerful nodes are mobile. We present an attempt to build a structure that fills in the space between the physical and logical structures. This kind of overlaying was difficult in networks such as the sensor network. Our philosophy is that the few powerful mobile sinks/robots could take the burden of being the network organizers in addition to being the data collectors and management infusers. We also suggest that the interface to the application layer would be through a communication plan. This plan could be seen as a complementary to the abstraction given by Abstract region [130] and that in [131].

2.5 Summary

The existence of many tradeoffs in the design of sensor network protocols makes room for further investigation and makes the sensor network a continuous hot research area. Tradeoffs such as data accuracy and freshness versus energy savings, efficiency versus complexity, latency versus energy consumption, and reliability versus energy savings must be well investigated.

In wireless sensor networks, communication links are unreliable and the connectivity is maintained through the working algorithms. One of the most important research issues is the consideration of the system reliability.

In sensor networks, nodes do not require sophisticated services from each other. The flow of data is mostly toward specific hosts. Simple but efficient data gathering protocols are important requirements for the sensor network. Different protocols involve different data delivery models that restrict their suitability to multiple upper layer applications. Integration of multiple protocols could be one solution in achieving the wide applicability of these protocols.

The evolution in sensor network hardware technology enables both mobility and multiple radios to be incorporated and leads to heterogeneous sensor networks. Communication protocols that go along with these new trends have a significant impact on the success of sensor network applications.

Mobility has a great impact on designing protocols for sensor networks. It could provide greater flexibility to the old static scenario, but it increases the complexity of designing the communication protocols. Many benefits have been already investigated and new benefits are yet to come.

Decoupling the communication protocols from the application semantics to increase the modularity and reusability is still an open research direction that should be carefully studied. Given that the sensor network is mainly application oriented, the benefits of the clear cut between various networking aspects over the current cross-layer approaches require more research efforts to be clearly identified.

Chapter 3

Prolonging Lifetime in Sensor Networks

This chapter describes our proposed solution for prolonging sensor network lifetime. The chapter is organized as follows: Section 3.1 introduces the underlying problem; Section 3.2 explains the impact of enabling sinks mobility and provides an overview of the proposed solution; Section 3.3 explains our assumptions about the underlying network model and suggests a deployment model as part of our suggested solution; Section 3.4 discusses the energy expenditure expected by the proposed protocol and the requirement of an associated novel data forwarding scheme; Section 3.5 describes the proposed mobility strategy and the design details of the proposed boundary-peeling data collection protocol; Section 3.6 summarizes the performance of the approach; and Sections 3.7 and 3.8 offer a discussion and a brief summary.

3.1 Introduction

Enhancing sensor network lifetime is an important research topic for wireless sensor networks. As mentioned previously, sensor nodes have a constrained power supply. These nodes are required to work unattended and unwired depending mainly on their power supply with no chance for replenishment. The network lifetime is associated with the nodes lifetime, which should be efficiently utilized. Maximizing the lifetime of a sensor network could be the ultimate goal that will convert the theoretical achievements in sensor network research to successful real world deployed networks.

The main threat to the lifetime of sensor networks is the energy hole that can be formed around the sink or the data collector [1]. The energy hole tends to be unavoidable in networks depending on stationary-based sinks. This is definitely true for networks with uniform node distribution and uniform data reporting [2].

Extensive work has been proposed by the literature to enhance the sensor network lifetime using different conceptual approaches. Routing mechanisms that aim at balancing energy consumption and load through the network have gained considerable attention. Some of these protocols consider mobile sinks. However, their main concern is how to route toward the mobile sink, rather than how to exploit the mobility of sinks to enhance the lifetime of the network. A good survey of such mechanisms is [132].

The authors in [2] consider the guidelines study of energy balancing, which leads to enhancing the lifetime in sensor networks that include static sinks. The authors conclude that the optimal energy balancing is impossible under some conditions. They also suggest a solution based on training the network into coronas with equal widths. The widths of such coronas should be determined based on the energy loss factor. The model described above is used in [6]. The authors show that a suboptimal energy balancing could be achieved. This could be done by populating the sensors over the coronas era using a controlled non-uniform node distribution. The authors describe a rule for such controlled distribution that leads to enhancing the network lifetime.

Many approximation algorithms have been developed for maximizing the lifetime of the network. Most of these algorithms are interested in the behaviour of sensor nodes rather than the sink node. For instance, the effect of node density on network lifetime is studied in [133]. The authors conclude that the number of sensors should be selected carefully as the network lifetime does not grow proportionally to the increased node population. One of the approaches that consider the behaviour of the sink node is found in [134]. The authors determine the location of the base station node (the data collector) that could result in maximizing the lifetime. The algorithm considers the routing strategy that would be used but no mobility is considered.

Exploiting mobility to extend network lifetime has been introduced recently into wireless sensor networks. More and more work has been devoted to addressing the effect of sink mobility on enhancing the performance of the network.

Mobility in sensor networks is different from its counterpart in other networks, such as MANET. In a sensor network context, mobility is usually controlled. The movement of the mobile sinks and/or sensor nodes is guided by a certain goal. In some proposals [135, 136], mobile sinks would be used as transportation units where they move to collect the data from the stationary nodes. Such approaches result in significant energy saving but the latency problem is the main concern. Other approaches consider the mobility of the base station as an assisted facility to improve the network performance [9]. In such assisted mobility protocols, the main regime of data transfer is the wireless multi-hop transmissions; mobility helps to reduce hot spots and increases the load balancing through the nodes. So these approaches are always associated with data collection mechanisms that maximize the benefits of the mobile sinks.

Our work belongs to the assisted mobility protocols. We envision a solution that exploits sinks mobility and multiplicity to maximize the lifetime of sensor networks. On the one hand, sink multiplicity reduces the single point of failures and balances the load distribution. On the other hand, sink mobility could reduce hotspots and balance the energy consumption, which, in turn, contributes to enhancing the lifetime. However, sink multiplicity and mobility complicate the design of the networking protocols.

3.2 Mobility Enabled Multiple-sinks

The degree of mobility in WSN could be varied. It is possible that all the nodes have movement capabilities. In this case, the network is referred to as *mobile sensor network.* In other scenarios, only a few nodes have such capability and these nodes usually work as *mobile relays or sinks.* For each of these scenarios, the main concern is the movement pattern or the trajectory that describes the movement scenario. In particular, how to come up with a movement pattern that allows for achieving the upper goal? Another question is how to coordinate among several mobile nodes in a way that boosts the performance of the network operations? Answers to these questions are always protocol specific. As in all aspects of the sensor network, mobility cannot be considered separately. It should be studied in the context of some other protocols.

Mobility in the sensor network is usually termed as *controlled mobility.* This notation refers to the fact that mobility in WSN is guided by an upper level goal. The node could be attached to or carried by a mobile entity. The movement is controlled either directly by online commands from the monitoring site or by pre-programmed algorithms.

In this chapter, we introduce a solution based on a deployment strategy of multiple mobile sinks and an associated data forwarding mechanism. The key behind our solution is that it combines the benefits of the existing solution (multiple sinks, clustering, and the non-uniform node distribution with performing neither clustering nor non-uniform node distribution) while avoiding their limitations. We investigate controlled sink mobility to answer the above questions. By exploiting sink mobility and multiplicity, our solution aims at (a) leveraging load balancing, (b) minimizing energy consumption, (c) reducing overhead and complexity, (d) maintaining network connectivity, and (e) effectively adapting to the topological changes.

In our proposal, data traffic is directed away from the network center (where the number of nodes is relatively small) toward the network peripheral (where the number of nodes is intuitively large as we go from a narrow to a wider area) in which mobile sinks would be initially randomly deployed. Data reports are collected at the network perimeter and travel over the network perimeter towards the stationary sinks. Eventually, perimeter nodes at the network peripheral would be exposed to a *peeling phenomenon* (have their energy depleted, exposing other inner nodes to be perimeter), which results in partitioning one or more sinks from their one-hop neighbors. The partitioned sinks move discrete steps following the direction of the progressive peeling (i.e., from outer to inner layers towards the network center) and the connectivity is re-established. The network center is used as a pulling force for mobile sinks at the same time as a disparate force for internal data propagation. It provides a by-product increase in the load balancing through segmentation of the data propagation directions. The protocol is totally dynamic. The overhead of exchanging topology updates messages associated with the mobility is not required. We show that our solution leads to a sub-optimal energy balancing through the network. The balance of both load and energy consumption leads to prolonging the network lifetime. We also investigate a "guard region" concept where a subset of the coverage area at the outer layer of the network could participate in protecting the core area of interest from being peeled-off. This concept raises the notation of the coverage tolerance ratio CTR that depends on the application. The performance of the protocol is shown by intensive simulations that consider realistic conditions of the underlying network settings. Many experiments are conducted to compare the performance of our protocol with the centric-sink model where no mobility is considered, as well as the MobiRoute [9] where sink mobility is considered. The results show the efficiency of the proposed model in expanding the network lifetime, minimizing overhead associated with sink-mobility, and achieving a high degree of reliability.

3.3 Network Model

Network infrastructure: We assume a two-dimensional terrain area. The distribution of the nodes within the terrain area is uniformly random. We assume that this uniformly random distribution results in a connected network topology [137],[138]. Sensor nodes have the same capabilities in communication, computation, and storage. Each sensor node knows the coordinates of its location using either GPS or any existing localized techniques [139],[140]. Two nodes are considered to be neighbors if each one is located in the transmission range of the other.

Sinks or data collectors are special nodes with no restriction in their energy resources or their communications capabilities. We assume multiples of these special nodes are initially placed arbitrarily at the network peripheral. While we suggest such exterior placement of sinks as part of our solution, it could actually be forced by the underlying application requirement. In many situations, these special nodes cannot be internally placed [9], [120]. For example, they cannot be internally placed when they could lose the points of connection to the network by being randomly dropped in a sensing field that might be inhospitable. Or they cannot be internally placed if it is required that these access points be accessible to humans (for maintenance and replacement purposes, etc.) while the sensing field itself remains free of human involvement. This might take place, for example, when the sensing field is dangerous or contaminated, or when the sensing field must remain untouched by humans in order to preserve its accuracy [141],[142]. One possible solution in this case could be to deploy the nodes externally at the edge of the network. In this case, these nodes connect to the sensor nodes over low-powered radio and connect to the end user (if they are not the end points) over long radio.

Energy and traffic models: We assume that all sensors have the same initial energy rate e. Although sensors use their energy while they are transmitting, receiving, and sensing, we assume that the transmission process is the dominant factor of the energy depletion [32]. We use the uniform data reporting model where each sensor i generates data with fixed rate μ_i. Each node consumes a unit of energy to transmit one unit of data. So the number of transmissions performed by each node is a good indicator of its energy consumption rate. No data aggregation is considered. Each node transmits its own data in addition to any data units that it receives to forward on behalf of other nodes. Sensor nodes complete their task when they forward their data to any of the data collector nodes.

Mobility Model: We assume that each sink has a controlled mobility facility [7],[143]. Sinks travel using a discrete mobility pattern M where the movement is performed in steps (only under some conditions as we will explain in the next section).

3.3.1 Balancing the Energy Expenditure

In this section, we show that our protocol leads to a sub-optimal energy balancing through the network as defined in [6]. Authors in [6] consider a circular network model where the network area T is partitioned into coronas with equal widths (fig. 3.1). They prove that, if the sink is located at the center of the network c and the nodes are distributed non-uniformly according to the following rule:

$N_i/N_{i+1} = N_{i+1}/N_{i+2} = \ldots = q$ (where $q > 1$ and N_i is the number of nodes in area A_i known as corona *i),* then a suboptimal energy balancing could be achieved. In other words, if the number of nodes in corona *i,* which is nearer to the sink, is larger than the number of nodes in adjacent corona *i+1* by a factor *q* then a suboptimal energy balancing is achieved. The authors define the suboptimal energy balancing as the ability to balance the energy among all the inner parts of the network except the outmost one.

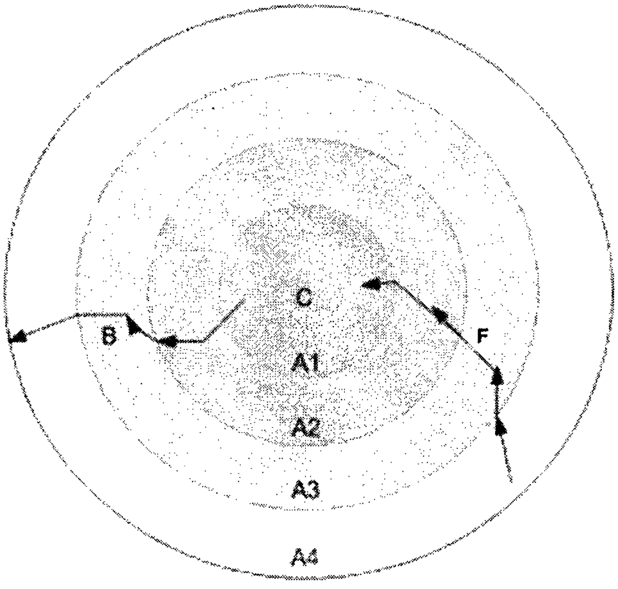


Fig. 3.1 Area – Dissemination direction relationship. The network is partitioned into a number of coronas. Paths B, F represent two directions of traffic. *C* represents the network center.

Distributing a large number of nodes within the coverage area using the above rule could be difficult in real world deployment. Controlled deployment is usually more acceptable if only a few nodes are to be deployed. In this case, sinks are the best candidate for such strategic deployment. We prove that without the need for such controlled sensor nodes distribution, we could achieve a similar sub-optimality. This could be achieved by considering deploying the data collector nodes at the network peripheral. In addition to such exterior deployment, a data dissemination strategy that allows the direction of the traffic to be reversed (from inner parts with relatively smaller number of nodes to the outer parts with larger number of nodes) should be applied.

In figure 3.1, path *F* represents one possible data traffic path from the outer corona to the inner corona where the sink is assumed to be located at the network center. Non-uniform node distribution is represented by darkening the corona (i.e. the darker corona represents the increased number of nodes) so the number of nodes should be increased with geometric proportion from the outer parts to the inner ones.

In our deployment model, no controlled node distribution is performed. Instead, the direction of the traffic is reversed. Considering the same network model in fig 3.1, path *B* represents a possible traffic path. Assuming uniform node density $D= N/A$ where N is the total number of sensor nodes and:

$$D_1=D_2= D_i=\ldots=D_n$$

$$D_i =N_i/A_i \quad , 1\leq \mathrm{i} \leq \mathrm{n} \qquad (1)$$

$$\text{Intuitively } A_1 < A_2<\ldots< A_i \text{ so } N_1 < N_2<\ldots< N_i \qquad (2)$$

$$N_{i+1}/ N_i= N_i / N_{i-1}=\ldots=q \text{ and } q >1 \qquad (3)$$

This means that with uniform node distribution, corona $i+1$ has a number of nodes larger than the adjacent corona *i* (which is nearer to the network center). This is intuitively enforced by the underlying area to be covered by node distribution. Thus, if the traffic is reversed toward the outer corona where data collector nodes would be deployed, then we could get the same benefit without intentionally performing a controlled non-uniform node distribution.

3.3.2 Data Dissemination Strategy

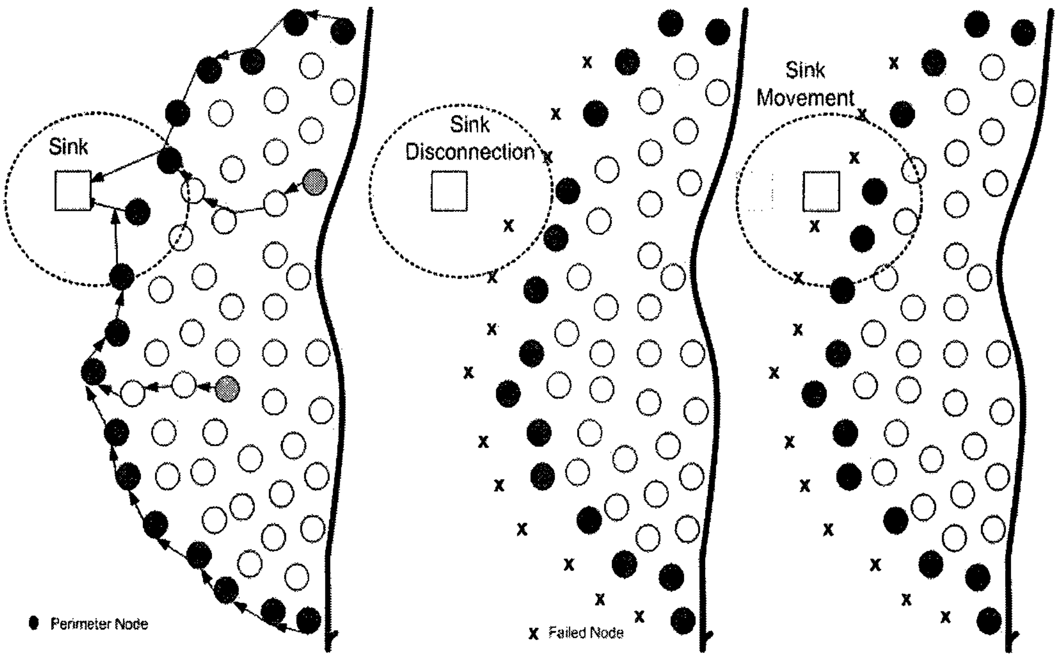
Reversing the direction of the data traffic from inner parts to outer parts and deploying the data collector nodes at the network peripheral are not enough to get the benefit of the sub-optimal balancing of the energy. Without a data dissemination strategy that could take advantage of the previous considerations, the external sink deployment imposes higher communication costs than the centric-sink model [144]. Implicitly, communication paths would be longer in terms of hops count. Nodes around the data collector have to carry the data from a relatively larger number of far nodes. These nodes are exposed to drain their energy resources and die faster than others forming an early partition. So a data dissemination strategy is required to take advantage of the sub-optimal energy balance that could be obtained by reversing the direction of the traffic. In the following subsection, we describe our boundary-peeling data collection protocol that works for both regular (circular, grid) and irregular network topologies. So the network model in fig 3.1 is relaxed to the general network model described in section 3. The protocol aims to prolong the network lifetime defined as the time until a network partition occurs [3]. Partition could be either a sink-network partition (where a sink node loses its connection to its one-hop neighbors), or internal partition where group of nodes form an isolated island.

3.4 Boundary-Peeling Data Collection Protocol

The proposed data collection protocol consists of three phases: initial, continuous, and conditional. Fig 3.2, shows the conceptual steps of the data collection protocol.

3.4.1 Phase I: Topology Recognition

This phase is performed once at the setup time. It consists of three steps. The boundary of the network is recognized at the first step. This step allows each boundary node to know that it is indeed a boundary node. The second step is the computation of the virtual centroid (*v-center*) of the network. This is done offline at one sink (if multiple exist) elected to perform the computation. At the third step the v-center is broadcasted to all nodes. Each node uses the v-center to learn its *allowable forwarding directions* (to be used in phase II). If the center of the network is known in advance, as in the case of the circular network model, it could be used directly (so we use the terms *v-center* and network center interchangeably). In reality, networks are neither circular nor regular. Networks with irregular topologies are an important class of sensor networks that require more attention.



(a) (b) (c)

Fig. 3.2. The boundary-peeling data collection process. (a) The network boundary is recognized and data is directed to the boundary where it travels to the sink. (b) Sink-network partition. (c) Sink movement toward the network center.

We use the *BoundHole* algorithm [88],[89] to define the boundary of the network and to allow perimeter nodes to know that they are located at the network boundary. The algorithm was developed to detect the holes within the sensor field. It inherently identifies the outer boundary of the network. Peripheral sinks cooperatively implement the *Boundhole* at the initial phase. Assuming a total of *m* sinks $S_0, S_1, .., S_{m-1}$, we divide the boundary into *m* boundary segments associated with the *m* sinks. Each segment starts at one sink and ends at the next sink in a counterclockwise direction. So the outer boundary is represented by the following sequence: $S_0, p_1, p_2 \ldots, p_i, S_1, p_{i+1}, p_{i+2} \ldots, p_k, S_2, p_{k+1} \ldots S_{m-1} \ldots p_j, S_0$. where *p's* are the sink-to-sink nodes located at the outer boundary. Each perimeter node *p* keeps information about its upstream and downstream p's nodes in addition to counters (registering its hop-count) to each end-sink. A node also creates a pointer to its parent (upstream) node toward the closest end sink.

Perimeter nodes would exhibit the heavier load amongst the network nodes. Frequently, such nodes would lose their energy and die exposing other interior nodes to be perimeter. The algorithm is supported by a local maintenance process (at the level of one-hop neighbors) that allows the replacement of the dead perimeter nodes by fresh ones. This local maintenance process allows dynamic coping with topological changes affecting the outer boundary and capturing the continuous updates.

3.4.1.1 Computation of Virtual Centriod

To compute the *v-center c*, we consider the network as a closed polygon in which a finite set of perimeter nodes makes up a set of virtual vertices, or *v-points*. The *v-center* is defined as the median of this set. The concept of the *v-center* provides only an approximate solution as the network is not really a polygon with known vertices. The *v-points* can be selected using a threshold distance value (*h* number of hops) as following: once the network perimeter is marked, perimeter nodes that satisfy *kh* distance (where k is an integer variable with values 1, 2, 3,.., etc.) to the sink that is elected to perform the computation and report their coordinates. The sink collects the positions of the v-points and determines the coordinates (x_c , y_c) of c as in (4) where A is the virtual network area using the selected v-points [145]. The v-points could also be selected as the ends and middle of each boundary segment. We note that the smaller the value of h, the more precise the *v-center*.

$$x_c = \frac{1}{6A}\sum_{i=0}^{K-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i),\ y_c = \frac{1}{6A}\sum_{i=0}^{K-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \qquad (4)$$

3.4.1.2 Forwarding Directions

The *v-center c* computed at the previous step acts as a center of a disperse force that directs the data away from it toward the network peripheral. Each node learns its allowable forwarding directions (out of four basic directions Up, Down, Left, Right) according to its location with respect to c. For each node, the four basic directions are represented by four abstract reference points on its communication circumference (the circle with the node as the center and the communication radius as the radius). The allowable directions are those that direct the data *away-from* the *v-center*. Fig 3.3, provides an example. If the location of the *v-center c,* with respect to node n_l, is at the *up-right* corner of an imaginary rectangle, n_l excludes both Up and Right directions. n_l considers the remaining directions (Left and Down) as its allowable forwarding directions. The node alternatively selects its current forwarding direction (the direction where it should transmit its data) out of them. The v-center offers by-product load balancing. It intuitively partitions the network with respect to data propagation process.

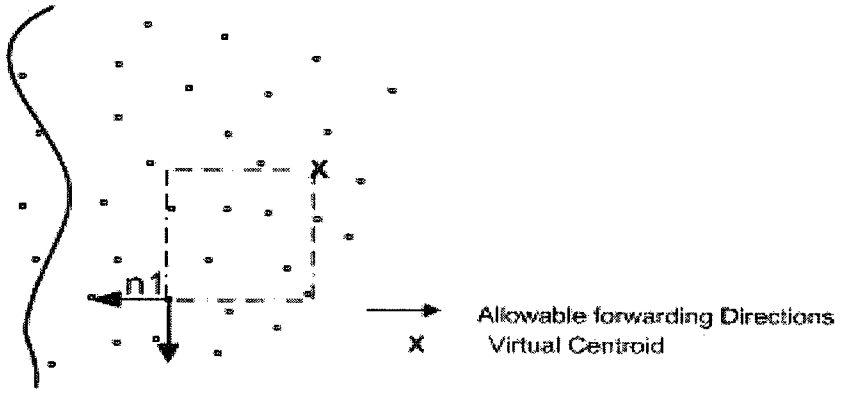


Fig. 3.3. Directions of data flow based on the *away-from-centroid* dissemination strategy

3.4.2 Phase II: Data Forwarding

The data forwarding phase is a continuous phase. Interior (non-perimeter) nodes forward their data toward the network perimeter. The node selects the current direction of data forwarding and transmits the data to the next forwarder in that direction. The next forwarder is the neighbor closest to the reference point of the chosen direction, among those closer to that point than the node itself. To prevent looping back, each intermediate forwarding node should follow the direction chosen at the originating node. The process continues until the data arrives at any perimeter node. Data that arrives at the perimeter region travels the perimeter toward the closest sink.

We should mention that forwarding based on the away-from-centroid rule is loop free. This forwarding strategy is based on geographic greedy routing mechanisms [63]. Such mechanisms select the best neighbor node to be the next forwarder to the destination based on a selection rule that optimizes certain metrics. The main difference is that here there is no specific destination to route for. The global destination is the boundary itself and the boundary exists everywhere. A local destination is a local reference point that represents the selected direction on the circumference of the node. While the direction is fixed, local destination changes at each step. So the data forwarding strategy is locally greedy and ensures that there is progress in each step. Assuming that the v-center is a unique point in the network, this strategy is looping-free. To explain this, in fig 3.4, a node *u* selects the *Left* direction as its forwarding direction. The local destination is a point L_u. Only *v* and *e* are the candidate neighbors to select from. Node *u* selects *v* (the closest to L_u) as the next forwarder and transmits the packet and waits to hear *v* transmission. The node *v* follows the same direction. Its local destination is L_v. Node *v* has only one candidate neighbor *s* that satisfies the selection rule so *u* transmits to *s*. If no neighbor satisfies the selection rule, the node defines the selected direction as an internal stuck direction and sends notification to node *u*. This indicates to *u* that the selected node *v* failed to transmit. *u* should ignore node *v* and retransmit to the next suitable neighbor node *e*. The node *v* should not participate in any transmissions to its stuck direction. If node u failed to find a neighbor that could resume the transmission, no further processing should be done by *u* and the packet is dropped. This ensures one-step backward healing. The first transmission by *u* prevents the originating node (for example node *a*) to explore other neighbors. With a moderate density of nodes, the rate of packet dropping tends to be low. In fact, if the network is covered and connected, then the packet drop rate tends to be zero. In [146] the authors prove that geographic forwarding schemes will always succeed if the network is sensing-covered. The network is described as a sensing-covered network if the communication range R_c of each node is greater than or equal double of its sensing range R_s [146]

The same claim applies to the proposed forwarding strategy, as we will prove in this section. The relationship between the packet loss and the degree of node connectivity, when the assumption of being sensing-covered is not forced, is investigated empirically and depicted in fig 3.17. Moreover, recovery mechanisms such as one-hop or two-hop flooding, as well as face exploring [147], already exist in the literature and could also be combined with our forwarding mechanism.

To prove the correctness of the boundary peeling data collection protocol, we need to prove that if the network is covered and connected, the path to the perimeter will be always discovered. We first introduce the following definitions.

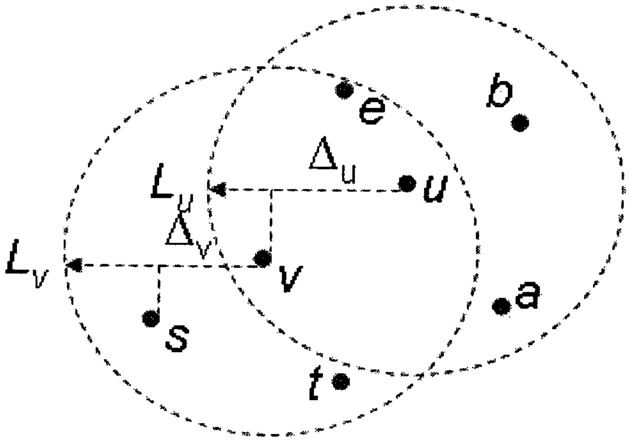


Fig. 3.4 Forwarding Progress using *away-from-centroid* dissemination strategy

Definition 3.1 Away-From-Centroid Forwarding: Let L_u be the local destination point on the circumference of node u that represents u's furthest direction from the center of a given topology. Current node u finds the next forwarder node v, such that the distance $\| (v\ L_u) \|$ is the smallest among all neighbours $\in N_+ (u)$ in a given topology. $N_+ (u)$ is the set of neighbours of u in the forward (positive) direction with respect to the chosen direction. In figure 3.4, $N_+ (u) = \{v, e\}$, $N_+ (v) = \{s\}$, Δ denotes the progress in a given local direction.

Definition 3.2 Stuck-direction is a local direction L where node u does not have neighbours, or where all neighbours in that direction have no neighbours in their L direction.

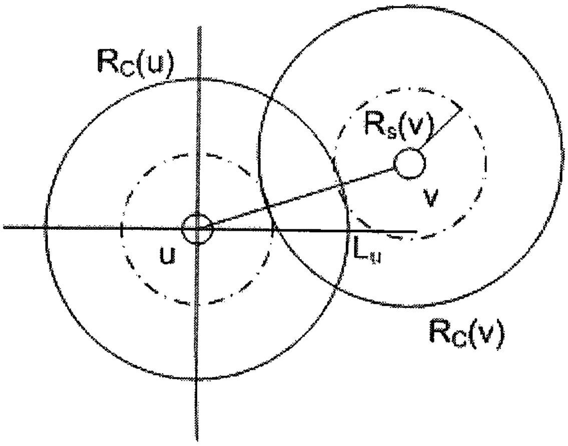


Fig. 3.5 Finding next hop neighbors when the network is sensing-covered. The dashed circle indicates the sensing region of the node.

Lemma 3.1: In the sensing-covered network, a node *u* will always find the next forwarder using the away-from-centroid forwarding strategy.

Proof: We will prove this by contradiction. We will assume the network is sensing-covered and the chosen direction is the right direction represented by L_u in figure 3.5. If the network is sensing-covered, the right half of the communication region of node u must also be sensing-covered (this applies to the other direction of the communication region of u as well). If node v is the nearest node to L_u and $||uv|| > 2R_s(u)$ (meaning that v is not within the communication region of u), then the right half of the communication region of u is not covered. This implies a contradiction and proves that the forwarding strategy will find the next hop.

Lemma 3.2: If node *u* selects node *v* as its next forwarder in direction *L,* so Δ_u, the progress in the direction *L* is positive with respect to the chosen direction.

Proof: From definition 3.1, *v* must $\in N_+(u)$ accordingly $\Delta_u = ||(u\, L_u)|| - ||(v'\, L_u)||$ where *v'* is the projection point of *v* on the line $u\, L_u$ is positive. This ensures the progress of data in the chosen direction.

Lemma 3.3 If the network is sensing-covered so there exists at least one single or multi-hop path from any interior node k to the perimeter path.

Proof: It is well established that the sensing-covered networks are also connected [137],[146]. When the network is connected, there exists a single or multi-hop path between every pair of nodes within the network [137]. Given this definition, if *b* is a perimeter node in the boundary path and the network is connected, then there must be a single or multi-hop path connecting node *k* to node *b*. Fig 3.6 presents an example in which the boundary is the sequence *a,b,c,d,e,f,g,h,i,j,a* and node *k* has only one neighbor. Node *k* connects to the boundary by the forwarding path *k,l,m* that intersects the boundary at node *b*.

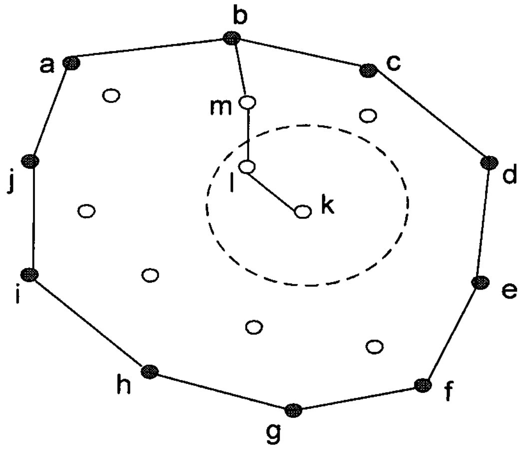


Fig. 3.6 Example of interior-path and perimeter path connectivity.

Theorem 3.1: If the network is covered and connected, the path to the perimeter will be always discovered.

Proof: Using lemma 3.1 and 3.2, the forwarding strategy shows progress in the chosen direction. Using lemma 3.3, the progress will eventually lead to the perimeter of the network.

□

3.4.3 Phase III: Sink Mobility

The mobility phase is a conditional phase: sinks stay stationary at their initial deployment locations until movement is enabled. Sinks continuously validate the enabling conditions and make the movement decision when required.

3.4.3.1 Enabling Sink Mobility

Mobility is enabled upon satisfying one of the following conditions: (a) a sink is exposed to a full partition at its current location, (b) the current number of its one-hop neighbors is less than a threshold value, or (c) the rate of receiving data at the current location is less than an expected threshold value.

Enabling sink mobility according to the first and second conditions is a straightforward mechanism. The third condition implies a predictable mechanism that could be performed by each sink while stationary. Each sink can measure the receiving rate at its current position, and mobility is enabled only if degradation below a certain level is noticed. Such degradation in the data collection rate could be a sign of an internal partition within the network. An internal partition cannot be directly detected by the sink if it does not occur at the level of its one-hop neighbours. Also, a combination of more than one situation could be considered. This could happen, for example, if the sink detects that the current number of its one-hop neighbours is less than a threshold value and the receiving rate goes below the expected rate at the current position. The threshold values depend on the application, but are expected to be function of number of sinks, network diameter, and the data generation rate.

Boundary Peeling Data Collection Algorithms

```
Input: h (integer parameter), $\varepsilon$ (waiting time units), $\lambda$ (connectivity threshold)
For all s $\in$ sinks // Initial Phase, execution frequency=1
{
Determine the boundary-segment
Compute the v-center
Broadcast v-center to all nodes
}
For all n $\in$ N {
   if Recieved-packet=v-center
   Compute the allowable directions
    Else if Received-packet = EXPLORE-packet
   Send topology-information to sender
   Else If Received-packet = GLUEBOUNDARY-packet
   {
   if Boundary-Condition=TRUE {
   Determine Next-Perimeter
   Memorize Chlid And Parent Perimeter
   Send GLUEBOUNDARY-packet to Next-Perimeter
   Send CONFIRM-packet to Sink
   }}
   Else if Received-packet= EVENT
   {
   If n $\notin$ boundary-nodes {
   Determine Next-Forwarder
   if Next-Forwarder $\neq$ Null {
   Send Received-packet to Next-forwarder
    Else
     Drop Packet && Notify Sender }
    Else if s $\in$ One-hop-neighbours
  Send Received-packet to s
   Else Send Recieved-packet to the Parent-Perimeter }
   }
  Else drop packet
  }
For all s $\in$ sinks {
   Wait timeout (current-timestamp + $\varepsilon$ )
  Check number of active neighbours
  if Connectivity $\leq$ threshold $\lambda$ {
  Move single step towards v-center
  Send EXPLORE-packet to Neighbours
  Wiat (topological-information of Neighbours)
  Send GLUEBOUNDARY-packet to Perimeter-Neighbours
  Collect (CONFIRM-packet) from Perimeter-Neighbours
  }
  If Received-packet = EVENT-packet
   Process Received-packet
  }
```

Fig. 3.7 Pseudo-code of Boundary Peeling Data Collection Protocol

3.4.3.2 Mobility Pattern

The movement pattern is such that the sink moves discrete steps toward the network *v-center*. So the *v-center* acts as a center of a pulling force that attracts the moving sinks (in contrast to the "away-from-centroid" dissemination strategy that is assumed for internal data propagation). The moving sink checks the validity of the moving condition in each step and moves to regain its initial situation. In each step, the sink moves a progressive distance equal to its transmission range in the direction of the line joining the current location of the sink to the network center.

3.4.3.3 Post-Mobility

Once the movement action is complete, the sink informs its current one-hop neighbors of its presence in their transmission region. It sends them an *explore* message and collects their replies. The sink reassigns the role of perimeter neighbors among them and sends a *glue-boundary* message. The perimeter nodes locally glue the boundary within their segment and adjust their hop-count towards the sink. They also modify their pointers to the upstream neighbors towards the closest sink if required. This mechanism retains the connectivity of sinks to the network boundary. A simplified pseudo-code of the proposed protocol is presented in fig 3.7.

3.5 Guard Region

The data forwarding mechanism described above increases the load on perimeter nodes. Eventually perimeter nodes at the network peripheral would be exposed to a *peeling phenomenon* (have their energy depleted exposing other inner nodes to be perimeter). The proposed deployment model with the associated data collection protocol reduces the negative effect of such peeling phenomenon. The peeling is distributed over the network boundary. The direction of the peeling is from outmost layers to inner ones (external peeling). Since the proposed mobility strategy allows the movement of sinks to follow the direction of the progressive peeling (towards the network center, i.e. from outmost to innermost), the sink-network connectivity is always maintained. This raises a question about the coverage percentage that could be tolerated by the application.

_In our solution, the loss in coverage is distributed along the network perimeter rather than concentrated around the data collector node/sink this kind of coverage loss could be tolerated by the application as following.

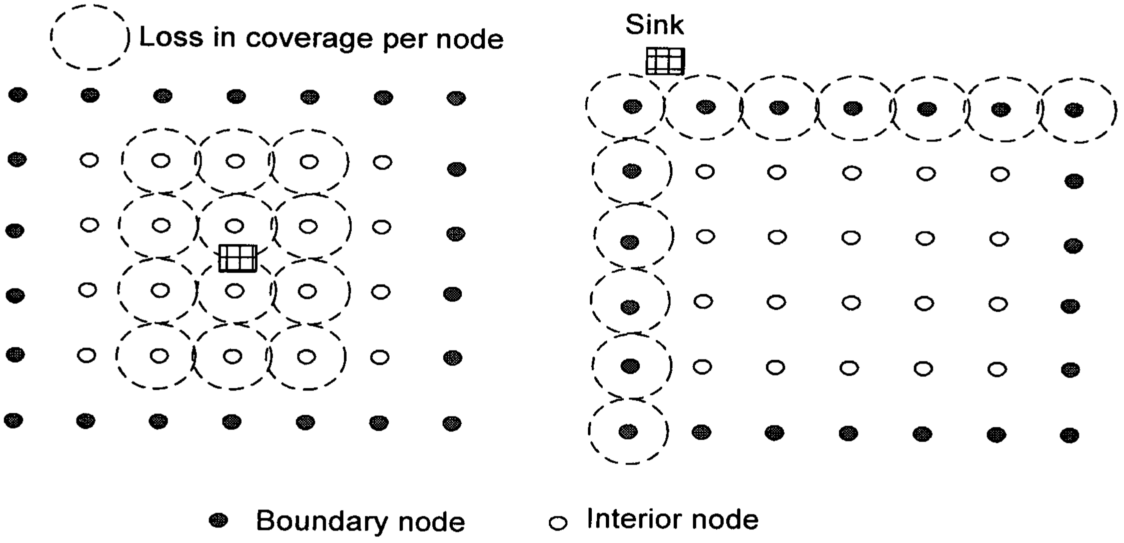


Fig. 3.8. External peeling vs. internal peeling.

Definition 3.3 External peeling is the collective effect of the increase in rate of dead nodes where the dead nodes are spread over the boundary of the network rather than concentrated in the core area (internal peeling). Figure 3.8 provides an example of external peeling vs. internal peeling when the same number of nodes dies.

Definition 3.4 Coverage tolerance ration CTR is the ratio between the area of depleted nodes to the total area of interest A'/A.

If the uniform node distribution is considered, the CTR ratio could be expressed in terms of number of depleted nodes to the total number of nodes *N'/N*. In the case of internal-sink deployment the peeling phenomenon also occurs. The peeling is concentrated around sinks. The direction of such peeling is toward the outmost layers. This internal peeling results in loss of coverage and early partition of the sink.

The outmost layer could be thought of as *guard layer* that protects the inner layers. If the deployment is planned such that the outer layer is assumed to be an additional layer to the core network layers, exposing such a layer to a peeling would not affect the interested coverage area. The network lifetime would increase in proportion to the number of nodes in the outer layer. The coverage would be maintained according to the application requirements.

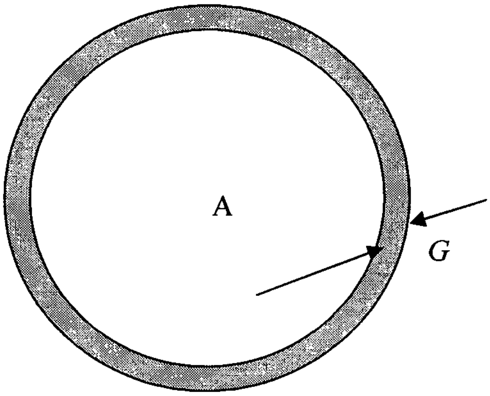


Fig. 3.9. Guard region: *A* is the core area of interest and *G* is its guard area.

Figure 3.9 shows the concept of the guard region. If A is the area of interest, *G* would be its *guard area* that forms a layer of distributed nodes with the goal of increasing the lifetime of the network. Lifetime in this case would be the time until the layer *G* is peeled off and nodes within A start to suffer energy depletion. *G* could be formed either by increasing the number of nodes at the outer layer, or by populating the nodes over a wider area to secure the network lifetime. The number of nodes *N'* that could cover *G* area could be determined as follows: if we consider the uniform node distribution model and the coronas based network training (as in figure 3.1), the number of nodes in outer layer $N'=N_n$ could be calculated in terms of both node density D, number of coronas n, and corona's width r as follows:

$$N_1 = D \times \pi \times r^2$$

$$N_i = D \times ((\pi \times (i \times r)^2 - (\pi \times ((i-1) \times r)^2)) \text{ , and}$$

$$N_n = D \times \pi \times r^2 (2 \times n - 1)$$

If a general model of node distribution is considered, then the number of nodes could be calculated with a goal of covering the guard area. If we consider A=A+G then where A is the total area to be covered , N could be determined according to the following relation, where r is the communication range of sensor nodes [61], [138].

$$N = \frac{A}{\pi \times r^2} \times 5.1774 \times \log N$$

The above relation binds the area to be covered to the required number of nodes. An area A could be fully covered with N nodes if the above relation is satisfied.

The increase in the lifetime t' could be expressed in terms of m, the number of peripheral sinks, where D is the density of nodes and p is the pattern of distribution.

$$t' = f(m, D, p)$$

Increasing m and/or D would result in the direct increase of t'. P has an indirect impact on t'. If p represents uniform node distribution, t' would be directly proportional to the other parameters. Random node distribution could result in random peeling. As there will be dense and sparse regions within the coverage area, the peeling could affect one part of the network more than other parts.

3.6 Performance Evaluation

We evaluated our protocol using the wireless sensor network simulator (WSNS) described in [28]. The simulator uses a multithreading mechanism to allow accurate network configuration in addition to attractive visualization ability.

3.6.1 Methodology and Setup

We built our protocol above the infrastructure layer that defines the coverage-area dimensions, the power model, the model of nodes' distribution, and the transmission/sensing ranges. Two categories of experiments were performed to test the behavior of the protocol in both dense and sparse networks. We used network terrain dimensions of 400 m × 400 m with sensor densities ranging from 200 to 400 nodes (randomly distributed). Different radio ranges T_x (ranging from 60m to 90m) were used to maintain the connectivity of the network with the changed densities. Experiments were conducted with different numbers of sinks deployed dynamically at the network peripheral. In all the experiments, the initial energy of the sensor nodes was equal. We assumed that each data transmission would consume one unit of energy. For simplicity, the third mobility condition mentioned in the previous section was not simulated. We stopped the simulation when a partition occurred.

A third category of experiments was conducted with the centric-sink network model and the performance was compared to our model. In the centric-sink model, a single sink was located at the center of the network and the shortest path routing SPR was applied as the underlying routing protocol (this model is considered as an ideal model in [2], [6], [9], and [120]). Since nodes are stationary and no underlying MAC protocol is considered, routing tables of the SPR are determined at the initial phase. We then allow a control packets transmission session in every simulation round (to update the routing information and cope with routing failures due to the increase in the number of dead nodes). We assume that each node consumes 0.5 unit of energy to transmit one control message. Results from the simulation are collected at different rounds. Each round is equivalent to two data transmission sessions and one control transmission session. To ensure consistency, all the plotted results are the average of the outputs of 5 runs of each experiment.

The last category of experiments is conducted to compare the performance of our mechanism to the Mobiroute [9]. MobiRoute is a protocol that considers mobility to enhance the overall network lifetime. We have simulated MobiRoute at the application layer where no MAC related issues are considered. We simulated both its static and adaptive strategies. The protocol requires that one mobile sink moves around the network peripheral collecting data at certain anchor positions. In static methodology, the anchors are randomly chosen and stay fixed during the network operation. In adaptive version, the anchors are selected upon determining their power profile P-profile as defined in [9]. The protocol involves two main phases: data collection phase and sampling phase. Each data collection round is preceded by a sampling round. In a sampling round, the sink stops at the anchor positions and determines the optimal visiting time by sampling the global power consumption. The visiting time is then applied to control the movement of the sink while in the data collection round.

3.6.2 Metrics and Experiments

For each configuration scenario in the conducted experiments, the level of each node with respect to the network center is determined. Level 1 nodes are those located at a distance less than T_X with respect to the network center c. Level 2 nodes are those located at a distance larger than T_X and smaller than two T_X with respect to c and so on. Fig 3.10 shows relative node densities of different network layers for both dense and sparse network configurations.

Layer one is the nearest one to the network center. In dense networks, the nodes are distributed over three layers (we increase the transmission radius to simulate dense configuration). In sparse networks, nodes are distributed over five layers. Snapshots of simulation runs of our protocol at different simulation rounds are shown in figure A1.1. It shows the progressive updates (shrinking) of the network boundary (that we call the peeling phenomenon) in a dense network scenario. It also illustrates the movement of sinks towards the network center (sinks are not necessarily moving simultaneously). It could be a case where an internal partition occurs. Internal partition refers to the case where some internal nodes form an isolated island. Even at the time such partition occurs, as we will show in fig (A1.1-d), a large part of the network is still connected to one or more sinks and the network is still operable. This feature of our protocol increases the utilization of the network resources.

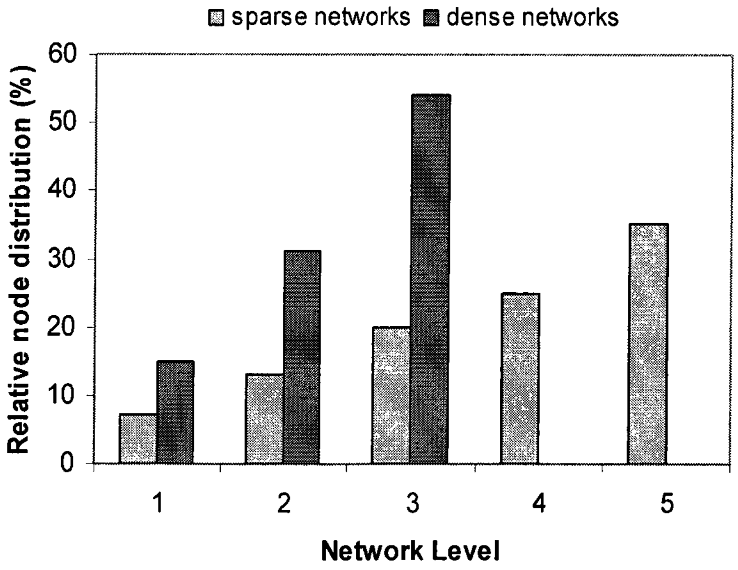


Fig. 3.10. Relative node distribution

3.6.2.1 Average Load

To show how our deployment model with the associated data collection protocol results in load balancing through the network, the *average load* at each network level is measured and depicted in both fig 3.11 and 3.12. The figures show that our protocol definitely balances the average load at each network layer except the outmost one (network level 5 in sparse settings and 3 in dense settings). The figures also show that increasing the number of peripheral sinks results in decreasing the average load at the outmost layer. The inner layers would exhibit approximately the same average load. In the case of the centric-sink model with the shortest path routing, the innermost layer (level 1) is exhibiting the higher load amongst others. The gradual increase in the average load from outmost layer to innermost layer is higher than that exhibited by our model for both sparse and dense networks.

In the case of MobiRoute, the average load depends on the number and positions of the selected anchor points. Small numbers of anchors increase latency as the sink is allowed to collect the data only at such points. Moreover, the packet loss would be increased as the packets are redirected through longer hop paths towards the new anchor positions. Large numbers of anchors at the network peripheral reduces load balancing and increases energy consumption. The sink is required to broadcast topological update messages when it arrives at each anchor position. The network might be exposed to a sink-network partition before the sink completes one data collection round. So the number and positions of the anchors must be carefully selected. The sink requires this topological information in order to adapt to the topological updates.

Fig 3.11 and 3.12 show the average load of MobiRoute with eight anchor points selected uniformly over the network peripheral. The load exhibited at each layer is higher than that exhibited in our protocol. MobiRoute tends to balance the load among the network layers at a higher average load value. The innermost layer exhibits the lower average load in both dense and sparse configurations and the outmost layer exhibits the higher average load.

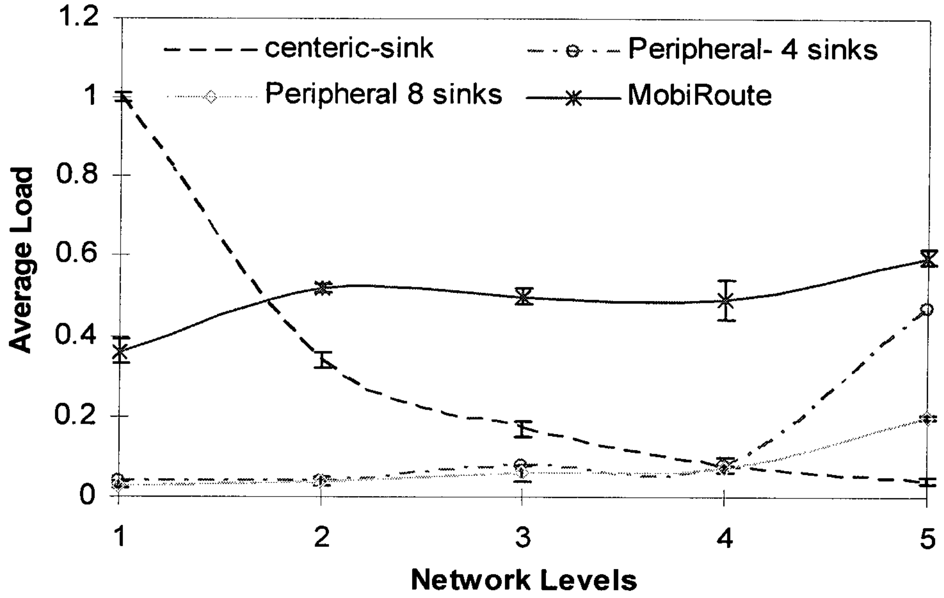


Fig. 3.11 The average load at each network level (sparse network configuration).

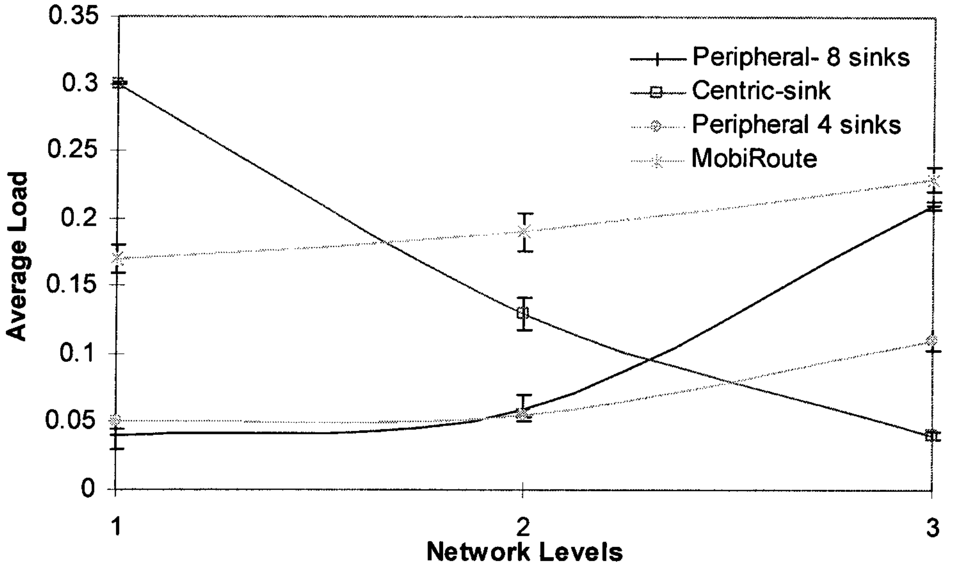


Fig. 3.12 The average load at each network level (dense network configuration).

3.6.2.2 The Rate of Dead Nodes

The *number of dead nodes* is measured at each simulation round and the average rate of dead nodes is depicted in fig 3.13 and 3.14. The figures show that our protocol exhibits a higher rate of dead nodes than that exhibited in the case of the centric-sink model as well as MobiRoute. Most of these nodes are distributed over the network boundary, which reduces the negative effect of their early deaths. The figures illustrate the simulation rounds at which the partition occurs. The results are the average results obtained for different network settings where 10 simulation rounds are allowed per each setting. In sparse settings, our model maintains the network connectivity more than the time exhibited by the other models.

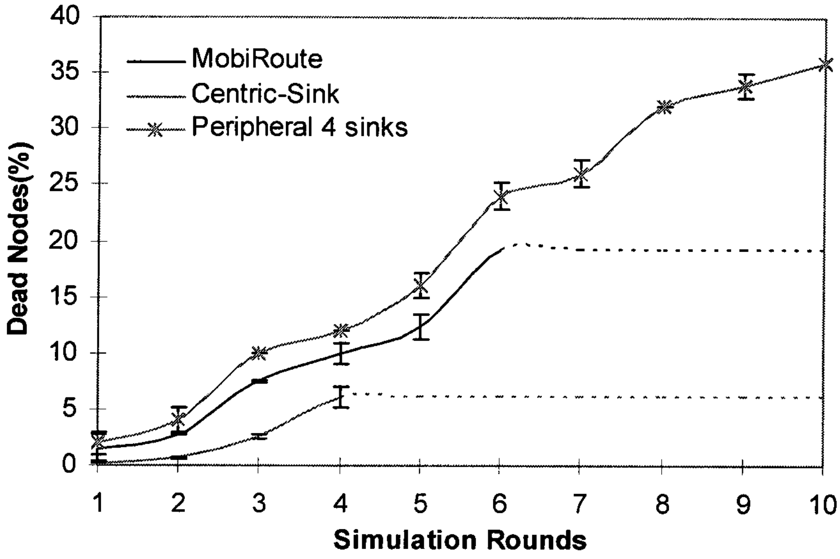


Fig. 3.13 The rate of dead nodes at different simulation rounds for sparse configuration. Dotted segments explain the network partition.

In dense settings, the partition is more delayed than in the case of sparse networks for the three models. The partition occurred around the seventh simulation round in the case of the centric-model, while no partition appeared using our model over the ten simulation rounds. In the case of MobiRoute, the rate of death is comparable to that in our case. Most of these dead nodes are centered on the anchor positions. Fig. 3.15 shows a partition pattern in the case of the centric-sink model (a snapshot of a simulation run).

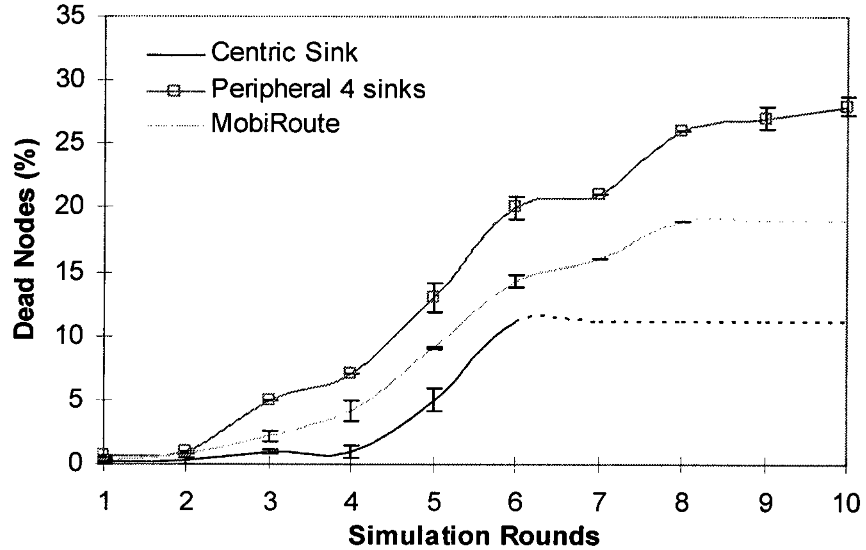


Fig. 3.14 The rate of dead nodes at different simulation rounds for dense configuration. Dotted segments explain the network partition.

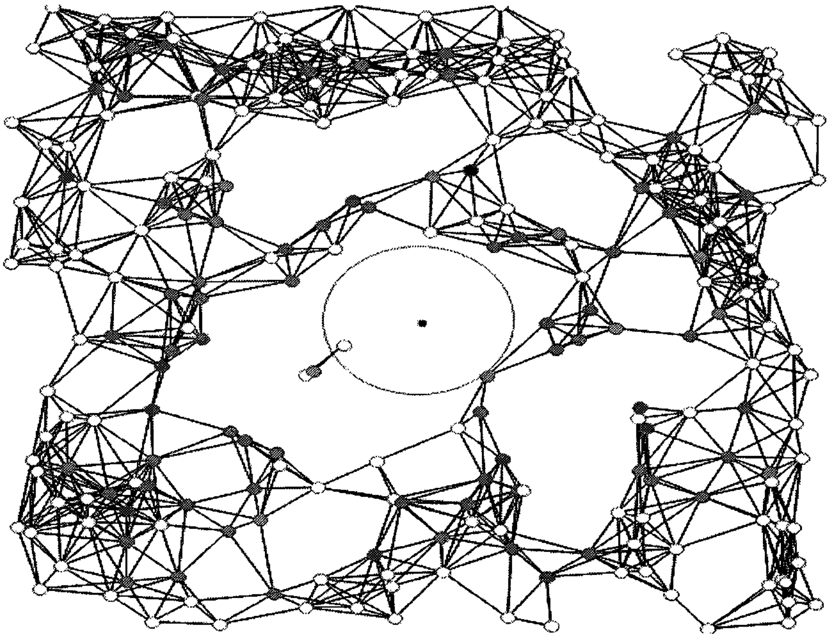


Fig. 3.15 A possible partition pattern in the case of the centric-sink model; the sink is represented by the small square at the center; the circle around the sink represents its connectivity region.

3.6.2.3 Average success Rate

We also measure the *success rate* of the three models in the two network categories. The success rate measures the capability of the live nodes to send their data successfully to the sink. Fig. 3.16 shows the average results obtained for both dense and sparse configurations.

The three models demonstrate a 100% success rate at the early simulation rounds. Our deployment model, with the associated data collection protocol, keeps the success rate high during the whole simulation (in both network configurations). The figure shows that, at the time a partition occurs, the decrease in the success rate is very small. Most of the live nodes are able to send their data successfully to the sink. Fig (A1.1-d) shows an example where only a small percentage of the live nodes form an isolated island.

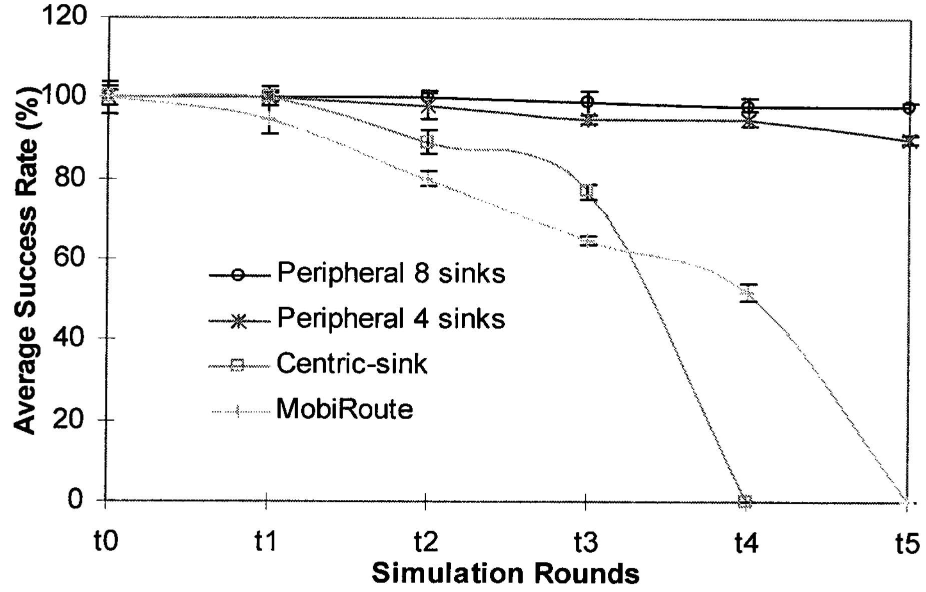


Fig. 3.16 The average success rate.

The success rate in the case of both MobiRoute and the Centric-sink model heavily depends on the transmission rate of the control messages. MobiRoute uses two different control messages: s-beacon messages to inform the last-hop nodes when links to the sink are broken, in addition to topological update messages that are broadcasted to all nodes when the sink arrives in an anchor position. Such messages provide freshness for the routing tables. Increasing the rate of control messages increases the success rate, yet decreases the network lifetime. This is due to increasing the energy consumption rate at each node, which in turn fastens the network partition. When the sink is partitioned from the network, the success rate drops to zero even though a large number of nodes could be alive. In the case of the centric-sink model with a static sink when a partition occurs, the network lifetime ends. In the case of MobiRoute, the movement pattern that depends on visiting fixed anchor points at the network peripheral does not seem to be enough to adapt to the topological updates. The network is vulnerable to the sink-network partition even though mobility is included. The sink loses its connectivity to the network once the one-hop neighbors at the selected anchors have their energy depleted.

3.6.2.4 Packet Loss

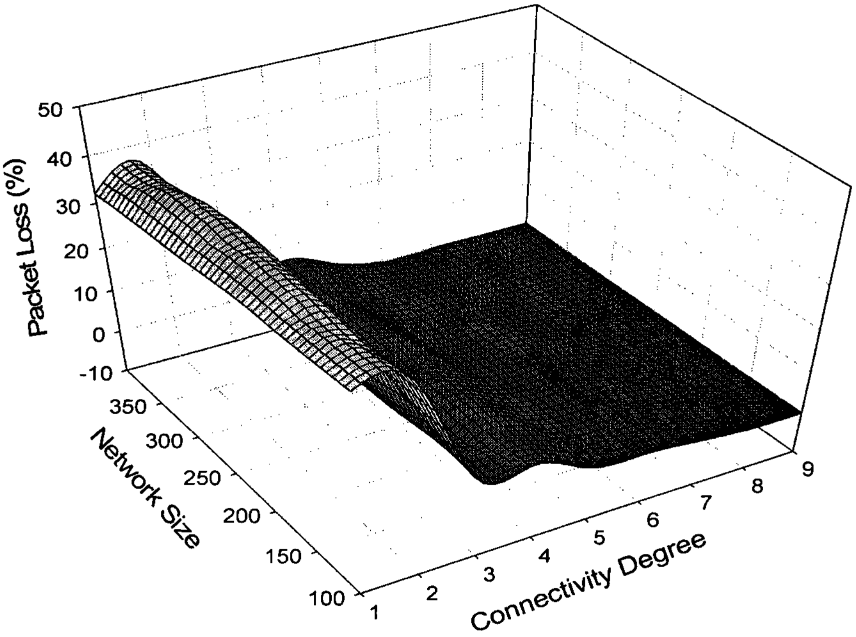


Fig. 3.17 The rate of packet drop vs. connectivity degree.

In this experiment, the rate of packet loss is measured under various degrees of connectivity, where nodes are distributed uniformly random. The average degree of node connectivity (the average number of one-hop neighbors) is increased from 1 to 9 for different network sizes. Each node is allowed to transmit one data packet in each direction and then the rate of the packet loss is measured. The results in figure 3.17 show that the packet drop rate exhibits a sharp decrease at the node connectivity of 3. It decreases approximately from 40% to less than 5% when the degree of node connectivity reaches four neighbors. It also shows that, at higher node connectivity, the packet loss decreases to zero.

3.6.2.5 Normalized Network Lifetime

The normalized lifetime is measured for two situations: with and without considering guard region GR. Without considering GR, the normalized network lifetime is shown in fig. 3.18. The figure shows the relative increase in the network lifetime achieved by our protocol. It also shows the effect of increasing the number of peripheral sinks.

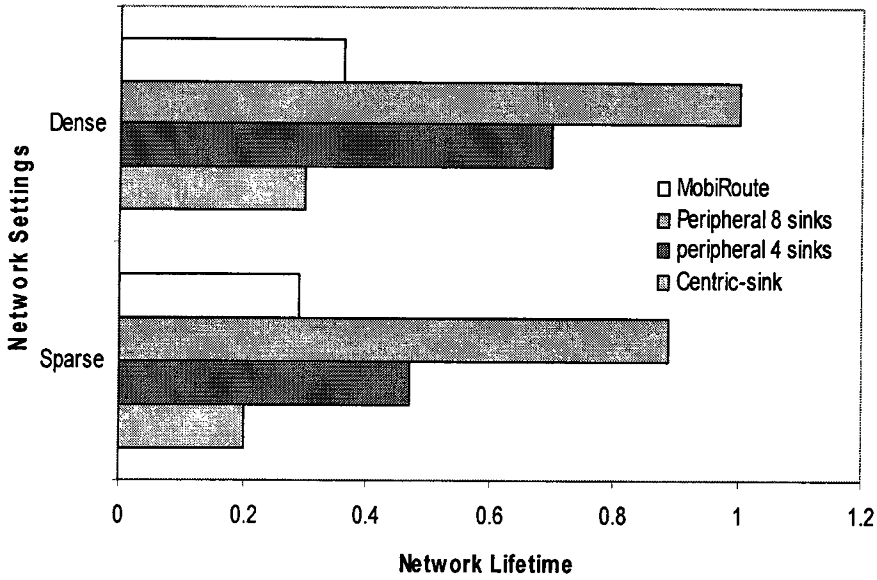


Fig. 3.18 Normalized network lifetime

The normalized lifetime with a guard region applied to the network area is depicted in fig. 3.19. A guard region is constructed at the outmost layer by increasing the number of nodes at that layer by 25% and 50% of its initial number of nodes. Fig. 3.19 shows the increase in the normalized lifetime of the network in both dense and sparse configurations.

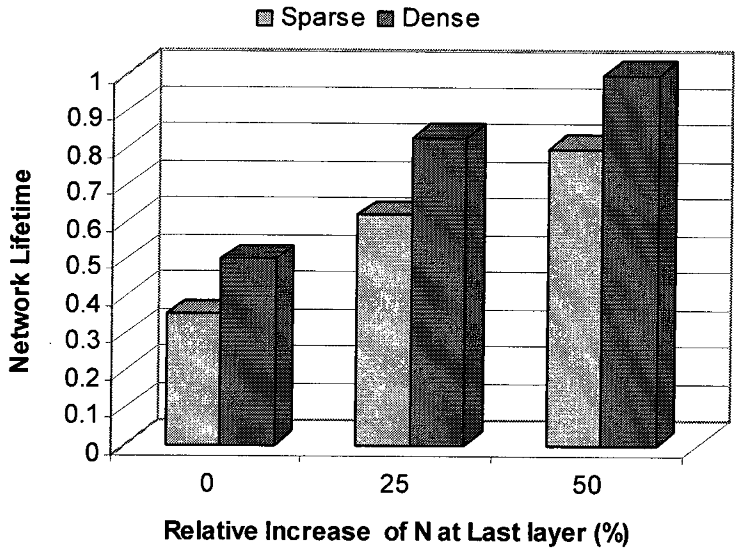


Fig. 3.19 Normalized network lifetime with guard region at outmost layer

3.7 Discussions

Random sink mobility: our solution described in this chapter exploits the controlled mobility of the sink to maximize sensor network lifetime. Random mobility of the sink was the early view of sink mobility and has received a lot of attention [93],[94]. Techniques that maximize lifetime by balancing the load while mitigating the negative impact of random sink mobility are also available [69],[97].

Converge loss: Coverage in sensor network could vary according to the application requirements (in terms of composition of the area of interest and the redundancy in coverage). For example, an application could require area coverage (where each point in the area of interest should be covered) or target coverage (where only specific points/targets within the area have to be covered). The redundancy level of coverage could range from α -

coverage (where a fraction α of the region should be covered) to k-coverage (where each point in the region should be covered by at least k sensors) [148]. This variety of the types and levels of coverage makes coverage a huge research issue that is out of the scope of our work. However, there is a fact that if a single node dies, this will lead to a loss of its sensing capability and hence participates into loss of coverage. This is the reason that the lifetime of the network is usually defined as the time until the first node dies. The argument against this definition is that the network could still provide useful information long time after this event. Moreover techniques that build upon such definition alleviate themselves from coping with topological changes (the first node to die results in the first change in the network topology) [149]. So defining the lifetime based only on loss of coverage is not sufficient. It could be the situation where only those few nodes in the vicinity of the sink die. This could result in an inoperable network even though most of network nodes are alive with their coverage preserved. So tolerance in coverage loss could be actually an application decision. Our solution in this chapter provides a mechanism to allow application to tolerate the loss in coverage while insuring the sink-to network connectivity.

Reliable boundary recognition: While we opt for the BoundHole algorithm to define the boundary nodes in a given topology, other algorithms such as *GFG* [87] could also be adopted to achieve the same goal. This algorithm has a boundary detection ability in nature. Reference [150] shows how it could be utilized to detect the outer boundary, known as the outer face. While the *BoundHole* algorithm is a variant of face routing algorithms, *GFG* combines the greedy mechanism with the face routing to provide a routing service between a source and a destination.

Mobility and large scale irregularities Sensing fields that have large obstacles could prevent smooth mobility within the sensing field. When holes and obstacles are distributed irregularly over the sensor field, the toward-centroid mobility policy might be not enough to cope with such large scale irregularities. In such situations, a shape segmentation algorithm for sensor networks [17] could be adapted prior to sink deployment. Sinks could be distributed at the boundaries of the resulting shape segments and the segment-centroid is calculated. Toward-centroid policy is then locally applied at each shape segment. The distribution of sinks would be according to the number of shape segments and/or the lengths of each segment boundary.

Cost vs. performance enhancement The results show that our deployment strategy, with the associated data collection mechanism, outperforms both static and adaptive MobiRoute. Of course, such enhancement in the performance comes at the rate of increasing the hardware cost: multiple sinks are considered instead of a single mobile sink. We believe that this kind of hardware cost is intuitively acceptable if it could be reflected in performance enhancement. Many companies are ready to pay such costs to benefit from such reliable and dependable networks [151]. At the same time, in large scale sensor networks, it is intuitive that relying on a single sink to gather the data and monitor the network is not the best choice.

Variable data traffic and data delivery model The simulation part of this work has considered the uniform continuous data traffic model. In such a model, all nodes within the network have to report a data packet at the same time or with a small random drift time window. This model is known to be the upper boundary of data traffic models where the data generation rate at each node is variable. The probability of congestion is higher than in such traffic models. We expect that our protocol, with its underlying data propagation mechanism, would exhibit better performance when such models are adopted. On the other hand, the applicability of the proposed solution in supporting the wide spectrum of the sensor network applications should be addressed. Many applications require submitting queries and getting only the interested responses rather than continual data reporting. The problem of developing an efficient query based data delivery model that extends the network lifetime is addressed in the next chapter.

3.8 Summary

The work presented in this chapter describes the boundary-peeling data collection protocol and the associated sink mobility and deployment models. The presented solution combines the benefits of using multiple-mobile sinks, clustering, and non-uniform node distribution to maximize sensor network lifetime (with performing neither clustering nor non-uniform node distribution). The contributions of the proposed mechanism includes the following: (i) it combines the benefits of using multiple and mobile sinks without introducing the problems associated with the sink mobility, (ii) it adapts to the topological changes in an efficient and dynamic manner with no overhead to exchange topological update messages, and (iii) it balances the load among the nodes dynamically. The protocol with the associated deployment model is highly reliable. It is able to retain the network connectivity as long as the coverage percentage is tolerated by the application. The above features of the proposed mechanism allow for better utilization of network resources. The mechanism is able to maximize the lifetime of the network while maintaining a high degree of network connectivity.

Chapter 4

Adaptive Rendezvous Data Dissemination for Irregular Sensor Networks

Rendezvous mechanisms that mix pull and push dissemination strategies have significant importance as reporting mechanisms in sensor networks. Ordinary rendezvous approaches are optimized for flat architecture and face challenges such as irregularities in topology and the uncontrolled propagation of events. In this chapter, we introduce TRDD, a scalable rendezvous data dissemination approach for irregular, tiered sensor network architecture. TRDD is topology-adaptive, query-adaptive and content-independent rendezvous protocol that aims at widen the applicability of rendezvous mechanisms by overcoming their limitation and enhancing their query capabilities. The chapter is organized as follows. Section 4.1 provides an introduction. Section 4.2 explains the limitations exhibited by the existing rendezvous mechanisms. Section 4.3 describes the design details of the proposed Topology-based Rendezvous Data Dissemination (TRDD) protocol. Section 4.4 describes the adaptive behavior of the proposed protocol. Section 4.5 summarizes the performance of the approach. Summary and discussions are outlined in Section 4.6.

4.1 Introduction

Data dissemination is one of the significant topics of current research in wireless sensor network. As sensor networks are highly energy-constrained, the development of energy-optimized protocols has received considerable attention. In early work, sensor networks were assumed to be homogenous. Two basic data dissemination models, *one-many* and *many-to-many*, have been introduced. In the *one-many* models, specific data collector(s) (usually known as a sinks) are defined in the network. In the *many-to-many* model, each node in the network can act as both a data collector as well as a data producer [10].

Data dissemination models not only address the directions in which data is disseminated, they also define the underlying storage system. To date, three canonical storage systems have been introduced: local, external and in-network. In a local storage system, each sensor stores the detected events locally. Querying the network ("pulling data") can be expensive, since, with no prior information about the location of the relevant events, each node must be queried. Most pull-based approaches flood the whole network with queries. The high energy consumption makes this approach unsuitable for large-scale sensor networks, especially if the number of nodes with data to report is small relative to the network size. This model is used in [35], [52] among others.

In an external storage system, any detected event travels to the data collector where it will be stored ("pushing data"). This happens regardless of whether the events are useful to the application or the end user. Communications are no longer on-demand and network resources are wasted. An early example of this model is presented in [48].

In-network, or hybrid, storage approaches combine the advantages of both these models in an attempt to achieve the desired balance [70]. Sensors that generate events send them to other owner sensors in the network. Generally, hybrid systems can be classified as either content-based or content-independent. In content-based models, [71], [72], [73], [76], the owner sensors are usually defined by a mapping function linking the content of the data with a location or ID within the network. In content-independent models, [70], [74], [75], events and queries are sent to a self-structured sub-network regions. Querying the network in hybrid models relies on the concept of "rendezvous". Events travel to the rendezvous regions where they will be stored and queries are directed to the rendezvous regions on demand. Current rendezvous solutions are optimized for flat architecture where any node could be a data producer as well as a data sink.

Recently, some work in the literature demonstrates the need for heterogeneity and challenges the assumption that sensor networks are completely ad–hoc and have no centralized or cooperative management. This gives rise to the trend towards sensor network tiered architectures [61], [62], [126],[152],[153] where multiple sinks exist within the field to cooperate in monitoring the network and collecting data to be sent to the end user (an end tier). The existence of multiple sinks renders the rendezvous data dissemination protocols that previously consider multiple independent sinks, or that consider no specific sink inefficient. Coordination and load balancing become essential components of the devised protocols. Most of the existing solutions do not consider the nature of cooperation between the existing sinks. When sink mobility is considered the design of efficient rendezvous data dissemination protocol for such architecture is more complicated. Existing solutions consider only random sink mobility where the movement of sinks is uncontrolled (e.g. mobile users).

In this chapter, we start by analyzing the challenges that should be tackled in order to design an efficient data dissemination rendezvous scheme for irregular tiered wireless sensor networks. We then present a novel solution that overcomes the limitation of the existing solutions and tackles these challenges. We aim to enhance the efficiency of the hybrid model and contribute towards widen of its applicability. To illustrate the efficiency of the proposed solution, a simple generic analytical model is provided. Moreover, a comparison based on

Grid analysis model to compare our solution to benchmark protocols that exemplify both query-driven and hybrid schemes is given. Simulation experiments are also conducted to compare the performance of the proposed solution to the benchmark Two-Tier Data Dissemination (TTDD) protocol [18]. Both analysis and simulation results show the robustness and efficiency of the proposed solution in any network topology whether regular or irregular. Results also show the reduction in the communication cost and the even distribution of events propagation through the network.

4.2 Limitations of Data Dissemination Rendezvous Solutions

1-Uncontrolled Events Propagation

Rendezvous approaches that balance pull and push strategies are known to be complementary and efficient under some conditions [18], [71]. The main limitation is that they cannot avoid the problem that we call "uncontrolled events propagation". Current rendezvous approaches include both push and pull components that are working independently. With a long time-interval between successive queries, resources are wasted in communicating, processing and storing unimportant events that travel to the rendezvous region/nodes even though they are not of current interest. Resources are also wasted when events are not queried before being discarded by the temporary destination nodes. (Usually a "lifetime" parameter is used to determine how long the temporary destination stores events).

2-The impact of topological irregularities

Irregular network topologies impose a challenge for rendezvous schemes that build virtual infrastructure over the arbitrary topology. Irregularity may result in performance degradation of such schemes as it results in unbalanced rendezvous infrastructure. For example, in an irregular sensor field, the virtual grids in TTDD [18] become unbalanced. The grid cells would have vastly varying areas which result in degradation of the performance. This is due to the increase in the flooding rate in the grid cells and the increase of formation of hot-spots dissemination points. In many situations, the virtual infrastructure will suffer imperfection that questions the reliability and the design precision of the scheme. For example , if comb branches in Comb-needle CN [74] are not uniformly distributed over the network area due to irregularity, needles initiated in some regions become disconnected and unable to intersect the comb branches. Another example could also be found in schemes that imply row-column intersection such as [86] where rows initiated by nodes within irregular regions will never intersect columns initiated on other regions. Rendezvous location-service schemes (such as quorum-based solutions [81] and grid-based ALS [84]) adapt better to such imperfection by utilizing a recovery mechanism (in most cases, face routing [87] around voids and perimeter of the network) at the cost of extra overhead. Moreover, finding the optimal value of some static topological parameters (e.g. the cell size parameter in ALS and TTDD , the needle length s in CN and the width of the line w in [78]) becomes a problem when the network shape is irregular [17]. Figure 4.1 (a,b) shows some examples : (i) nodes that are located along the horizontal line H-line1 will never be able to access data transmitted along the vertical line V-line1 and visa versa, (ii) A comb initiated by node at location 2 will not be complete and needles at other region with length α will not be able to access the data within the comb, (ii) the impact of changing the cell size and/or the x_{base} ,y_{base} is also displayed.

3-Operability within Tiered Architecture

Advances in hardware design have provided powerful wireless devices (access points/sinks) that could be adopted with little cost. These advances raise the trend towards sensor network tiered architecture. The end user (management site) takes advantage of their power by utilizing multiple units to manage the network remotely [154]. When such powerful units have a mobility facility, the mobility could be controlled to further add to the flexibility of the network [123]. To the best of our knowledge, existing rendezvous data dissemination solutions are not optimized for tiered architecture as they do not specify any strategy for cooperation between such powerful units. The dissemination infrastructure is usually built upon the assumption that sources and destinations of the data are unknown to each other and the data entry could be initiated at any point in the network. This implies that each node could be a source as well as a destination of the data. This also could be the case when the destinations (sinks) move randomly and independent from each other as considered by most rendezvous-based location-service schemes. When data access points/sinks are stationary or the mobility of these units is controlled, the destination of the data is known or predictable. This contradicts the basic assumption and renders the resulting rendezvous infrastructure inefficient.

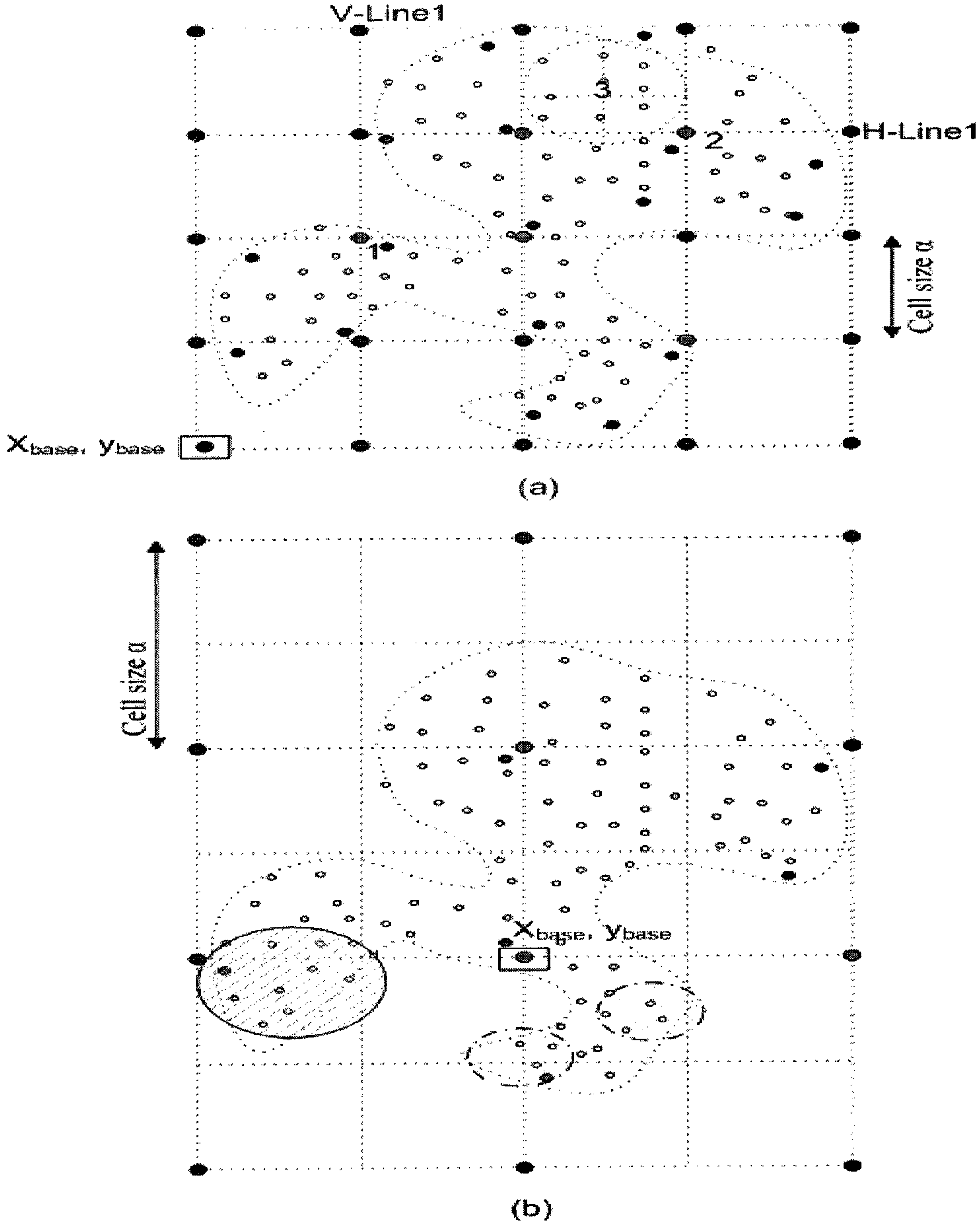


Fig. 4.1. Examples of the impact of topological irregularities.

4.3 Topology-based Rendezvous Data Dissemination (TRDD)

Our contribution towards developing efficient rendezvous data dissemination protocol for large-scale sensor networks starts by considering the key limitations discussed above. We emphasize synchronization of underlying pull and push phases in the sense of making them aware of each other. This would prevent resource wasting and leverage on-demand data communication. Then we approach the topological adaptation by smoothly integrating a topology detection algorithm from the literature so that the devised protocol would be topology-adaptive. This allows the protocol to work with equal efficiency in both regular as well as irregular sensor networks. To the best of our knowledge, most academic work considers topology detection and data dissemination separately even though data dissemination at the upper layer is greatly affected by the underlying protocols. We show that, by integrating them, we can provide robust rendezvous data dissemination. Moreover, the proposed protocol is operable and optimized for sensor networks tiered architecture. The existing access points/sinks cooperate as a singe tier to build a reliable rendezvous infrastructure that maintains efficient communications. When these sinks are mobile, the mobility could be controlled according to our proposed strategy presented in chapter 3. The ability to adapt the dissemination behavior to the query types and the dynamicity of the underlying phases produce a robust operational network. This increases the interesting features that could make it the model of choice, especially for remote sensing applications.

4.3.1 TRDD Overview

As in all rendezvous solutions, the proposed solution consists of two phases: the push (events propagations) phase and the pull (query propagations) phase. First, the dissemination structure of the proposed rendezvous scheme is based on a simple geometric idea. Considering the geometric polygon, any straight line from inside to outside (or vice versa) would intersect with the closed boundary of the polygon. In any network topology, whether regular (e.g., Grid) or irregular, the network perimeter could be considered as a polygon that provides a region that, for the interior nodes, is intuitively closed. Assuming that queries propagate over the perimeter region, building event paths that progress in any direction would meet the submitted query at the query region. This basic idea is enough to enable building a self-structured rendezvous scheme that works with equal efficiency for any topological shape. Second, submitting the query over the network perimeter (pull-phase) ensures that a relatively small subset of network nodes receives the query. If this subset explicitly triggers the interior nodes to start reporting their data (initiate the push-phase), the communication traffic produced by both phases is synchronized. So the traffic would be mainly on-demand and no events propagate between successive queries. Finally, the even distribution of the query region over the network perimeter allows the even propagation of events through the network. Data forwarding strategies that spread the traffic and result in a high degree of load balancing are proposed

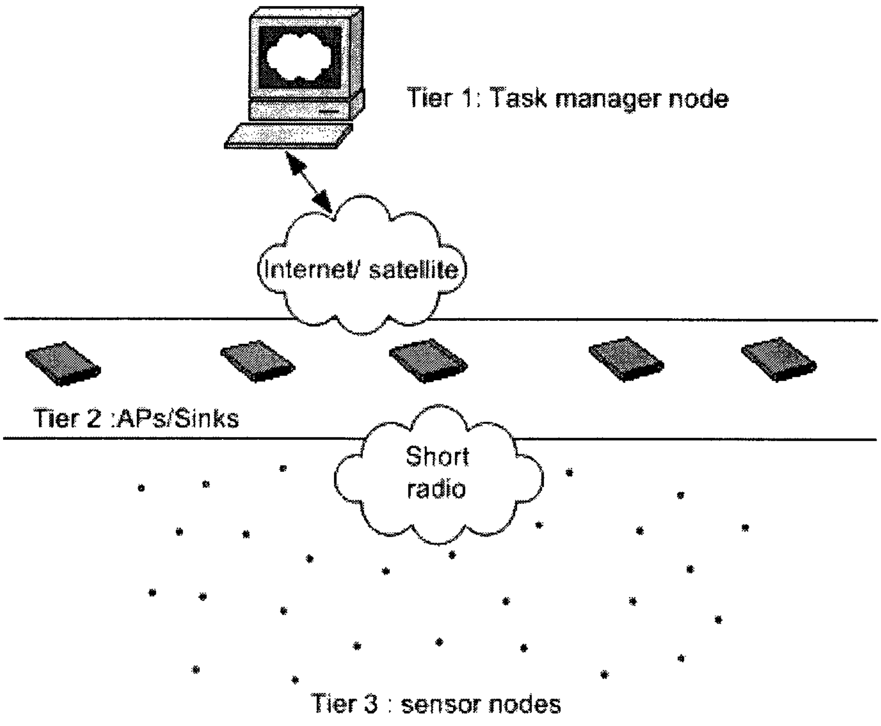


Fig. 4.2. TRDD Sensor Network Architecture Model

4.3.2 Environmental Model

Deployment Model: figure 4.2 shows the underlying network model. We generalize the network model presented in chapter 3 to into a flexible three-tier network model. A two-dimensional terrain area, with *n* randomly distributed sensor nodes makes up the *lower tier*. Sensor nodes are assumed to be aware of their coordinates. This could be achieved either by using GPS or by existing localized techniques [155]. The middle tier consists of *m* stationary sinks randomly placed at the network peripheral ($m << n$). The upper tier, the management site, can query and control the whole network through these sinks. Control information and queries are simultaneously received from the management site through a transient network. Sinks work cooperatively to build the dissemination structure (hence, they work as a single tier). Although the proposed protocol works for any number of sinks, the effective number depends on the network size and topology shape. So the number can vary according to the application. First sinks are assumed to be stationary then the impact of sink mobility as well as the interior deployment of sinks is investigated.

Query Model and Power Model: In sensor networks, data transmissions are known to be the greatest source of power consumption [32]. So the number of transmissions during the dissemination process is a good indicator of the rate of energy consumption. For discussion purposes, snapshot queries [35] are considered. Snapshot queries are of interest as active events only when the query is issued; neither future nor past events are important. Events and queries are stored as soft-state and allowed to be discarded after an event lifetime or query lifetime, each of which is an adjustable parameter. The ability of the protocol to behave adaptively according to the type of the submitted queries is also described.

4.3.3 TRDD Design

Considering the tiered architecture model in figure 4.2, the basic concepts of the TRDD is as follows. The multiple sinks cooperatively build a query region QR along the perimeter of the network. This QR surrounds the network interior and forms a virtual closed region. The construction of the QR explicitly triggers interior nodes to initiate event paths to carry the events to the QR. The events could propagate in any direction yet the intersection at the query region is guaranteed. Different policies by which nodes could select the direction of events propagation (random, round-robin, centroid-based) are investigated. All of them guarantee the query-events intersection. In addition to the topological adaptation, the protocol has the ability to adapt its dissemination behavior according to the type of the submitted queries. The detailed design of the TRDD protocol involves two main procedures: building the query region when the queries are submitted and propagating the events towards the QR.

4.3.3.1 Building the QR Over the Network Perimeter

In order to build the QR over the network perimeter, the network perimeter should be either known or discovered. If the underlying network is a regularly shaped networks (such as in the case of Grids or Circular networks), QR can be constructed over the network perimeter based on a trajectory-forwarding mechanism [156] where queries are propagated guided by the well-defined trajectories that represent the network boundary. More internal trajectories can be embedded in order to establish virtual sectors in the network. This requires the network map information to be at the management site so that sinks are supported with the trajectory segments that they are required to embed. A regularly shaped network is perhaps a special case. In irregularly shaped networks, trajectories that represent the perimeter of the network may no longer be well defined. QR should surround the network and form a virtual closed region in order to allow the events paths to be built in any direction yet intersect the QR. This calls for boundary awareness. So the first step in building the QR is the boundary recognition process described in Chapter 3. The process of marking the network boundary can be implemented either once at set-up time or on-the-fly piggybacked with the query dissemination process. Implementation at set-up time requires local maintenance to cope with topological changes affecting the outer boundary. On-the-fly implementation does not require additional maintenance as the last updates in the outer boundary are always captured. Figure 4.3 is a snapshot of a simulation run that shows the implementation of boundary recognition process by four sinks located at the outer boundary of the network. The figure shows that, for irregular network topology, the boundary nodes can be determined and a closed region can be marked. The boundary nodes play the leading role in the query propagation phase

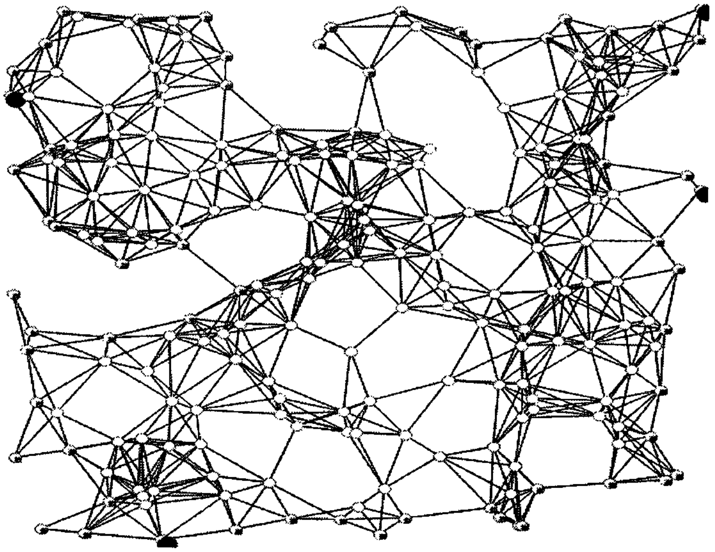


Fig. 4.3. Solid black circles represent sinks, Boundary nodes are highlighted with small dark rectangulars.

4.3.3.2 Query Propagation

Sinks construct the virtual QR along the network perimeter once the query is received from the management site. The QR is made up of the boundary nodes discovered by the boundary marking process and their one-hop neighbors. Each sink initiates multi-hop query propagation towards the immediate sinks. The propagation is performed hop by hop: each boundary node receives the query packet, transmits it to its one-hop neighbors after setting a pointer to its parent boundary neighbor. The child boundary neighbor then transmits the packet to its own one-hop neighbors and the process continues. In this way each sink becomes a root of a thin query routing tree extends over the network perimeter towards the immediate sinks. Nodes in the QR report to the root sinks through the constructed paths in reverse direction. While constructing the QR, duplicate suppression is performed (nodes transmit the same packet only once using packet's unique ID). Upon receiving the query packets, the one-hop neighbors of the boundary nodes transmit what we call a beacon packet. This beacon could range from a single bit to multiple bits. In some cases hardware facilities are used to allow nodes to wake up other nodes in their vicinity over a separate low rate physical channel. In such cases the soft beacon could be replaced by such physical hardware signals at no cost.

The main roles of the beacon signal are i) to allow interior nodes located outside QR to update their neighbor table based on hearing from their active neighbors, ii) to indicate to these nodes that the QR is constructed. This in turn means that the query is submitted at the QR and triggers interior nodes to start reporting their events. Nodes with live events (those already detected whose lifetime has not expired) initiate event propagation paths that work as bridges to carry their data to the newly-constructed QR.

```
TRDD Query Propagation Algorithm
________________________________________________________________
Input: Packet p at node n
If p=Query-packet {
   Check current query-session
   If p not_exist
     {
      Set Parent=Sender
      Add p to the query-list of the current query-session
     if n ∈ QR
      Send p
      else {
      Create Beacon-packet
      send Beacon-packet }
      }
    else
   drop  p }
elseIf p=Beacon-packet {
Update Neighbors-List
Wait (timeOut)
Send p
}
Else drop P
```

Fig. 4.4 Simplified pseudo-code of TRDD Query propagation

Figure 4.5 shows an example taken from a simulation of the query propagation phase where boundary nodes are already known from a pre-processing phase. QR is constructed dynamically and the termination of the query propagation is confirmed. If boundary nodes are already known from the pre-processing phase, the query propagation extends from one sink to the next and stops when the ends of boundary query trees meet. For ideal propagation and homogenous nodes, this would occur around the sink-to-sink midpoints, with the sinks located in the middle of their query trees. If boundary nodes are determined on-the-fly (i.e. piggybacked with the query propagation), propagation moves from one sink to the next in counterclockwise order. The propagation stops when the next sink (or a node belongs to it) is reached. This means that sinks are located at the ends of their query trees. In the following, we consider the set-up time implementation of the boundary marking process in addition to *on-fly* local maintenance where there is no need for periodic boundary repairing. This means that if any boundary node discovers a disconnection while the query is propagated, it tries to glue the boundary by finding the nearest boundary node so the propagation continuous.

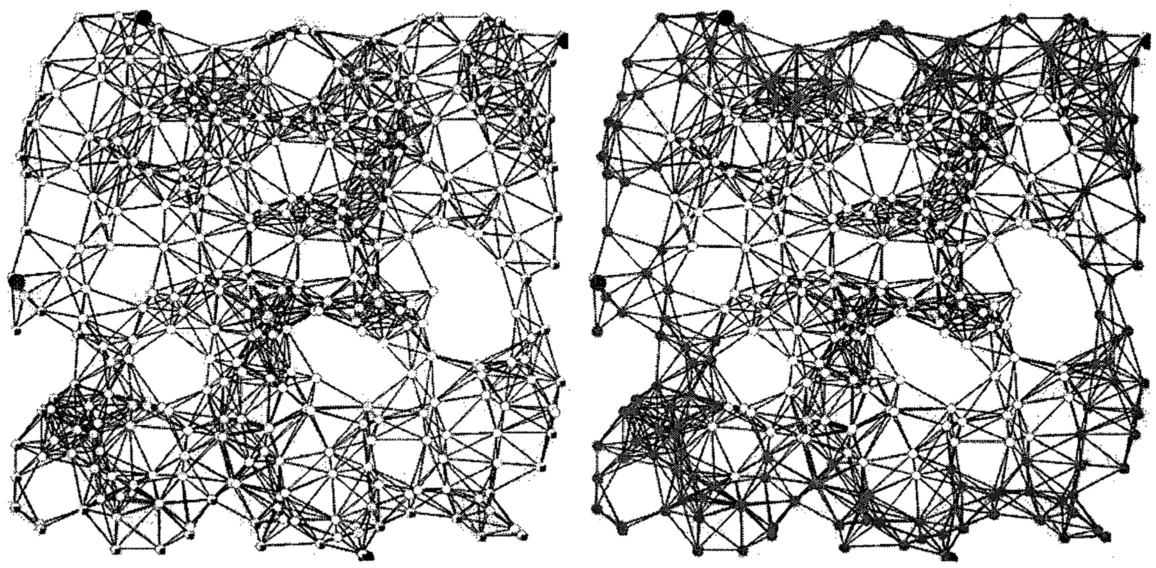


Fig. 4.5. Query Region (Simulation snapshot). Network topology with 320 nodes randomly distributed in a 400 * 400 area with communication radius=50 (Left) 56 boundary nodes are marked, (Right) a total of 164 nodes received the query (solid circles represent the QR nodes) after it was transmitted by boundary nodes.

4.3.3.3 Events Propagation

This phase takes place by establishing local multi-hop paths that bridges to the QR and carry the events. As the QR is distributed over the whole network perimeter and surrounds the network interior, event paths that progress in any direction would eventually intersect the QR.

We generalize the concept of centroid based forwarding strategy defined in previous chapter to add more flexibility to the events forwarding scheme. We first define what we call a direction group DG. This DG is a pair of directions that belongs to the basic direction set (U: up , D: down, L: left, R: right). The directional flow of an event path is then represented by a *direction-group (direction*$_1$*, direction*$_2$*). Direction*$_1$ is the primary direction of the data flow. *Direction*$_2$ *is* a secondary direction used when no progress can be achieved in the primary direction. When the primary direction is one of the (Up, Down), the secondary direction should be one of the (Left, Right) directions and visa versa. So the above four basic directions are represented by eight direction-groups: *{(L, U), (L, D), (R, U), (R, D), (U, L), (U, R), (D, L), (D, R)}*. For example, both *(L, U) and (L, D)* represent the Left direction. For each node, the four basic directions are represented by four abstract reference points on its communication circumference (the circle with the node as the center and the communication radius as the radius).

4.3.3.4 Policies for selecting the direction of flow

In addition to the *away-from-centroid* strategy described in previous chapter, nodes could select the direction of flow either: *Random-based or, Round Robin-based.* In *Random*, each node randomly selects one of the eight direction-groups. *Round Robin* starts with a random selection but then moves to the next in order.

While the Random and Round Robin policies guarantee intersection with the QR, they cannot guarantee that the event path is directed toward the nearest point in the QR. The *v-centroid* acts as a partitioning point that divides the network into four sections. It spreads the traffic away and towards the QR. The advantage of the v-centroid-based selection over the other ones is to reduce the likelihood of selecting the worst direction of flow, where events would propagate with a number of hops equal to the network diameter *D*. The worst case would be half the diameter of the network for those nodes located near the v-centroid. Travel cost in number of hops is inversely proportional to the location of the node with respect to the v-centroid (ie to half of the network diameter). On the other hand the other policies add more flexibility and the travel cost tends to be comparable to that of the centroid-based. For example based on round robin policy, the node consumes total of 2*D* for four events propagation paths so each event path would consume on average half of the network diameter. Fig. 4.6 provides a conceptual example of event propagation path while fig. 4.7 provides a snap-shot of a simulation run.

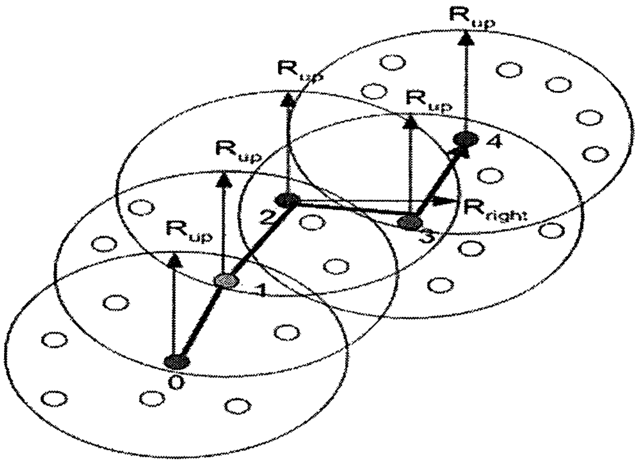


Fig. 4.6. Conceptual example of event propagation path based on direction-group of flow (Up, Right). Primary direction is represented by reference point R_{up}. Node 2 has no neighbors in the Up direction so it selects Node 3 because it is the closest neighbor to the R_{Right}.

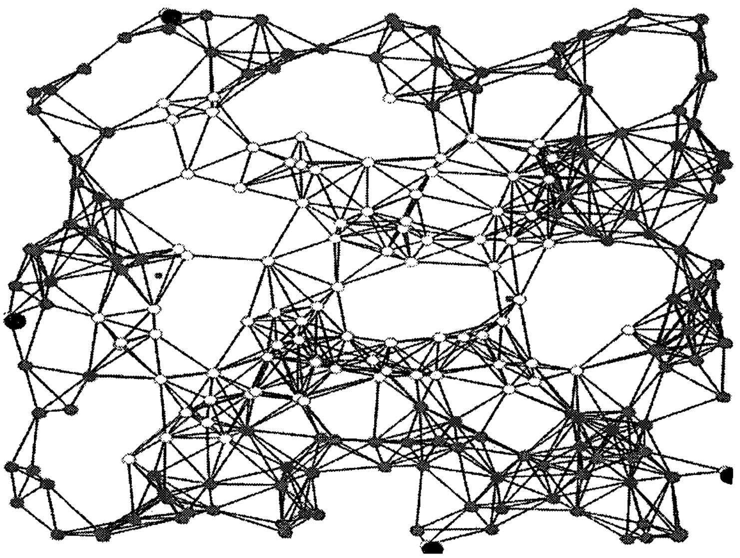


Fig. 4.7. Event propagation paths (simulation snapshot) : the arrows highlight the direction of the propagation.

Once the direction-group is selected using one of the above policies, the node selects its next forwarder neighbor offline in the primary direction. The next forwarder neighbor is the neighbor closest to the reference point of the chosen direction (selected from those closer to that point than the node itself). Recall that interior nodes outside QR receive beacon packets from their neighbors when a query is submitted, not the query itself. These beacons reflect the neighborhood status so nodes are already aware of their current active neighbors. If no neighbor satisfies the criteria for selection, the node selects the next forwarder neighbor in the secondary direction. The data is then transmitted to the next forwarder. The process continues until intersection occurs with the QR where the event is transmitted in the reverse path to the root sink only if it is relevant to the available query.

While establishing event propagation paths the following rules are applied: i) Intermediate nodes cannot change the primary direction chosen by the initiator, ii) they can temporarily change the secondary direction to its opposite one if they fail to find a suitable neighbor using the selected direction-group, iii) they should exclude the sender in order to avoid looping back.

The initiating node, the detector of the event, can change both primary and secondary directions to the opposite directions if no neighbor is found. For example, if the initiator selects the *(U/R)* direction-group and fails to find the next-hop neighbor, it changes the direction-group to *(L/D)*. The heuristic of these selection criteria implies that nodes are allowed to search their whole communication neighborhood for the next forwarder while keeping the progress toward the boards. With no additional reliability insurance, nodes drop the packets if they have no neighbors. Techniques such as transmitting in more than one direction (using more than one direction-group), using "explicit acknowledge", or backtracking [157] could further enhance reliability.

This design of the query propagation and the event propagation phases enhances both autonomy and scalability. Interior nodes do not know where the QR is located, yet the event propagation phase is implicitly forwarding the data in greedy mechanism to the QR. On the other hand, the QR is constructed in parallel and result in even distribution of the query over the network perimeter. This even distribution allows balanced events propagation as the traffic spreads through the network.

4.4 Dynamic Query-based Adaptive Behavior

The key idea is to classify the query categories and change the *behavior* of the protocol based on the query category. Five categories of the query in WSN have been defined in the literature: continuous, snap-shots, historical, event-trigger, and range [10]. Table 4.1 lists the query categories and example of each category.

Table 4.1 Query Categories and Examples

Query Type	**Example**
1. Historical Queries	Retrieve the average value of X in Jan. 2005?
2. Snapshot Queries.	Retrieve the *current* reading X?
3. Continuous Update Queries	Retrieve the minimum value of X during the night, *repeat* every 15 minutes.
4. Cascaded Queries	Retrieve X, If X= $value_1$ then retrieve Y , if Y = $value_2$ then *execute event e*.
5. Multidimensional Range Queries	What are the positions of sensors with X between $value_1$ to $value_2$ and Y between $value_1$ and $value_2$

The above categories in fact could be generally represented by three points in the time space: past (historical), present (snapshot), and future (continuous). The other two categories, cascaded quires and multidimensional range queries, are compound quires that could be further divided into the basic categories. For example, the multidimensional range quires could be repeated as

"*What are the positions of sensors with X between* $value_1$ *to* $value_2$ *(in past, now, in future) and Y between* $value_1$ *and* $value_2$ *(in past, now, in future)?*"

Snapshot quires are those of interest in the current image of the network. Once the responses are retrieved no further action are required by nodes. On the other hand other types of quires could require more query processing upon receiving the partial responses. For example, multi-resolution quires are kind of cascaded quires which require further details or different access rate of the replies. The application could query a phenomenon by first determine the occurrence of certain event , once determined further data collection rate is required from the nodes discovered that event. Example of multi-resolution query type could be

"Retrieve x from Sensor$_A$*, if x > v1 then retrieve y every 100 sec."*

Instead of transmitting a passive signal to only trigger events propagation, an active signal carry more information about the query could enable dynamic behavior of the proposed dissemination protocol. This would increase the query capabilities of the proposed solution and enable it to support different applications with wide spectrum of query requirements. We suggest a composite query structure CQS that is a query with two main parts: a query descriptor plus a splitter tail. A query descriptor could be attribute based query description or declarative query description which is out of scope of this work. A splitter tail is a Meta data that determine the category of the query. The behavior of the protocol is adapted according to the content of the splitter tail.

4.4.1 Adaptive Dissemination Behavior

The following behavior changes could increase the query capability of the proposed solution to support different type of quires.

a) Nodes could memorize the reverse path from the QR to the source node for further communications.

b) Nodes could aggregate the past readings (within specific time window) and transmit the aggregates with the current readings.

c) Nodes could advertise the aggregates over readings in specific time window.

d) Combination of the above.

Matching the behavior to the query type is based on the content of the splitter tail which should specify the query type. An example of simple adaptive behavior is given in figure 4.8. This adaptive behavior would enhance the query capabilities at the cost of transmitting more few bits (the query tail). For example, 7 types of queries could be defined by only 3 bits. This allows supporting different kind of queries at little cost. As long as the size of the tail is very small compared to the size of the query, the performance of the proposed protocol would not be affected.

```
Look-up( S_tail)
If QT = snapshot
Transmit the current live events
If QT=CasCQ,
Aggregate past reading,
Piggyback to the current reading,
Transmit; memorize the reverse path for further communication.
```

Fig. 4.8 Simple pseudo-code for adaptive behaviour dissemination

4.5 Performance Evaluation

TRDD is a data-dissemination protocol whose goals are to reduce communication cost and to achieve a high data delivery rate while adapting to network irregularities. So the following metrics are considered in evaluating the performance of TRDD: *communication cost, successful delivery rate, robustness,* and *events distribution over the network.* The *total number of transmissions* is used to reflect the communication cost. S*uccessful delivery rate* is the number of successfully reported events at sinks compared to the number of events that should be received; in other words, those relevant to the queries submitted. In addition, the percentage of the events reported at each sink is measured to reflect how event distribution through the network is balanced. This section begins with an approximate analytical model to determine the communication cost *C* of TRDD. Then the simulation process is described. Assumptions are that: $\mathbf{Q}$ is the total number of queries issued during the network lifetime, $\boldsymbol{E_T}$ is the total number of events detected during the network lifetime, $\boldsymbol{E_u}$ is the total number of events detected while no queries are issued (between successive query sessions during the network lifetime), $\boldsymbol{E_{ti}}$ is the number of events (relevant or irrelevant) detected when query q_i is issued, and that $\boldsymbol{E_{qi}}$ is the number of relevant events discovered when query q_i is issued. $\boldsymbol{E_{qi}}$ is subset of $\boldsymbol{E_{ti}}$; at the worst case $\boldsymbol{E_{qi}} = \boldsymbol{E_{ti}}$.

4.5.1 Generic Analysis

The cost of TRDD involves (i) the cost of the query propagation and (ii) the cost of events propagations. Query propagation requires setting up the boundary nodes, which could be done once at the setup time or piggybacked to the query propagation phase. Setting up the boundary using the underlying *Boundhole* only consumes the number of communication units equal to the number of marked boundary nodes since each node computes the next perimeter neighbor locally. The cost would then depend on the shape of the outer boundary of the network. Whether piggybacked to the query propagation phase or performed at the setup time, the cost of constructing the QR is *n'* communication units where *n'* is the number of the outer boundary nodes. The main difference is that when the boundary is marked at the setup time the query propagates away from the sink over the perimeter in both clockwise and counter-clockwise directions. When boundary marking process is piggybacked to the query propagation, query propagates counter clockwise over the perimeter. While this does not affect the cost of query propagation, it affects the length of the return path to the sink. On average, the construction of each query tree over the perimeter consumes *n'/m* communication units where *m* is the number of the deployed sinks.

To determine the cost of events propagation per query session, E_t is divided into two subsets: E' (the number of events discovered by nodes located inside QR) and E'' (the number of events discovered by nodes located outside QR). Assuming that the average return path over the QR consumes *r* communication units, each relevant event belongs to E' consumes *r* communication units to return to a root sink in the query tree at which it is detected. For the subset E'', relevant events consume communication cost that includes the cost of the interior path as well as the cost of the return path through the query tree at which the interior path is intersected. Assuming that events require *l* steps to arrive at QR, each relevant event that belongs to E'' consumes *(l)* + r where $l = a\ D\ (0 \le a \le 1)$ and *D* is the network diameter (the maximum number of hops between any two nodes within the network).

Irrelevant events to the query that belong to E'' consume only the interior cost *l* as these events do not have to return to a sink. Irrelevant events that belong to E' consume no cost. So the cost of TRDD per session can be summarized as following.

$$C_{TRDD} = n' + E_q r + E'' l \qquad (1)$$

In (1), the cost of synchronization/beacon signals is ignored as the total cost is dominated by the propagation of the events and queries. For the on-fly implementation of the *boundary marking,* the average length of the return bath *r* is *n'/2m* as the maximum length is *n'/m*. For set-up time implementation, sinks are located at the middle of their query trees so the average length of the return bath is *n'/4m* as the maximum length is n'/*2m*. Assuming a centroid-based policy and set-up time implementation of the boundary-marking process, $r = n'/4m$ and $l = D/4$. So (1) could translate to (2) which describes the average cost of TRDD:

$$C_{TRDD} = n' + E_q(n'/4m) + E^{''}D/4 \tag{2}$$

If Q is considered, the above is multiplied by Q. Equation (2) implies that the increase in the number of deployed sinks positively enhances the performance and reduces the impact of single point of failure. The worst case would be at $m=1$.

4.5.2 GRID-based Analysis

In this section, TRDD is analytically compared to TTDD [18], Directed Diffusion (DD) [49] and Geographic Hash Table (GHT) [71]. DD exemplifies the pure pull schemes; GHT exemplifies the content-based hybrid approaches and TTDD exemplifies the content-independent hybrid approaches. The same analysis guidelines that are used for grid networks in previous research are considered. Ideal propagation and n homogenous sensor nodes uniformly distributed over a grid-based network topology with one monitoring site (sink/AP) located at one of the corners are assumed. (This is considered to be the worst case of our protocol as $m=1$). Flooding the network with data requires asymptotically a cost of O(n), while data transmitted along a single path from any point in the network to a sink/AP at any one corner costs O($\sqrt{n}$) on average [18],[37],[71].

4.5.2.1 TRDD vs. DD

This section considers an optimized version of DD known as one-pull phase directed diffusion OPP-DD where exploratory messages and periodic refresh messages are ignored. With these assumptions, the communication cost of DD with one monitoring site placed at one of the corners would have a communication cost per each query session approximately equal to $n+ \sqrt{n}\, E_{qi}$. The query dissemination costs n units (DD requires the query to be transmitted to all nodes) while each response (relevant event) costs $\sqrt{n}$ to return to the monitoring site over a single path. Equation (3) represents the total communication cost when Q is considered.

$$C_{DD} = Qn + \sqrt{n}\sum_{i=1}^{Q} E_{qi} \tag{3}$$

The communication cost of our solution is as shown below. As the cost of establishing the QR depends on the shape of the network, this cost approximately equals $4\sqrt{n}$ for the grid network. The average length of the interior event paths would be $\sqrt{n}/4$ and the average return path at the QR is $\sqrt{n}$. So (2) could be rewritten for the GRID as shown in (4). When Q is considered, the total communication cost of TRDD is given by (5).

$$C_{TRDD\text{-}GRID} = 4\sqrt{n} + E_q\sqrt{n} + E''\sqrt{n}/4 \tag{4}$$

$$C_{TRDD\text{-}GRID} = 4Q\sqrt{n} + \sqrt{n}\sum_{i=1}^{Q} E_{qi} + \sqrt{n}/4\sum_{i=1}^{Q} E'' \tag{5}$$

From (3) and (5), as long as $\sum_{i=1}^{Q} E'' \leq 4Q(\sqrt{n}-1)$, TRDD outperforms DD. The expression can be further simplified *as* $E''_a \leq 4\sqrt{n}$, where E''_a is the average number of events discovered outside QR at each query session. To determine the extent to which DD could outperform TRDD, we use the ratio $\frac{C_{TRDD\text{-}GRID}}{C_{DD\text{-}GRID}} = \frac{0.25\,E'' + (E_q + 4)}{\sqrt{n} + E_q}$. The extreme case is when all the nodes n have events that are relevant to the submitted query. In this case $E_q = E_t = n$, $E'' = n - 4(d/2)\sqrt{n}$, where d is the degree of connectivity (the average number of one-hop neighbors). d is divided by 2 for boundary nodes as neighbors are not exist in their whole communication circles. The above ratio translates to $\frac{0.25\,(n - 2d\sqrt{n}) + (n + 4)}{\sqrt{n} + n}$. This tends to be ≤ 1.25. For example, when the network size n= 400, the above ratio is 1.08 for d=5 and 1.1 for d=4. For n=900 the ratio is 1.13 for d=5 and 1.14 for d=4. So for large-scale sensor networks where a small subset of nodes discovers events at each session, TRDD would work very efficiently. Even in a rare extreme case, TRDD is able to compete well.

4.5.2.2 TRDD vs. GHT

The GHT data dissemination protocol is a famous hybrid protocol for sensor networks. Similar to our solution, analysis of the approach (based on Grid analysis), shows that it achieves a communication cost of O ($\sqrt{n}$) [37],[71]. The following equation describes GHT's communication cost:

$$C_{GHT} = Q\sqrt{n} + \sqrt{n}\, E_T + \sqrt{n} \sum_{i=1}^{Q} E_{qi} \tag{6}$$

In (6) the total cost of GHT is assumed to be based on the propagation of the events and queries. Periodic beacon signals needed by GHT are ignored. This equation represents the total cost of GHT's pull and push components: Pushing events to the rendezvous location consumes $E_T\sqrt{n}$. Pulling data requires the query to be submitted and the relevant events to be captured in order to send them back to the data collector node (sink). This consumes $Q\sqrt{n} + \sqrt{n} \sum_{i=1}^{Q} E_{qi}$. E_T could be expressed as in (7), meaning that (6) can be re-written as in (8).

$$E_T = \sum_{i=1}^{Q} (E_{ti}) + E_u \tag{7}$$

$$C_{GHT} = Q\sqrt{n} + (\sum_{i=1}^{Q} E_{ti} + E_u)\sqrt{n} + \sqrt{n} \sum_{i=1}^{Q} E_{qi} \tag{8}$$

From (5) and (8), we can compute the difference $C_{GHT} - C_{TRDD}$ as in (9)

$$\mathrm{C_{GHT}}\text{-}\mathrm{C_{TRDD}} = \sqrt{n}\,(\sum_{i=1}^{Q} \mathrm{E_{ti}} + \mathrm{E_u})\text{-}(3\mathrm{Q}\sqrt{n} + \sqrt{n}/4\sum_{i=1}^{Q} \mathrm{E''}) \tag{9}$$

This difference is always positive as long as ($\sum_{i=1}^{Q}$ E_{ti} + E_u) - 1/4$\sum_{i=1}^{Q}$E" ≥ 3Q. If we consider the average over Q, the above expression reduces to E_{ta} + E_{ua} - 1/4$E_a^{''}$ ≥ 3. If we recall that $E_a^{''} \subset E_{ta}$, the last inquiry is almost always true. Moreover, when $E_T \geq Q\sqrt{n}$, pure pull schemes like DD definitely show a smaller communication cost than GHT [71], While this case turns DD to be superior to GHT, TRDD competes well. TRDD keeps the communication on-demand so in contrast to GHT i), the communication cost of TRDD depends on E_t rather than E_T , ii) the cost of TRDD does not relate the network size to the number of quires submitted during the network lifetime.

4.5.2.3 TRDD vs. TTDD

In this sub-section, it is enough to prove that GHT outperforms TTDD in order to show that TRDD outperforms TTDD. The cost of TTDD as described in [18] involves the cost of three different processes: event announcement, query and data forwarding processes. TTDD builds a virtual Grid over the whole network for each event announcement. Building a virtual grid consumes $\frac{4n}{\sqrt{m}}$ communication units where $\sqrt{m}$ is the number of nodes in the side of a grid cell. It depends on the grid cell size α and the transmission radius r. The sink locally floods the current cell by the query which costs m communication units. The query then travels over the Grid to the source node in $\sqrt{2}\ \sqrt{n}$ units and the response travels back to the sink in $\sqrt{2}\ \sqrt{n}$. So the total cost could be rewritten as in (10).

$$C_{TTDD} = Q\,(\sqrt{2}\ \sqrt{n} + \mathrm{m}) + \frac{4n}{\sqrt{m}}\,E_T + \sqrt{2}\ \sqrt{n}\sum_{i=1}^{Q} E_{qi} \qquad (10)$$

Comparing (10) and (6), each of the three components of (10) is greater than its counterpart in (6). If $\sqrt{m}$ is as small as 1 the second term would be very expensive and would turn the asymptotic cost to O (n). If $\sqrt{m}$ is as large as $\sqrt{n}$ (the upper bound value) the first term again turns the asymptotic cost to O (n). Literature shows that both small and large values of cell size ($\sqrt{m}$) reflect some problems on the performance of the TTDD [18]. We can then conclude that $C_{TTDD} \geq C_{GHT} \geq C_{TRDD.}$ Simulation based results are provided in the next section.

4.5.3 Simulation

In this section, both TTDD and the proposed TRDD are simulated and evaluated using the wireless sensor network simulator (WSNS) described in [28]. Two categories of experiments are conducted: the first is based on the on-fly dialect of the proposed protocol in which boundary nodes are tested while query trees are extended. The second is where boundary nodes are pre-determined at set-up time, and local maintenance is applied on-fly. In all experiments, the direction of the events propagation is selected using the centroid-based policy. To insure consistency, all the results plotted are the average of 5 runs for each experiment.

Table 4.2 Simulation Parameters

Network Terrain Area	400 m × 400 m
Network size	150-to-400 :step 50
Node distribution	Uniformly random
Radio Communication Range	50~70 m
Sensing Range	40 m
Sink(s) distribution	1,2,4 sinks deployed at the peripherals
Energy per transmission	Data/query : 1 unit, announcement (TTDD) 0.5 unit
TTDD-Cell size (for comparison)	100m

Sinks deployed dynamically at positions that satisfy the following characteristics: *min(x), max(x), min(y) and max(y),* where x and y are the Cartesian coordinates. This ensures that the modeled sinks are deployed at the outer boundary of the network. The number of sinks and their deployment positions could be adjusted according to the applications. To simulate TTDD, the cell size of the grid is set to 100m. This setting goes along with the setting in [18].

4.5.3.1 Energy Consumption

As mentioned previously, energy consumption is determined by the number of transmissions required for the dissemination process. Both query dissemination and events propagation (data forwarding), and data announcements (in TTDD) costs are investigated.

4.5.3.1.1 Query Cost

To evaluate the cost of querying the network, experiments are stared with all sinks having the same query to be submitted. Experiments are conducted with 1, 2 and 4 sinks. The network size is gradually increased and the number of transmissions is determined. To simulate the TTDD, a single source node is randomly selected to build the virtual grid of the TTDD with cell size set to 100m. The quay propagation stops when this node is reached or a path to it is found (which is built by another sink). Figure 4.9 shows the query cost (number of transmissions required) for disseminating the query by both of the protocols for different numbers of sinks.

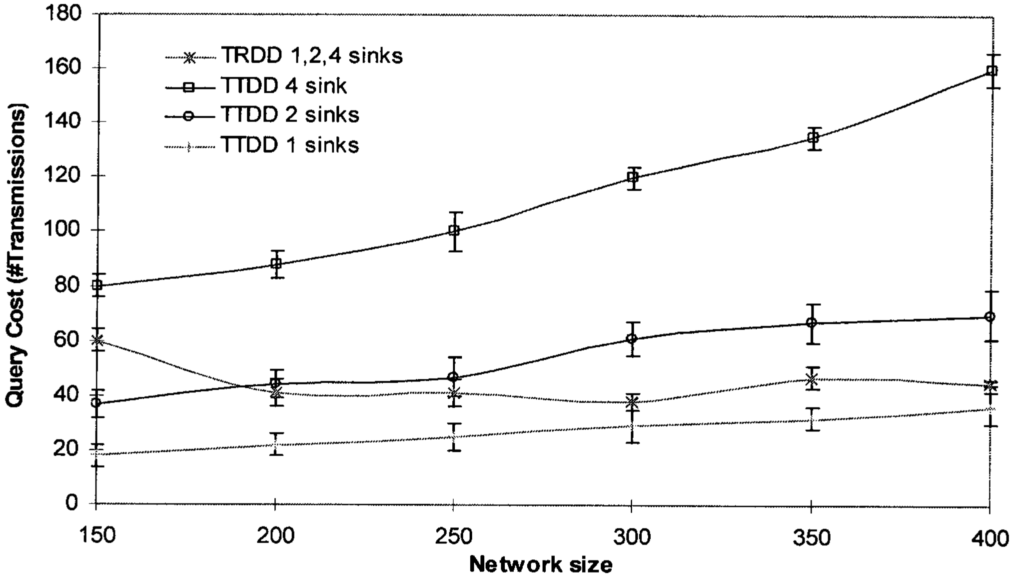


Fig. 4.9. Query cost vs. network size.

The figure shows that the cost of querying the network in case of TRDD does not rely on the number of sinks. For a specific network size, TRDD consumes the same query cost for all the tested number of sinks. This is due to the fact that the query cost depends on the number of perimeter nodes regardless of the number of deployed sinks. On the other hand, TTDD requires each sink to flood the cell where it is located to find a dissemination node. It then builds a query path to the data source. So the increase in number of sinks results in the increase in the query cost of TTDD. The figure also shows that while the increase in the network size conducts a little effect on the query cost of TRDD, the query cost of TTDD increases linearly with the increase in network size. In the case of TRDD, the number of perimeter nodes, and hence the query cost, tends to decrease with the increase in network size. To elaborate more, the ratio of the perimeter size (number of perimeter nodes), as well as the ratio of the QR' size to the network size are investigated.

Figure 4.10 shows the perimeter nodes as well as the nodes reached by the query dissemination phase (QR) as a percentage of the total number of nodes in the network. In

Figure 4.10, the transmission radius is fixed at 70m while the network size is increased. The figure shows that the ratio of the perimeter nodes to the network size is decreased as a result of increasing the network size. Similarly, the number of nodes reached by these perimeter nodes is decreased. This is because the number of interior nodes is relatively increased. An interesting case would be a small network of 150 nodes, where the perimeter size would be around 40% of the network size and the QR would cover 90% of the network.

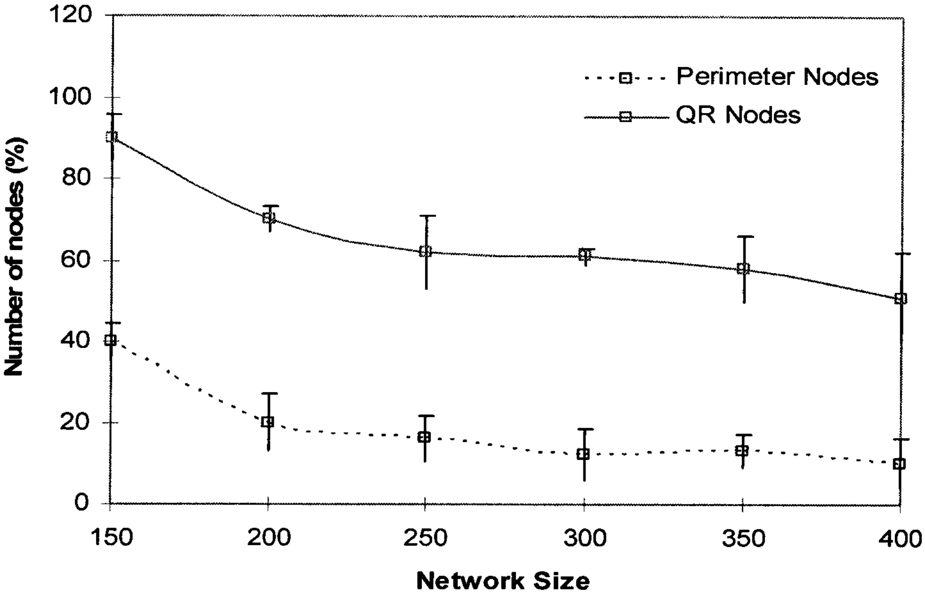


Fig 4.10. Query coverage percentage.

4.5.3.1.2 Data Forwarding Communication Cost

To measure the cost of data forwarding, events are generated as flying objects that move everywhere in the network. When these objects intersect the sensing range of a node, the node becomes a source and generates event packets. In TRDD, event packets are assigned a time of creation and stored locally until a beacon signal arrives (when a query is submitted). In TTDD, events are announced by propagating announcement packets in a form of virtual grid. E_t is increased from 25 to 100 with an increment step = 25. The total number of transmissions is measured for different network sizes (150,250,350). Both single sink and multiple sinks situations are considered (repeatable results are omitted).

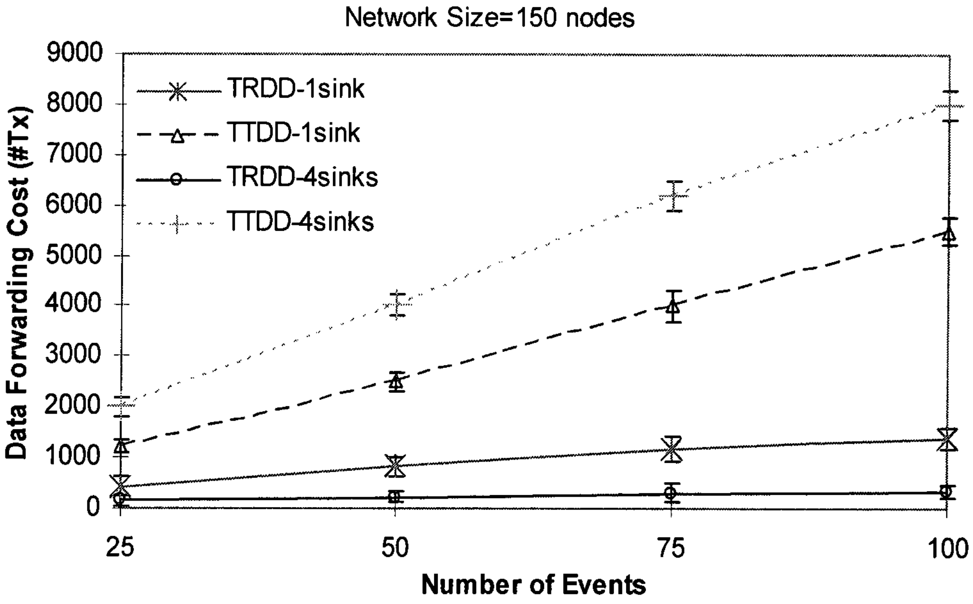


Fig. 4.11. Data Forwarding Communication Cost, Network size=150 nodes.

Figure 4.11 depicts the communication cost of data forwarding for different event sizes when network size is 150 nodes. Figure 4.12 depicts the communication cost of data forwarding for different event sizes when network size is 350 nodes. Results of single sink and four deployed sinks scenario are depicted. For a given network size, the number of transmissions increases with the increase in event size for both protocols. In contrast to TTDD, the increase in network size has little effect on the number of transmissions. This is because both the perimeter return paths and the local forwarding paths exhibit slowly increase with the increase of the network size. Moreover, the slope of the curves that represent TRDD is slowly increased with the increase of the events size. Recall that the QR spans the perimeter nodes and the one hop neighbors of these nodes, events that occur in the QR are transmitted to the sinks through the perimeter paths only. For a specific network size, the increase in the number of sinks results in a noticeable reduction in the number of transmissions. This is because the average number of hops of the return paths to any of these sinks proportionally decreases. This matches with the analysis given in the previous section. The reduction in the communication cost of TRDD is approximately 75% when four sinks are deployed when the network size is 150 nodes. For a network of 350 nodes (Figure 4.12), the average reduction is approximately 80% with four sinks deployed. This is better explained by figure 4.13.

On the other hand, any increase in network size, events size and number of sinks result in a dramatic increase in the communication cost of TTDD. This is because TTDD requires each source to announce its events in the form of a Grid of announcements and to build a data forwarding path (in the reverse direction of the query path from the sink) towards each sink.

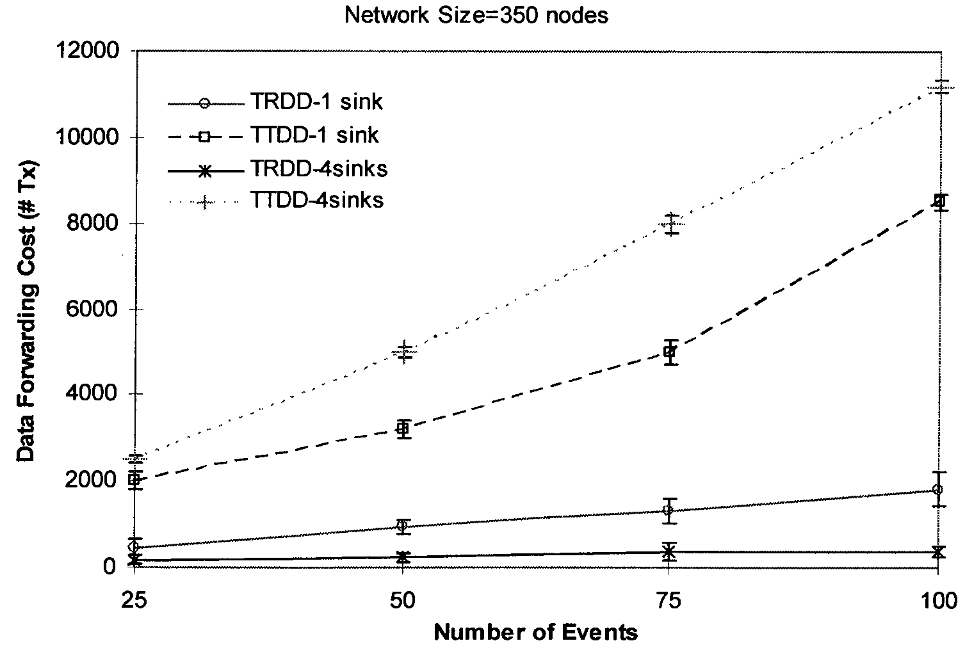


Fig. 4.12. Data Forwarding Communication Cost, Network size=350 nodes.

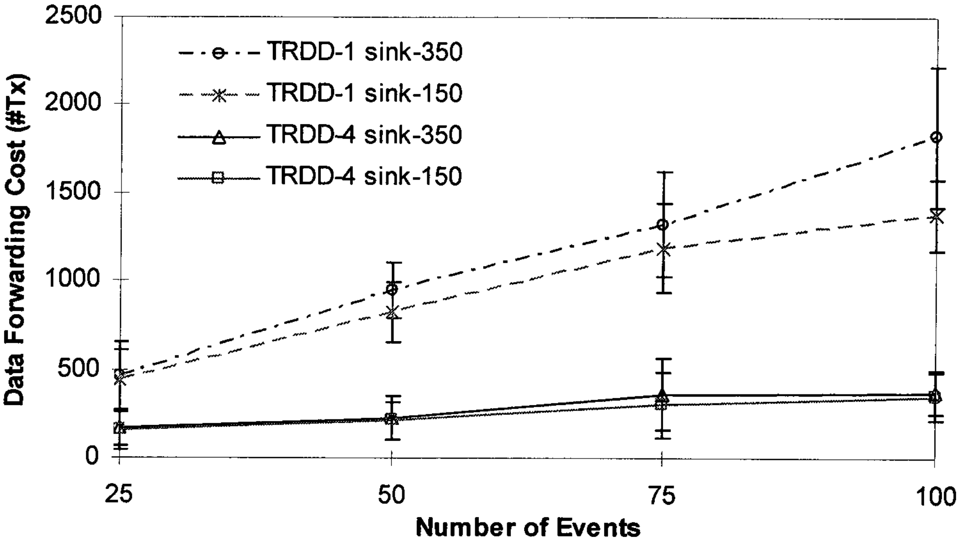


Fig. 4.13. Data Forwarding Communication Cost of TRDD.

4.5.3.2 Impact of sink interior deployment and mobility

To evaluate the performance of TTDD and TRDD in the case of mobile sinks, different mobility patterns have been investigated. We consider that sinks (i) move randomly, (ii) move along the network diagonal, and (iii) move along the network peripheral. The random and the diagonal movement patterns also represent the **interior deployment** of the sink. In this situation, each sink would have to project itself at the network perimeter by establishing local paths just as other nodes do when they have events to propagate. The query would therefore move over the local paths until it intersected with the perimeter as described. The cost of these sink-to-perimeter paths would be added to the normal cost of the query and the events.

We measure the number of transmissions required for the sink to get the interested data. We consider both multiple sources and multiple sinks scenarios. However, the results of the multiple sinks are omitted. They would increase the total overhead in TTDD while decreasing that of TRDD. We consider event generation rate of 0.2/sec and query submission rate of 0.05/sec. The mobility rate is increased from 0.1/sec to 0.3/sec with an increment step 0.5/sec. For the purpose of clarity, we omit repeatable results. The communication overhead is measured during a total simulation time of 120 sec and the results are plotted.

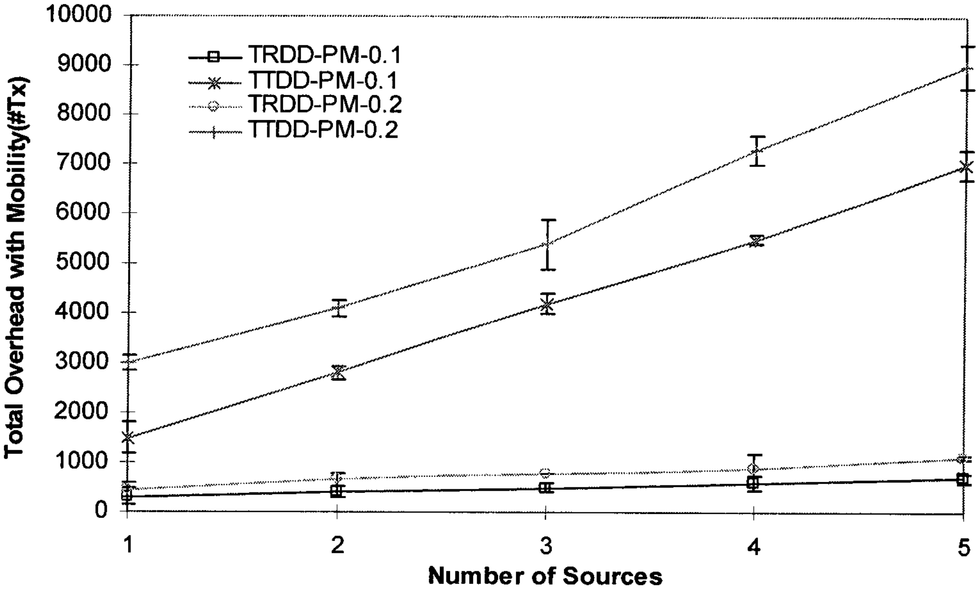


Fig. 4.14. Total communication overhead with peripheral mobility.

Figure 4.14 depicts the results for the peripheral movement scenario. Figure 4.15 depicts the results of the diagonal movement scenario. Figure 4.16 depicts the results of the random movement scenario. The three figures show that TRDD outperforms TTDD. In TRDD, the diagonal movement (DM) and the random movement (RM) incur extra cost in TRDD than the peripheral movement (PM). The reason is that both DM and RM also represent the interior deployment of the sink. In TRDD, the sink only transmits when there is a query to be submitted regardless of the movement rate. In TTDD, the cost of transmitting the events announcements incurs high communication overhead in TTDD regardless of the sink movement. Moreover, the increase in the movement rate increases the total communication overhead of TTDD more than TRDD. This is due to the requirement of the sink to (i) look up for the new dissemination points of the newly announced events by frequent local flooding of its current cell, (ii) flood the cells along its way when moving from cell to cell to acquire information about the current announcements.

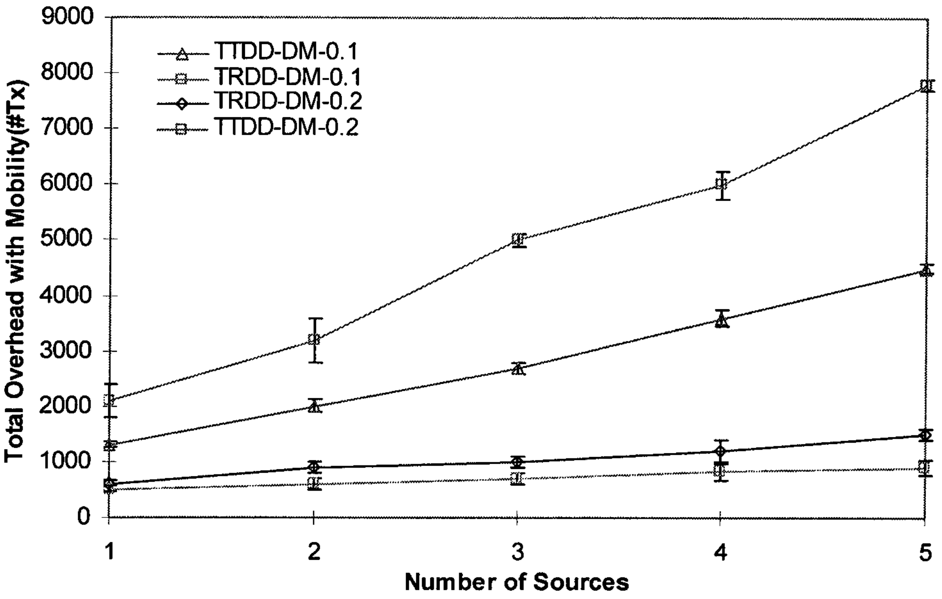


Fig. 4.15. Total communication overhead with diagonal mobility.

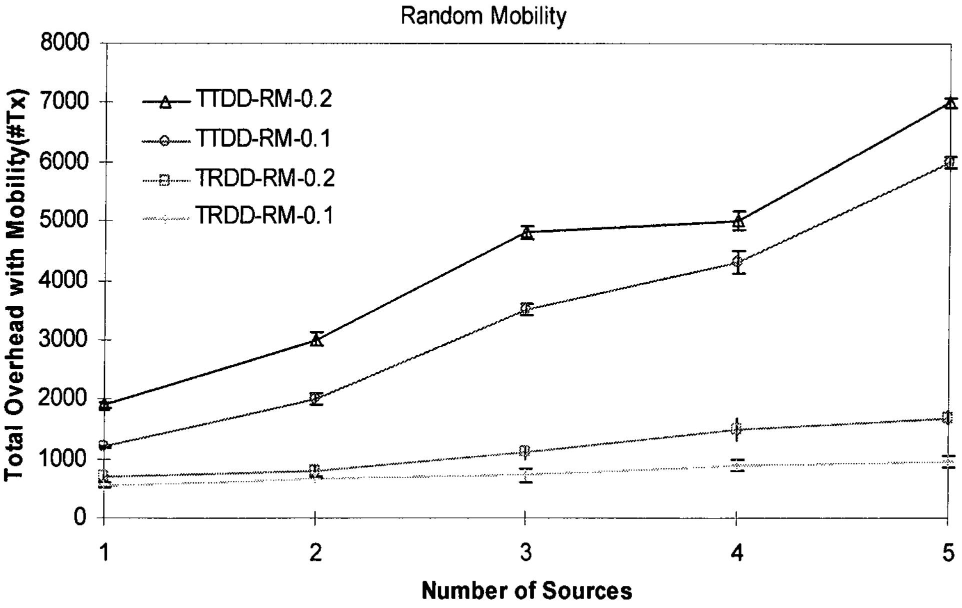


Fig. 4.16. Total communication overhead with random mobility.

Both diagonal and peripheral mobility patterns tested above represent the sink mobility where the existing multiple sinks move randomly and independently. When the deployed sinks have a controlled mobility facility, the improvement of maximizing the network lifetime presented in Chapter 3 is applicable to our rendezvous scheme. This would allow sinks to retain connectivity and maximizes the network lifetime while reducing the communication cost. The longer the network operation time the shorter the communication paths (as the boundary of the network is shrinking and the sinks move closer to the center).

4.5.3.3 Robustness

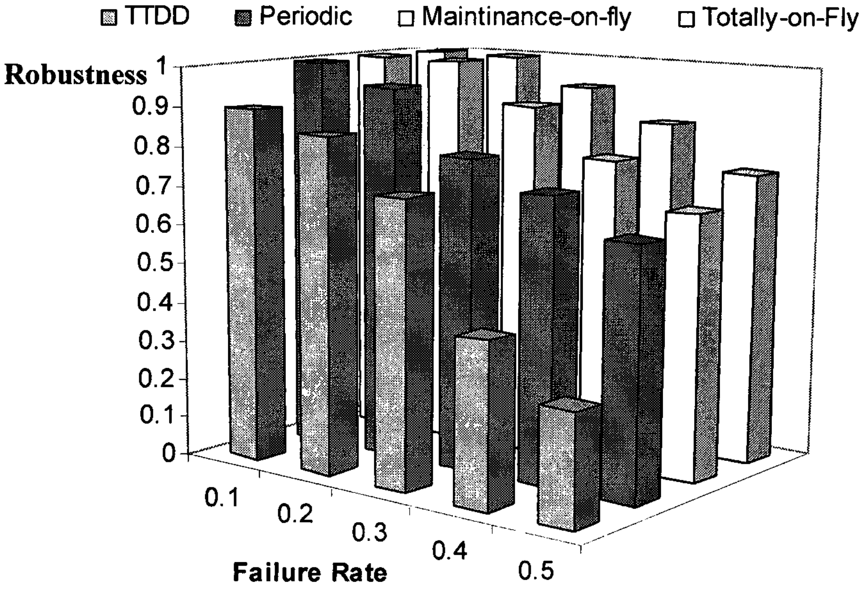


Fig. 4.17. Robustness

The main reasons for frequent topology changes could be the sleep-awake pattern and the permanent failures in nodes (as the sensor nodes are assumed to be stationary). If any sleep-awake control protocol is considered, the assumption is that it cannot hinder the efficient operation of the network. So such underlying mechanisms are not considered to reduce the robustness of our solution. On the other hand, the proposed protocol is reactive in which the

QR and event paths are constructed dynamically over the nodes that are currently alive and in which all computations are localized (based on the information of the one-hop neighbours). So the permanent failure in nodes is considered as the significant factor that could reduce the robustness.

To test the robustness of the protocol against such topology fluctuations, an increasing failure rate (f=0.1 to f=0.5) is modeled and the data dissemination sessions are implemented with three different boundary-marking processes: totally on-fly each time a query is submitted, at the set-up time with periodic local maintenance, and at the set-up time with the on-fly local maintenance. The robustness of the protocol is shown in Figure 4.17. The figure shows that the protocol is highly robust for each process. The on-fly dialect also allows the protocol to achieve higher robustness than with the periodic local maintenance process. This is because the most recent updates in the perimeter are freshly captured. Figure 4.17 also shows that the decrease in the level of robustness of TTDD is much more than that of TRDD with the increase of the failure rate. In the case of TRDD, the QR spans the network perimeter which reduces the impact of node failures. When more than 50% of the maximum network size is turned off, the boundary of the network spans most of the live nodes. The protocol behaves approximately similar to the pull approaches (in that most of the nodes transmit the query). Most of the live nodes are still able to forward their data to the sink. In the case of TTDD, irregularity due to failures of nodes results in the increase in the failure rate. It increases the rate at which the sink fails to find an immediate dissemination node within its current cell. It also increases the rate at which source nodes fail to establish the announcements.

4.5.3.4 Success rate

In the experiments above, all the generated events are assumed to be relevant to the submitted query. To investigate the success rate, two categories of events are simulated. One of these categories is queried. Figure 4.18 shows the percentage of the average successful delivery rate for different numbers of relevant events.

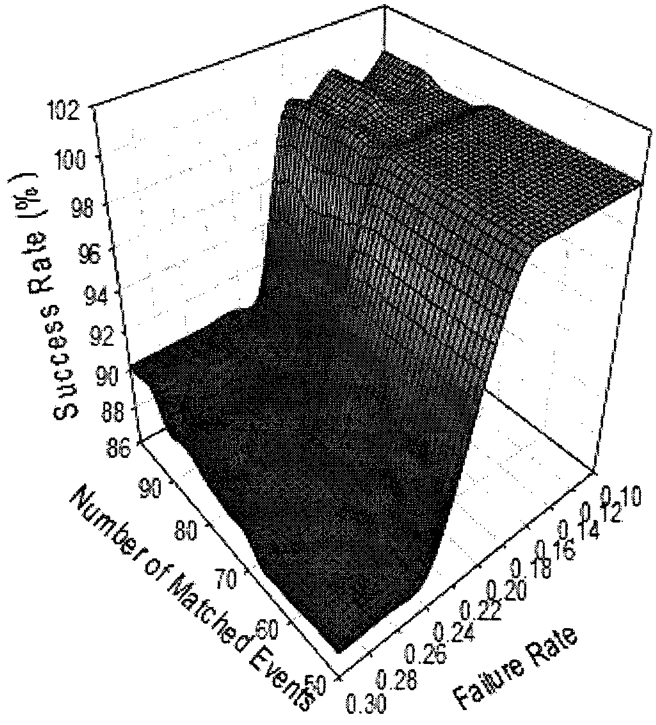


Fig. 4.18. Average Successful Delivery Rate

While E_t is fixed at 100 events, the number of relevant events is increased from 50 to 100 events with an increment of 10 events per step. The curves in Figure 4.18 show the successful delivery rate at different failure rates. The success rate of TRDD is higher than that of TTDD for different event sizes. The figure shows that TRDD achieves 100% success rate at small failure rate (f=0.1). It is also able to survive a high failure rate of f=0.3 with more than 80% success rate. In contrast to TTDD, the increase in the number of relevant events when the failure rate decreases tends to increase the success rate of TRDD. In TRDD, the increase in the relevant data increases the chance of balanced distribution of the data and the existence of the relevant data within and near to the query region. In TTDD, when the number of relevant events increases, managing the data traffic on the two tiers becomes a problem. For example, with 50 relevant events, sources first announce the total 100 events in the form of virtual grids and then sinks establish queries to all sources with the relevant events in order to collect them back utilizing the established grids. This increases the failure rate and decreases the success rate.

Figure 4.19 shows the percentage of the average reported events per sink ($sink_1$, $sink_2$, $sink_3$, $sink_4$) for different event sizes. The figure shows that the distribution of events propagation is balanced through the network as the loads are balanced among the cooperative sinks.

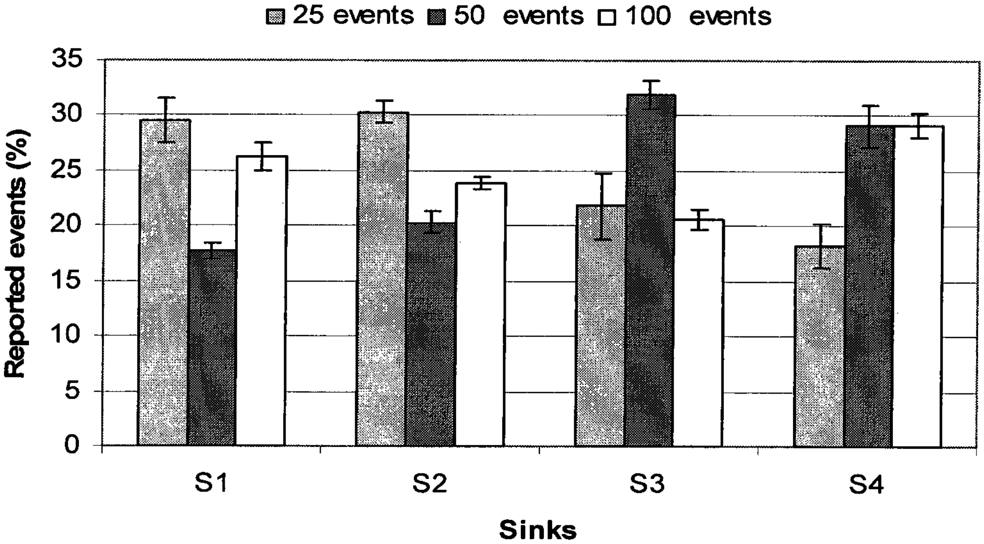


Fig. 4.19. Average Distribution of Reported Events

4.6 Summary

In this chapter, the Topology-based Rendezvous Data Dissemination (TRDD) approach is proposed for data dissemination in tiered sensor networks. The proposed solution efficiently implements a cooperative rendezvous data dissemination strategy in irregular networks that involve multiple sinks. The protocol overcomes the challenges that hinder the wide deployment of hybrid schemes. The protocol dynamically extracts and adapts to the topological features while data is collected. The even construction of the query region over the network perimeter allows the even propagation of events through the network. This, in turn, spreads the traffic and results in a high degree of load balancing. The synchronization of the underlying phases results in efficient on-demand communication that reduces the resource wasting. The proposed solution dramatically reduces the transmission costs and results in efficient operations in large-scale sensor networks. The cooperation between the multiple sinks reduces the impact of failure. Both analysis and simulation show that the robustness of the protocol remains high even at high failure rates.

Chapter 5

Mobility-based Generic Infrastructure for Sensor Network

In sensor networks, there is a requirement for decoupling the underlying communication primitives from the upper layer protocols. This aims to increase the flexibility and ease the development of the designed protocols. This chapter provides an overview of the proposed Layered Infrastructure Protocol (LIP) that exploits mobility to organize an arbitrary network topology into a *physical* co-centric circular layered infrastructure (CLI). This chapter is organized as follows: Section 5.1 introduces the significance of building this generic infrastructure and explains our philosophy towards achieving it; Section 5.2 justifies the assumptions regarding the network model. Section 5.3 describes the design details of the proposed LIP. LIP's correctness and termination are demonstrated in Section 5.4. Section 5.5 discusses some of the implementation issues of the resulting CLI infrastructure. Section 5.6 summarizes the performance of the approach. Section 5.7 provides a brief discussion on how CLI can support upper layer protocols. Finally, section 5.8 summarizes this chapter.

5.1 Introduction

The random deployment process in Wireless Sensor Network WSN results in an arbitrary network graph that is referred to as a network topology [19]. Communication over such a graph is secured via multiple hops. This mode of communication implies that each node is able to play the role of a router, and thus forward messages over multiple hops on behalf of other nodes—the arbitrary topology graph is structured by the communication algorithm [95]. The communication process then takes place through the primitives of the communication algorithm. Most of the current proposed protocols for WSNs are described as cross-layered protocols [20], which reflects the fact that these protocols are in most cases tied up to specific communication algorithm. In turn, the underlying communication algorithm is most optimal when a specific system architecture is employed.

Increasing the reusability of the designed protocols requires decoupling the underlying communication primitives from the upper layer protocols primitives [23],[24]—building a generic infrastructure at the level of physical links is the recent research direction to achieve this goal.

Taking into consideration that sensor networks are application-oriented [10], the problem of building a generic infrastructure is challenging, and is an open field of research. The infrastructure should have the following characteristics: (1) be generic enough to be leveraged by upper layer protocols with equal efficiency, and (2) be flexible enough to support both multi-hop and data mule –like communication regimes.

In this chapter, we propose a novel mobility-based protocol for building such an infrastructure. Mobility in sensor networks is different from the one in systems like MANETs. In sensor networks, mobility is controlled ([95] and [143]), and is usually guided by a goal related to the upper layer applications—it has been shown that mobility provides great flexibility to Static Sensor Networks (SSNs) [94]. In this paper, we uncover a new advantage that mobility could provide to SSNs. We show that augmenting SSNs with a few resource rich mobile nodes simplifies the organization of a generic network infrastructure that could be leveraged by multiple upper level protocols with efficiency and robustness. The philosophy behind our work is that the powerful mobile robots assume the overwhelming role of network organizers, in addition to being the data collectors and management infusers. The proposed Layered Infrastructure Protocol (LIP) allows these mobile robots to discover the existing links and to arrange them into physical co-centric circular layers infrastructure (CLI), where the infrastructure layers correspond to physical rings in the topology. Each layer is assigned one or more robot/sink that acts as a mobile probe; this mobile probe is able to access the data directly (data mule-like), in addition to accessing the data that has been relayed to it via the multi-hop regime. Mobile probes navigate the network and access data through a set of virtual access points that are present in each layer. The infrastructure maintains the proximity of nodes. Nodes that are *neighbours within the infrastructure* are physical neighbous. The protocol builds the infrastructure locally (i.e. based only on one-hop neighbour information). In addition, changes in topology trigger only local updates. With these features, the proposed infrastructure (CLI) adheres to the following design objectives:

(i) **Generic**: efficiently supporting different upper layer protocols (e.g., routing, data collection, data aggregation and broadcasting).

(ii) **Flexible**: efficiently supporting different communication configurations (both multi-hop and data mule-like communication).

(iii) **Maintainable**: failures neither destruct the infrastructure nor hinder the upper layer operations.

(iv) **Complete**: providing logical relationships among nodes without hiding physical relationships.

(v) **Knowledgeable:** the infrastructure basic units (the layers in LIP) should reflect physical topological information in order to ease the access of the network.

Our contributions, as presented in this chapter, can be summarized as following: (i) design of a protocol (layered infrastructure protocol LIP) that constructs an infrastructure (Circular layered infrastructure CLI); that infrastructure adheres to the design objectives, namely being generic, flexible, maintainable, complete and knowledgeable; (ii) design of generic communication primitives that provide multiple communication configuration to the upper layer protocols.; (iii) a customized cost model that predicts the cost of communication over CLI; (iv) validation the performance of CLI in supporting upper layer protocols.

This chapter focuses on the architectural aspects of the proposed infrastructure rather than the application-related aspects; we do, however, delve briefly into the application aspects. We justify the correctness and termination of LIP. We use extensive simulation to highlight the efficiency of the proposed mechanism (by considering realistic conditions of the underlying network settings). Some of the upper layer processes such as routing, data collection and multi-resolution are implemented to show how applications can utilize the underlying infrastructure—comparison to MSC [26] is conducted. The results draws attention to the efficiency of the protocol in supporting upper layer processes, minimizing overhead associated with the infrastructure maintenance, and achieving a high degree of reliability.

5.2 Network Model

This chapter focuses on the deployment scenarios that involve both mobile and static nodes. We consider the case where most nodes are static and only a few powerful nodes are mobile robots/sinks (we refer to these mobile sinks as *mobile probes*).

Mobile Probes: the assumption is that these probes have a communication capability that allows direct communication among them. When the facility of direct communications does not exist, communication between the probes is achieved throughout a centralized management site (end user), if it exists. The initial positions of the probes are invariant to the protocol as they will be redistributed over the terrain area according to the strategy suggested in section (4.3). Controlled mobility facility is assumed at each probe. Each probe travels using a discrete mobility pattern in which the movement is performed in steps. According to [7], such controlled movement is directed toward specific coordinates. Also, movement could follow a straight line and/or a curve pattern.

Sensor nodes: we assume that the network's distribution pattern is based on the uniformly random distribution of sensor nodes within the terrain area [137][158]. We assume that each node knows the coordinates of its location using either GPS or any existing localization techniques [155]. Nodes have a common transmission range T_x. Two nodes are considered to be neighbours if each of them is located in the transmission range of the other.

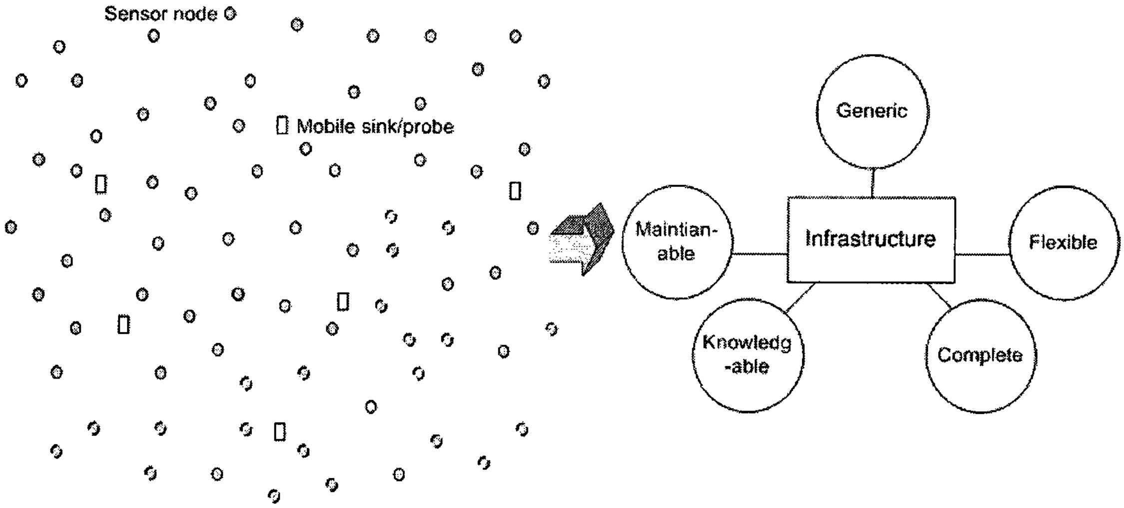


Figure 5.1. Problem Visualization.

Problem statement: having defined the network model, the main problem investigated in this chapter may be formulated as follows: given an arbitrary network topology that adheres to the above network model, construct an infrastructure that adheres to the design objectives–Figure 5.1 visualizes the problem.

5.3 Layered Infrastructure Protocol (LIP)

The proposed Layered Infrastructure Protocol (LIP) organizes network nodes into a co-centric circular layered infrastructure CLI. LIP discovers the physical circular layers that may exist in the network topology. Figure 5.2 depicts a CLI .

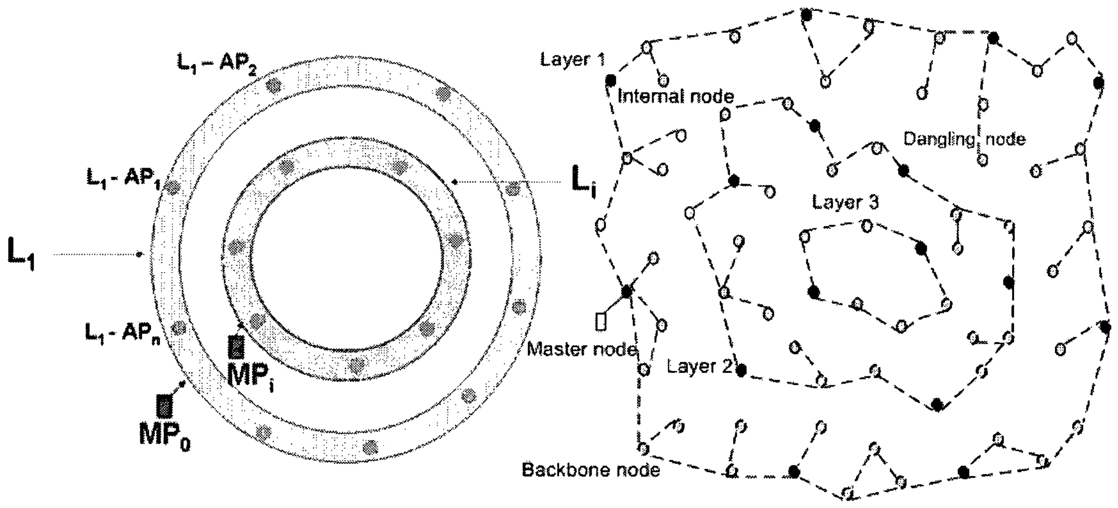


Figure 5.2. The conceptual and the resulting Circular Layered Infrastructure CLI using LIP

To achieve this, LIP elects one of the mobile probes MPs as the master node. This node initiates the execution of the protocol starting at an edge point in the network. The protocol partitions the network into the following supersets: network layers (*L*), back-bones of the layers (*B*), internal nodes of the layers (*I*), access points at each layer (*AP*), and mobile probes (*MP*). Each item in *L* has a corresponding item in each of *B*, *I*, *AP*, and *MP*. A node *n* could belong to one or more sets. A layer L_i in the set *L* consists of a closed sequence of nodes (backbone) B_i and all of their one-hop neighbours I_i. The backbone B_i of the layer forms a physical cycle that starts and ends at an initial node. A subset of the backbone node sequence is selected as the layer access points AP_i. The positions of these access points guide the movement of the probes around the created layers. The probes move to access the data at these positions in addition to accessing the data that hops to them via the multi-hop regime.

LIP constructs the CLI infrastructure by the following three phases: iterative layers extraction, access point selection and mobile probe allocation. Figure 5.3 presents simplified Pseudo-code of the LIP algorithms.

5.3.1 Phase I: Iterative Layer Extraction

In this phase, the nodes are organized to form co-centric layers. At the end of this phase, the network is partitioned into the layer set $L = \{L_1, L_2, .., L_n\}$. Each layer L_i consists of layer-backbone nodes $B_i = \{b_1, b_2, b_3, .. , b_l\}$ and Internal nodes $I_i = \{i_1, i_2, i_3, ..\}$. Layers are extracted iteratively. L_1 is the outmost layer; L_2 is the next layer immediate to L_1 and so on. To extract the outmost layer L_1, the master node initiates a network boundary recognition process and considers the current outer boundary of the network as the layer-backbone B_1. The one-hop neighbours of the nodes in B_1 are the layer internal nodes I_1. Once recognized, all nodes within L_1 go temporarily into sleep-mode. By being off-operation, the new boundary of the network is recognized as the backbone B_2 of layer L_2. In the same way, the one-hop neighbours of nodes in B_2 represent the internal nodes I_2 of L_2. Iteratively, all possible layers could be extracted. This iterative process requires the master node to move iteratively in steps equal to its transmission range towards the centroid of the network. The centroid is calculated using the coordinates of some access points (defined in the next section) over the outmost backbone (that is considered as approximate polygon) using equation (4) in chapter 3. The message transmitted by the master node to recognize the outer boundary of the network carries a signature packet to indicate the current layer number. The first node in the sequence B_i (the initial node) is elected as the leader of the layer L_i. Backbone nodes in set B_i determine their position within the layer L_i. The position of the node in a layer is simply its hop count to the leader of the layer. The pair (*layer_number,* *node_position*) is used as an identification of nodes in B_i. Internal nodes assign themselves the same ID as their parent nodes in the set B_i of their layer in addition to a local identifier that distinguishes each of them. LIP utilizes the *boundhole* algorithm as a seed algorithm to be implemented by the master node to define the nodes in set B of each layer. This algorithm was originally developed to define the holes within the network. It inherently defines the outer boundary of the network as a closed cycle of nodes.

5.3.2 Phase II: Access Point Selection

In this phase, multiple nodes are chosen as access points AP_i at each layer L_i. AP_i is a subset of the backbone nodes B_i. AP_i nodes are selected according to their distance in terms of number of hops to the leader of the layer. For example, nodes at distance kh (k=1, 2, 3, etc... and h is a given parameter) from the leader node are selected as the APs of the layer. The access points' selection process could be implemented either piggybacked to the layer extraction process (as the parameter h that represents the number of hops between successive access points could be carried in the same boundary recognition packet) or separately after the layer extraction phase. Once APs are determined, nodes exchange their topological status. Nodes with no layer membership are considered as dangling nodes (e.g. node d_4 in figure 5.4). These nodes attach themselves to any connected internal neighbour. Access points provide different data access levels within the network. Mobile probes consider these access points as anchors at each layer. Probes explore the network by periodic or customized visits of these points. The number of access points at each layer determines the network access granularity. For example, when APs are chosen such that the number of hops between the successive APs is small, the number of APs would be relatively large, which provides a finer granularity. A greater distance between successive APs provides a coarse granularity.

5.3.3 Phase III : Adaptive Mobile Probe Allocation

Probes should be distributed over the whole terrain area in a way that provides load balancing to the upper layer applications. Each probe is assigned a home layer where it serves as the main probe for data access and network management. Probes use the positional information of the APs at their home layers as anchors over the layers; they visit these anchors either periodically or upon request. The movement starts at the initial position determined by the leader node at each layer. The periodic visiting time t to the access points should be specified by the upper layer application.

Probes are sequentially assigned to their layers. When a layer is extracted, an MP is assigned. Assuming the number of probes is m and the number of discovered layers is l, if m equals l, a one-to-one relationship is established between probes and their home layers. When $m \geq l$, the additional $(m-l)$ probes are considered as *helpers* that should be redistributed over the terrain area to serve layers from outer to inner ones. Special attention should be given to the event when the number of MPs is smaller than the number of layers; in this case, probes could be distributed such that each probe serves l/m layers. Probes could also be actively distributed to apply a more intelligent load balancing rule. As the area of each layer to be served is increased from innermost layers to outmost ones, probes that are assigned the inner layers could help those that have been assigned the outer layers.

```
Simplified LIP Algorithms
LIP-Phase-I: Layer Extraction
1   req :Master $MP_0$ at the edge of the network.
2   i=0, B=$\phi$ , L=$\phi$
3   Do
        a.  $B_i$=Get_$L_i$.backbone(N)
        b.  $L_i$.leader=$MP_0$_Elect_Leader($B_i$ , c)
        c.  If i=1 then c=Centroid($B_i$ )
        d.  $L_i$.Members=Concatenate({$B_i$},{n|n belongs to b.NEG forall b In $B_i$})
        e.  SetMembership($L_i$.Members,$L_i$.leader)
        f.  Go_Sleep($L_i$,setTimeout())
        g.  MP-NextPosition( $MP_0$)=MP_CurrentPosition+r // move one step in the direction
            MP-to-c
        h.  i=i+1
4   Until IsEmpty($L_i$) // no more layers  to be constructed
5   Layerextract.Secure() // MP continue move guided by MP-to-c line , make sure the layers
    are visited in the reverse order
Phase II: (Set_Access_Point)
1.  Pre:  node_counter=hop_count_toLeader // from phase1 step 3-e
2.  Input : $L_i$, h
3.  Initialize k=1
4.  For each $L_i$ in L
        a.  For each $b_i$ in $L_i$.B
        b.  If node_counter =kh Then AP_flag=true
        c.  Increment k
        d.  Next $b_i$
5.  Next $L_i$
6.  for each n $\in$ N
7.  n.Topology_Status.Exchange()
8.  n.LayerMambership.Secure() // dangling nodes attach themselves to any connected
    neighbour
9.  Next n
Phase III:  Allocate_MPs
Input: MP_list, L_list, e
1.  Do
    1.  $MP_i$_InitialLocation=Get_ProbeLocation($L_i$.leader_location, e)
    2.  If not Empty (MP_list) then
    3.  Allocate_MP($MP_i$_InitialLocation)
    4.  Else
    5.  Circulate through MP
2.  Until IsEmbty(L_list)
3.  if IsEmpty(MP-list)=false
4.  Circulate through L_list
```

Figure 5.3. LIP-Algorithms

5.4 Correctness and Termination of LIP

In this section, we provide proof of the correctness of the LIP algorithm, and we show that termination of the algorithm is guaranteed. To formalize the correctness of the algorithm we introduce the following definitions.

Definition 5.1 A **Layer** L_i is a network segment consisting of nodes arranged in a layer backbone B_i and a layer internals I_i.

Definition 5.2 A backbone B_i of layer L_i is a non self-intersecting polygonal loop of nodes $b_0b_1,..,b_0$ that starts and ends at the same node b_0 known as the leader node of the layer.

Definition 5.3 Layer Internals I_i of layer L_i are the one hop neighbours of nodes that belong to B_i of L_i

To prove that LIP is correct we use the following lemmas.

Lemma 5.1: LIP correctly extracts the physical circular rings that exist in the topology.

Proof: The correctness of LIP is based on the correctness of the *Boundhole* algorithm used to determine the backbone of each layer. *Boundhole* recognizes a sequence of nodes $b_0b_1,..,b_0$ that simulates the outmost boundary of the network. LIP considers this sequence as the backbone B_1 of outmost layer L_1. L_1 consists of B_1 and the one hop neighbours of all nodes in B_1. When L_1 temporarily goes to sleep mode during the setup operation (and thus temporarily withdraws from the network), a new boundary is created, which forms the new backbone B_2 of the next layer L_2 and so on. The correctness of *Boundhole* is proved in [88] and [89]. Given this correctness, LIP correctly extracts the physical rings in the topology.

Lemma 5.2: for any node p belonging to N, if p is connected then p should belong to a layer L_i created by LIP.

Proof: We will prove this by contradiction. Assume there is a node p that is connected and does not belong to any layer. If p is connected, it has at least one neighbour q that is connected. But p does not belong to any layer so p is a dangling node. According to LIP, p should attach itself to any one-hop neighbours (q in this case). So p should belong to the same layer that q belongs to. But p does not belong to any layer, which means that p is a disconnected node, thus contradicting the initial assumption that p is connected and does not belong to any layer.

Theorem 5.1 LIP is correct and the union of nodes belongs to the layers created by LIP $\{N(L_1) \cup N(L_2) \cup \ldots \cup N(L_n)\}=\{N-v\}$, where N is the total number of nodes, $N(L_i)$ is the number of nodes belonging to L_i and v is the number of disconnected nodes within the network (those that exist in void with no connected neighbours).

Proof: using lemma 5.1, LIP provides a sequence of layers that sweeps over the topology toward the innermost of the network topology. Using lemma 5.2, if any node cannot join the layers created by the iterative extraction phase, that node joins the layers as a dangling node. This means that the union of nodes in each layer is the total number of nodes in the network, except those that are disconnected i.e. $\{N(L_1) \cup N(L_2) \cup \ldots \cup N(L_n)\}= N-v$. If the assumption that the network is connected holds, v would equal zero and $\{N(L_1) \cup N(L_2) \cup \ldots \cup N(L_n)\}= N$.

To prove that LIP terminates, we need to show that the sequence of layers created by LIP is finite. We start by proving that the algorithm makes progress in each iterative step.

Property 5.1 If $L_i L_{i+1}$ are consecutive layers in the sequence $L_1 L_2 \cdots L_n$, then L_i and L_{i+1} are disjoint.

Lemma 5.3: Property 5.1 is true for $1 \leq i \leq n-1$.

Proof: the layer L_{i+1} consists of the backbone B_{i+1} and the internal nodes I_{i+1}. At the time of its recognition B_{i+1} is the current network boundary. This means that all nodes belonging to L_i must be in sleep/withdraw status and I_{i+1} are one hop neighbours of B_{i+1} existing in the inner direction of the sweeping process. So $L_i \cap L_{i+1}=\Phi$.

Property 5.2 the area of the network topology bounded by L_i *is* greater than the topology area bounded by L_{i+1}.

Lemma 5.4: Property 5.2 is true for $1 \leq i \leq n-1$.

Proof: intuitively, as the direction of sweeping is from the outmost layer to the innermost layer, the topology areas bounded by each layer are shrinking. Geometrically, L_1 is the outmost layer the network, so the area bounded by backbone of layer L_1 is the total network area, $A(L_1)=A$. The area bounded by the backbone of L_2 *is* $A(L_2) = A- A_1$ where A_1 is the area of layer L_1 (area surrounded by the backbone B_1 and backbone B_2 so A $(L_2) < A\ (L_1)$. Similarly, $A\ (L_3) = A- A_1 - A_2 =A\ (L_2)- A_2$ where A_2 is the area of layer L so $A\ (L_3) < A(L_2)$. Accordingly , we can generalize as follows $A(L_{i+1}) = A(L_i)- A_i$ so $A(L_{i+1}) \prec A(L_i)$

Theorem 5.2 LIP terminates and gives a finite sequence of Layers $L_1L_1 \cdots L_n$ that are not overlapping.

Proof: To argue the termination of the algorithm, we only need to prove that the number of the created layers n is finite. Given that properties 5.1 and 5.2 are true, there is a n such that the total area $A=\sum_{i=1}^{n} A_i$, but the topology area A of the network is a finite area, therefore n must be finite.

5.5 CLI Implementation Issues

5.5.1 Topology States

Each node in the network maintains a neighbour table with an entry for every physical neighbour. Figure 5.4 shows a sample of interconnections among layers and figure 5.5 shows samples of neighbourhood information of nodes L_1-i_3 and L_1-b_4. The first row in the tables refers to the node itself. Nodes classify their neighbours according to the services that they provide to them and the layer to which they belong.

Each backbone node b_j in the set B_i of layer L_i keeps information about the upstream b_{j-1} and downstream b_{j+1} backbone neighbours in addition to nodes that are internal neighbours. Backbone nodes should specify which of their internal neighbours lies to the immediate inner layers. Backbone nodes of layer L_{i+1} could have physical neighbours that are internal to the upper layer L_i. Such neighbours are considered foreign internal neighbours. FINs are associated to the upper layer (e.g. the node L_1-i_3 in figure 4 belongs to L_1, node L_2-b_3 considers node L_1-i_3 as foreign internal neighbour).

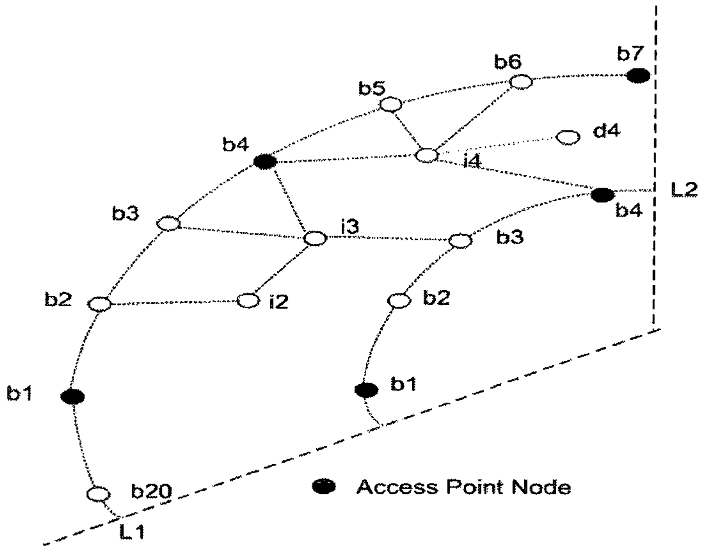


Figure 5.4. Example of interconnections among layers

Definition 5.4 Foreign internal neighbours FINs of L_i are nodes that belong to a layer adjacent to layer L_i and are physical neighbours of nodes in L_i.

On the other hand, internal nodes specify all neighbours that are backbone nodes in the same layer or at different layers in addition to the other internal or dangling neighbours; many situations could be noticed. An internal node at layer L_i could have multiple parent nodes at set B_i. In this case, an internal node selects one of them as the default parent and considers others in case of failure. Internal neighbours could be neighbours to access points at the same layer or at different layers.

Node	Layer	Function
i_3	L_1	Internal(I)/self
b_3	L_1	backbone(B)/parent
i_2	L_1	Internal (I)
b_3	L_2	backbone(B)
b_4	L_1	Backbone/Access Point (AP)

(a)

Node	Layer	Function
b_4	L_1	Access Point (AP/self
b_3	L_1	backbone(B)/parent
b_5	L_1	backbone(B)/next
i_3	L_1	Internal(I)/L_2
i_4	L_1	Internal (I)/L_2

(b)

Figure 5.5. (a) Example of neighbourhood information of node L_1-i_3, (b) The neighbourhood information of node L_1-b_4.

5.5.2 CLI Network Communications

CLI provides a variety of communications that leverage both multi-hop communications and mobility-based communications. Two main communication categories could be defined: node-to-node and node-to-probe. Such communication categories could be single hop or multi-hop according to the positions of the nodes with respect to each other. The destination node/probe could be in the same layer or at different layers. Table 5.1 provides different levels of communication provided by CLI. In-layer node-to-probe provides communication between nodes within a layer and the probe associated with that layer. In-layer node-to-node communication is the communication between two specific end points at the same layer. Layer-to-layer node-to-node is the communication between nodes at different layers.

Table 5.1. CLI network communications

In-layer node-to-probe	In layer node-to-node	Layer-to-Layer *node-to-node*
• Internal node L_i-I_i disseminates the data to its backbone parent neighbor b in the set B_i. • If b is not AP it disseminates the data to the immediate APs through its upstream and downstream backbone neighbors. • Data received at an AP node are stored until the node is visited by a probe.	• The originating node transmits the data to the attached b parent node. • The backbone b node determines whether to send the data over the backbone in a counterclockwise or a clockwise direction according to the position of the destination within the layer. • Assuming n_T is the total number of nodes at layer L_i, the dissemination would be clockwise if $(n_T - \text{destination\_position}) \leq {}^{n_T}\!/_{2}$ and counterclockwise otherwise.	• If the destination is located at the upper layer, the direction of data dissemination is counterclockwise else data is disseminated in a clockwise direction. • The data should exit the originating layer at the first node with a neighbor to the immediate layer and so on until the destination layer is reached. • Within the destination layer, in-layer dissemination is performed.

In addition to the communication levels provided in Table 5.1, layer-to-layer *node-to-probe* acts in a similar way to in-layer node-to-probe with the addition that the local probe transmits the data in a single hop to the destination probe. Since we assume that the mobile probes are able to communicate directly with each other, this level of communication is trivial. When latency is not a main concern [25], node-to-node communication could involve node-to-probe communications (in-layer or layer-to-layer); the originating node sends the data to its immediate access points; probes visiting the access points then carry the data to the destination node. This variety of communication levels raises the degree of flexibility of the proposed infrastructure.

5.5.2.1 Examples of Node-to-node communications

Figure 5.6 shows two communication examples. Example 1: if the backbone node $L_1.b_5$ needs to communicate with $L_1.b_{20}$, $L_1.b_5$ transmits the data to its upstream neighbour in the counter-clockwise direction. Data would be continuously transmitted until b_{20} is reached.

Example 2: let's now assume that $L_1.b_1$ requires communication with $L_2.b_4$. Node $L_1.b_1$ sends the data to its neighbours in a clockwise direction. Data resides in layer L_1 until arrives at the first node with a neighbour to L_2, in this case L_1-b_3. Data arrives at node L_2-b_3 through node L_1-i_3. Within L_2, the direction of dissemination is clockwise according to the position of the destination with respect to the node L_2-b_3.

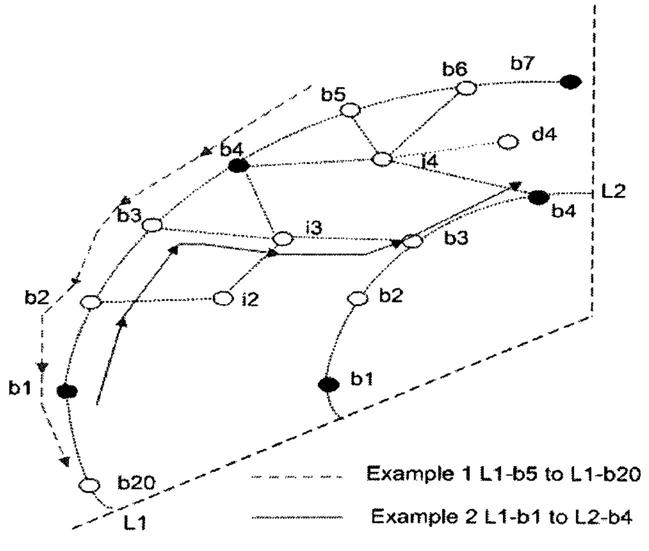


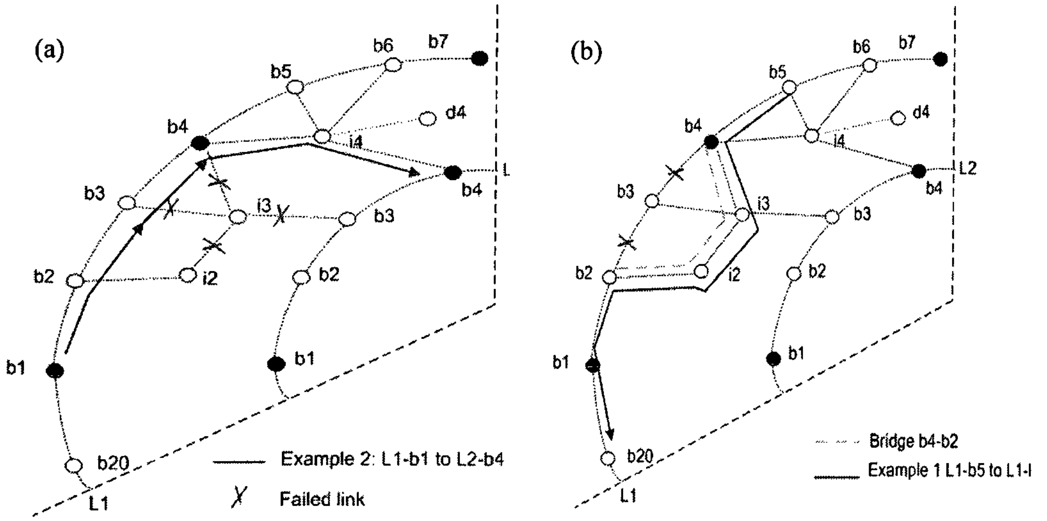
Figure 5.6. Example of node-to-node communications.

5.5.3 Localized failure management

To be effective, topological changes should only result in localized updates. If a link is broken as a result of a node failure, the neighbours of the failed node should be able to cope with this failure without destroying the overall structure. The first remark here is that CLI requires only independent maintenance at each layer. The second remark is that a failure that occurs at an internal node does not affect the overall operation of the infrastructure. To discover a failure, the leader node of each layer periodically transmits a failure discovery packet (sometimes referred to as heart beat). A node considers a neighbour as a failed one when it cannot hear from it. A node then considers the following localized failure handling:

(a) If the failed node is an internal node, its neighbours should mark it as *failed* node and exclude it during the communication. Figure 5.7 repeats communication example 2 when node L_1-i_3 fails.

(b) If the failed node is *backbone* node, its internal neighbours should replace it with another parent node. This replacement of the failed parent is straightforward as a node keeps information about all possible parents. Upstream b_{i-1} and downstream b_{i+1} neighbours of the failed node b_i cope with the failure as follows:



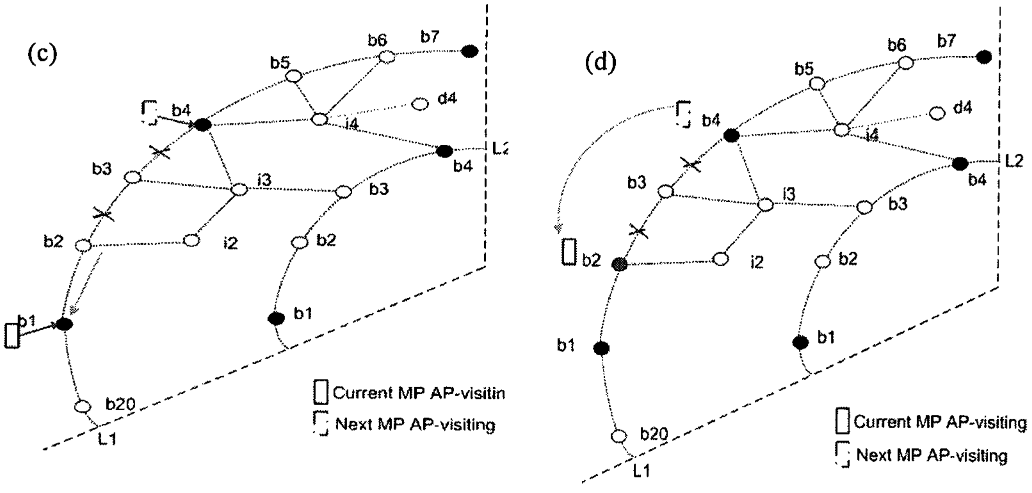


Figure 5.7. Examples of CLI failure handling.

(i) Establish a multi-hop bridge to the successor or the predecessor (according to their position with respect to *b*) of the failed node *b* through their internal neighbours; Figure 5.7-b repeats communication example 1 when node L_1-b_3 fails.

(ii) Consider the failed node as a cut point and compensate the lost communication through the immediate connected access point and the probe of the layer (Figure 5.7-c).

(iii) Send a request to the probes to consider them as new access points if they are not already access points. The new status should be sent to the one-hop neighbours to adjust their neighbourhood information (Figure 5.7-d).

This flexibility of the failure management process supports various configurations; healing a node failure can leverage node-to-node communication (multi-hop bridge) or the layer's moving probe could compensate failure in other scenarios.

5.5.4 New nodes redeployment

A node that is new to the field could simply join the CLI by sending a join request to its one-hop neighbours, collect the topology status of their neighbours and decide upon its own. Once it determines its topology status, it informs its one-hop neighbours. The following rules describe how new fresh node *t* integrates into an existing CLI infrastructure based only on its one-hop neighbour's information:

(i) If *t* has neighbours that are backbone nodes in only one layer L_i, *t* becomes an internal node in L_i. If there are internal neighbours that belong to the layer L_{i-1}, *t* becomes FI in layer L_{i-1}.

(ii) Irrespective of the neighbours that are internal nodes: if *t* has neighbours that are backbone nodes in two different layers L_i, L_{i-1}, then the node *t* connects as an internal node in L_{i-1} and as a foreign internal FI in layer L_i.

(iii) If all *t*'s one-hop neighbours are internal nodes and t has no neighbours that are backbone nodes in any layer, it becomes a dangling node in the layer L_i.

These join rules do not count for the new node becoming a backbone node. In sensor networks, new nodes are not usually expected to join the network individually (the above rules are sufficient in this case). A bulk/supplement of sensor nodes could be redeployed to the field instead. In this case, LIP could be repeated to integrate this bulk of new fresh nodes to the existing infrastructure.

5.6 Performance Evaluations

Partitioning the network into layers (that are separately monitored and managed by mobile probes) results in the following effects: it reduces the communication cost, decreases the effect of node failures, and raises the accessibility of the network. We investigate these aspects in the following sections.

5.6.1 Cost model

In this section, a rough cost model is derived in terms of the number of layers/nodes and access points at layers. The cost model provides information regarding the cost of the communications C over the CLI. The total cost C_t is the sum of the individual costs incurred at each layer. In (5.1), m is the number of the discovered layers.

$$C_t = \sum_{i=1}^{m} C_i \tag{5.1}$$

The outmost layer L_l is the largest layer in the network. The layers gradually shrink toward the network centroid c so that $C_i > C_{i+1}$. Assuming T_x is the transmission range and the nodes are distributed uniformly over the terrain area, the number of layers $m = R_l / T_x$ where R_l is the radius of the outmost layer L_l with respect to the centroid c.

R_l and c need not be known in advance. They are calculated as follows. Considering the outmost layer L_l as a polygon with a set of k access points $\{p_1, p_2, .., p_k\}$ as virtual vertices, c is determined according to the geometric equation (4) in Chapter 3. R_l could then be calculated using (5.2) as the maximum of the distances from c to each of these k points.

$$R_l = Max\| c, \{p_1, p_2, .., p_k\} \| \tag{5.2}$$

The length of the layer is the number of hops touring the boundary of the layer $|B_i|$. Assuming d is the degree of connectivity that represents the average number of the one-hop neighbours, so the number of nodes per layer $L_i \leq d\,|B_i|$.

R_i, the radius of layer L_i with respect to c could be approximately calculated using R_l and (5.3) where D_i is the width of each layer.

$$D_i = (R_i - R_{i+1}) \geq \text{the transmission range } T_x \tag{5.3}$$

The number of access points at each layer L_i is $k_i = |B_i|/h$ where h is the number of hops between successive access points within the layer. The total number of access points in the network is $\sum_{i=1}^{m} k_i$. The cost C_i could be rewritten as in (5.4) where f could be customized according to the nature of the required communication task. It could be customized to represent the communication overhead in terms of the number of transmissions/energy consumption induced in the network by a given task and time consumption to achieve this task.

$$C_t = f(\text{m, B, h}) \tag{5.4}$$

For example, the task that requires collecting data from only the access points incurs $\sum_{i=1}^{m} k_i$ transmission units. The time to achieve one complete layer touring round is g $|B_i|$ where g is the time for travelling one-hop (the upper bound of the one-hop distance is the T_x) which depends on the on the mobility speed of the MPs.

5.6.2 Simulation

5.6.2.1 Methodology of evaluation

To prove the efficiency of LIP, we first conducted proof of concepts experiments. The ability of LIP to effectively support upper layer protocols is then compared to MSC approach [26]. MSC is a cluster-based overlay for multi-scale communications for sensor network. It builds a tree of clusters where each node is assigned a global identifier based on the level of the cluster where it resides (called drum). This model is a good candidate to be considered as milestone to compare against, as it simulates both clustering and tree-based overlays that aim to support multi-scale communications in sensor networks. Three categories of experiments that involve upper layer processes are conducted. The experiments involve broadcasting, selective data collection and routing as samples of the upper layer protocols. WSNS [28] is used in all the described experiments. The simulator is a discrete event-driven simulator developed specifically for WSN. It also uses a multithreading mechanism to allow accurate network configuration in addition to attractive visualization ability. Table 5.2 displays the simulator parameters that have been considered during the simulation process.

Table 5.2 Simulation Parameters

Network Terrain Area	400 m × 400 m
Network size	100,200,300,400 nodes
Node distribution	Uniformly random
Radio Communication Range	40m to 100m
Transmit/receive energy dissipation	1: 0.69

The energy model is simplified based on [159] and [160]; it considers measurements of Cabletron Roamabout 802.11 DS NIC [159] and a similar IEEE 802.11 network interface card when operated in ad hoc mode [160]. To show that CLI adheres to the design objectives, communication cost, reliability and lifetime are used as the performance metrics. The communication cost reflects the energy savings (in both cases of mobility and no-mobility) that the infrastructure could provide to the upper layer applications. Reliability is another important factor that determines the ability of the protocols built over the proposed infrastructure to cope with the failures. To simulate failures, a random subset of nodes is selected as failed nodes while the upper process is performed. The lifetime measures the time until the robustness of the infrastructure decreases beyond a threshold. Only samples of the obtained results are presented in this section, and all the results are augmented with error bars that represent {+,-}2 σ_m where σ_m is the standard variance of the mean. This provides a confidence level of 95%.

5.6.2.2 Proof of concepts experiments

Experiment 1: in this experiment, four scenario settings have been tested. The results are depicted in figure 5.8. In test group 1: the network size is changed from 100 to 400 nodes with an increment step equal to 50 nodes. The nodes are randomly deployed to construct a connected network graph where every node has a single or multi-hop path to every other node in the network [158]. After LIP is applied, the percentage of the nodes that join the layers is measured (by testing the layer-membership), and the results show that 100% of the nodes successfully join the created layers. In test group 2, the network is initially deployed as a connected network, and 5% of the nodes are selected randomly from the pool of the total nodes. The edges to all one-hop neighbours of these nodes are removed from the network topology. LIP is then applied and the average percentages of nodes that successfully join the layers are determined. The results show that 95% of the nodes join the layers created by LIP with confidence level (95%). This means that all the nodes except those that are intentionally made disconnected are able to join the layers created by LIP. In test groups 3 and 4: the network is deployed randomly to create dense networks but not necessarily connected. The transmission range is set to 60m and 90m respectively so that the average number of the one-hop neighbour increases. The results show that more than 95% (Tx=60m) and 97% (Tx=90m) of the nodes are able to join the layers. For a specific Tx, the increase in network size increases the rate of the nodes that able to join the layers.

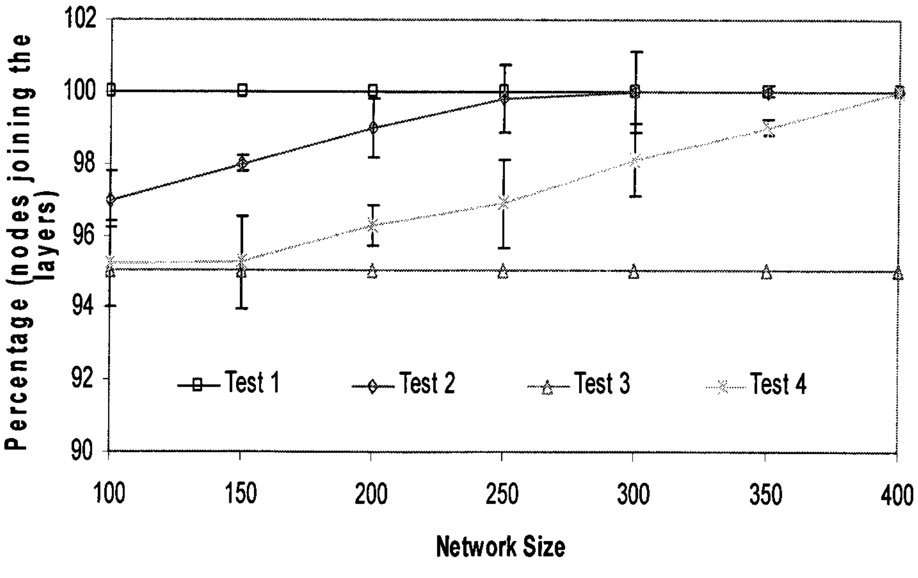


Figure 5.8. Rate of nodes joining the layer

Experiment 2: In this experiment, LIP is implemented both centralized and distributed. The centralized implementation assumes the global topological information exists at the master unit. This unit utilizes the global information to implement the three phases of LIP in a centralized fashion (this is the ideal case of LIP implementation). The distributed implementation is the real implantation of LIP as described earlier where nodes determine their topological status based only on the one-hop neighbourhood information that exists at each node. The maximum number of layers created by LIP for both centralized (highlighted by letter C) and distributed implementations is depicted in figure 5.9 for network size 300 and 400. Different communication radii are tested. The figure shows that the number of created layers is identical for both implementations. Moreover, the number of layers created by LIP depends on the communication radius rather than the network size. The increase in transmission radius increases the width of the created layers and reduces the number of the layers. This goes along with the analysis provided in previous section.

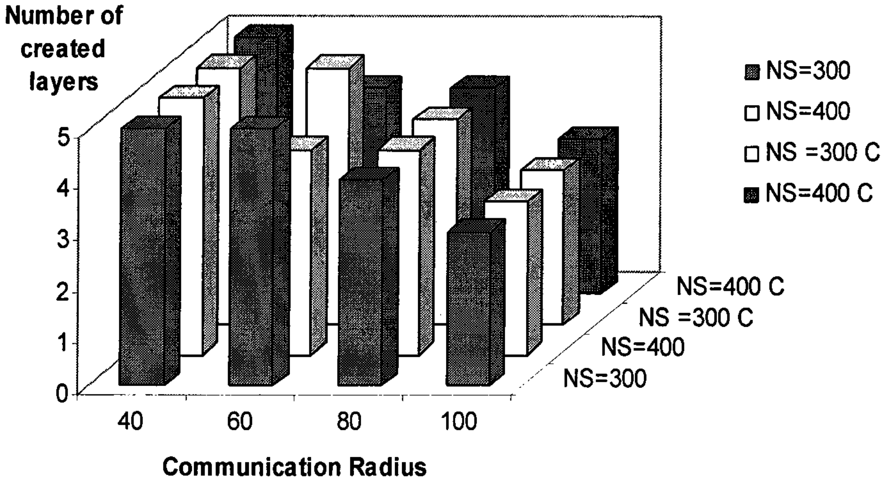


Figure 5.9. Number of created layers for different topology settings.

5.6.2.3 Performance Measurement Experiments

To measure the performance of LIP in comparison to MSC, three different tasks have been tested. These involve routing, selective data collection and broadcasting. Both of the tuneable parameters (h, the number of hops between successive access points in LIP and D, the ratio of n in MSC) are set to 2.

Scenario 1 (Routing): In this experiment, 10 pairs of (source, destinations) are randomly selected from the pool of nodes. The task is to utilize the node-to-node communication in LIP to route the data from the source to the destination. In MSC the shortest paths between the sources and destinations through the drum heads are utilized. The communication radius is changed from 40m to 100m while the network size is set to 400 nodes. Figure 5.10 shows the average energy expenditure for performing the routing task over both infrastructures. The result is the energy savings that LIP provides in comparison to MSC. With no mobility, LIP approximately consumes half of the energy expended by MSC. When mobility is involved, the average energy consumption of LIP is 0.16 of that of MSC.

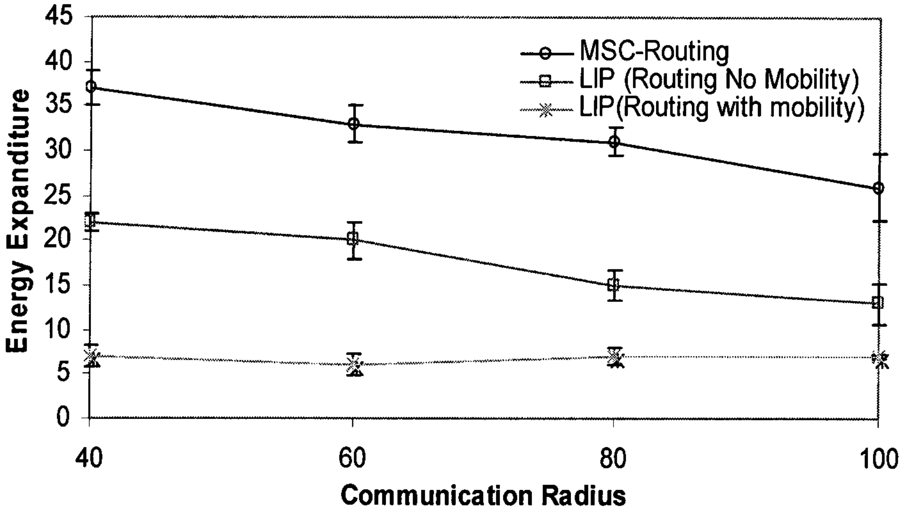


Figure 5.10. Communication cost vs. transmission radius for routing.

Scenario 2: In this experiment, 20 nodes are randomly selected from the pool of nodes (using their ids). The task is to transmit the data to the data collector nodes/MPs. For MSC, four data collector nodes are randomly picked from the group of cluster heads. The nodes transmit one packet to each cluster head. In LIP the nodes transmit one data packet to each of the MPs. The average energy expenditure is then measured and the results are depicted in figure 5.11. The figure shows that LIP provides approximately 200% of energy savings when mobility is not enabled. With mobility enabled the energy savings is 800% and the energy consumption tends to be stable with respect to the increase in the communication range. This occurs because LIP enforces the MPs to access the data from each layer independently. The figure also shows that the reduction in the energy expenditure of LIP is faster than that of MSC when the transmission radius increases; this result goes along with that of routing. In fact, SDC could be seen as a group of simultaneous routing tasks.

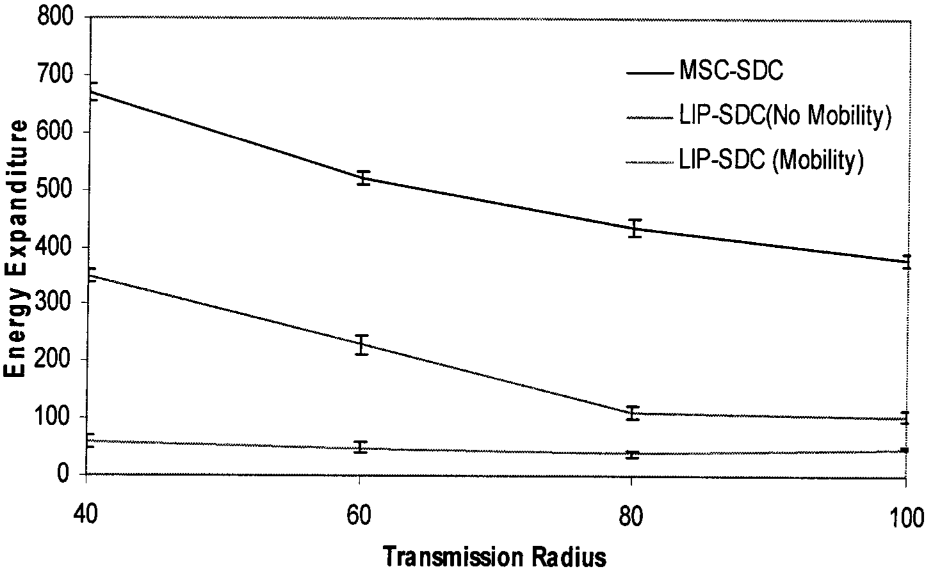


Figure 5.11. Communication cost vs. transmission radius for selective data collection.

Scenario 3: In this experiment, a data packet is required to be broadcasted to all nodes. The initiator node is picked up randomly, and the network size is changed from 100 to 400 nodes. The Tx is set to 60m. Figure 5.12 shows the energy expenditure required to accomplish the broadcasting task. For both protocols to perform broadcasting in LIP, only backbone nodes transmit the data. No mobility is considered.

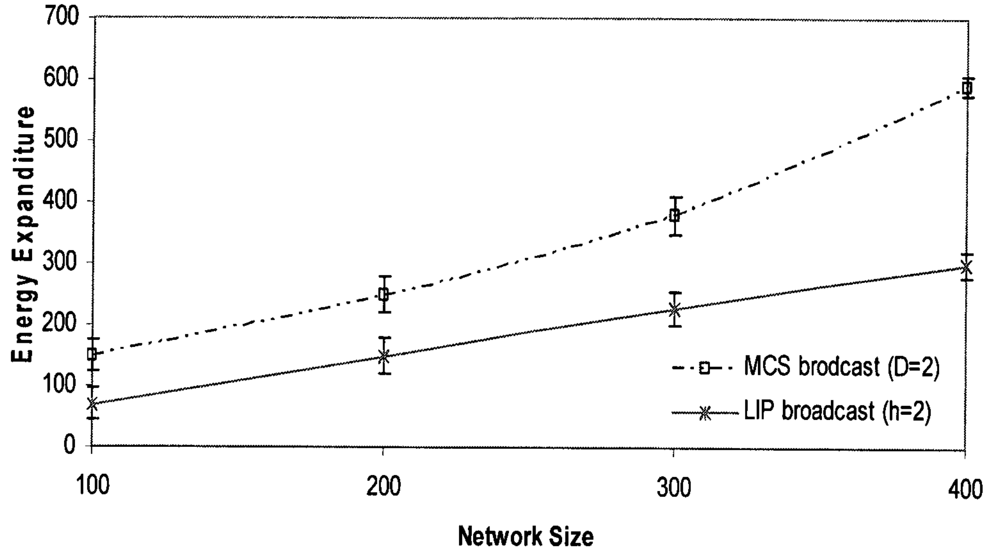


Figure 5.12. Communication cost vs. transmission radius for

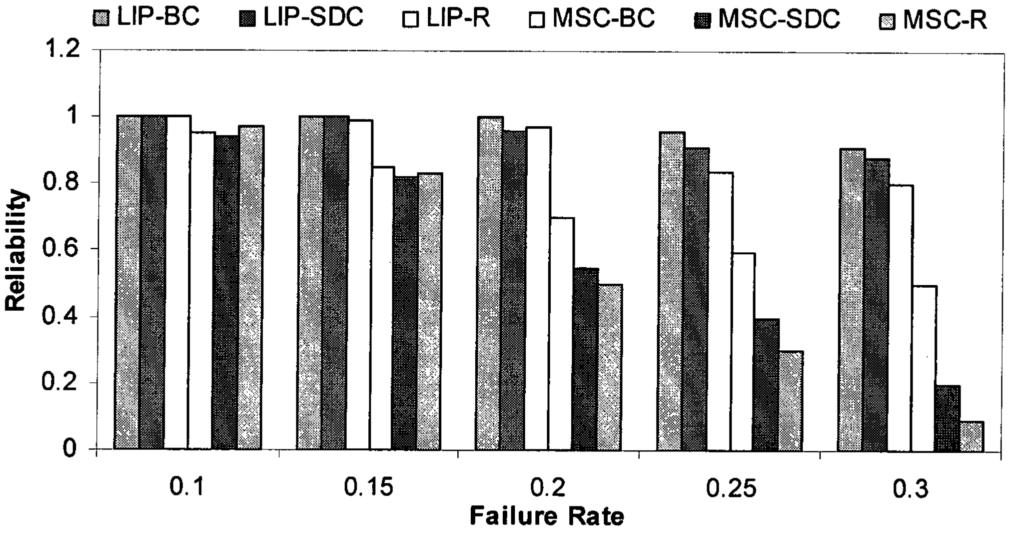


Figure 5.13. Reliability vs. Node Failure.

The figure shows that the energy expenditure increases when the network size increases for both protocols. However, MSC energy expenditure increases faster than LIP with the increase of the network size. This is due to the fact that the clusters are not disjoint and the redundancy in the receiving rate increases with the network size. This redundancy increases the energy consumption as well.

Figure 5.13 shows the average degree of reliability of the implemented upper layer processes: broadcasting, selective data collection and routing. The reliability measures the success rate of both MSC and LIP in performing the required tasks when the rate of failure increases. In figure 5.13, the network size is 400 and T_x is 60m. The tuneable parameters (*h* in LIP and *D* in MSC) are set to 2.The figure shows that CLI provides high reliability to the implemented tasks. The routing process is more sensitive to the increase in failures rate than are selective data collection and broadcasting. While LIP keeps the high reliability level, the reliability of MSC drops quickly when the failure rate reaches 0.2. This is anticipated as MSC builds a tree of clusters. When parent nodes that are drums fail, the nodes in their cells become disconnected, leading to rapid decrease in the MSC reliability level.

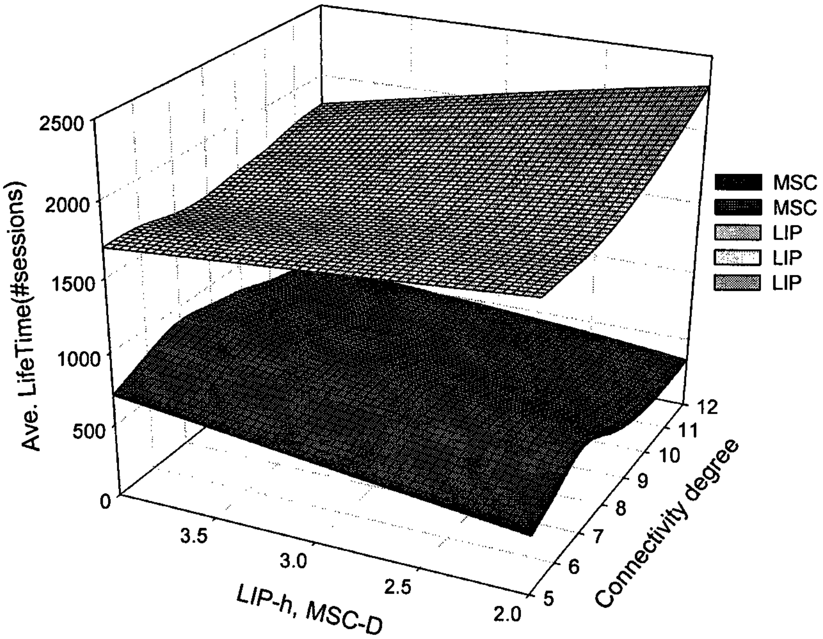


Figure 5.14 Lifetime vs. Degree of connectivity .

Figure 5.14 depicts the lifetime of both protocols for different values of h and D. Lifetime is measured by the number of routing sessions that both LIP and MSC are able to conduct until the failure rate reaches a threshold. According to the reliability results, this threshold is

0.2 (the value of failure rate at which the reliability of MSC drops very quickly). In LIP, when *h* increases the number of hops that data travels to reach the access points increases, and so the energy consumption increases as well. This leads to the slight decrease in the lifetime of LIP. The results also show that the lifetime of LIP is less sensitive to the increase in node connectivity than MSC. LIP is more stable at the low connectivity values. The lifetime of LIP tends to increase when connectivity increases. In MSC, both low and high connectivity values decrease the lifetime. This is due to the requirement of more transmissions at low values. At high values, the receiving rate becomes high due to redundancy of nodes within the cells. This in turn contributes to the decrease of lifetime. Moreover, the increase of *D*, while reducing the hierarchy level of MSC, increases the length of paths (number of hops) between nodes within the drums and between each drum heads. This also leads to the increase of energy consumption and the decrease of the lifetime of MSC.

5.7 Discussion on supporting upper layer applications

CLI has the ability to trade mobility overhead vs. communication overhead (higher number of access points implies that data travels in smaller number of hops to access points, hence to be offloaded to the moving probe). It also exhibits a high degree of resiliency to failure. This makes CLI a rich environment for developing efficient upper layer protocols. CLI could effectively support both multi-hop communications and mobility-based communications. In addition to the ordinary communication applications such as routing, aggregation and data dissemination, the following non-comprehensive list gives information on the vitality of the resulting CLI infrastructure.

Applications based on Adaptive MPs Allocation: MPs could be allocated based on the expected workload. Dynamic allocation to a critical region (region of interest) allows MPs to perform the required task simultaneously at the APs located at this region (leverages parallel task execution). The critical regions are application-dependent. For example, they could be regions with nodes that exhibit higher data traffic or where certain critical events occur. Such regions could also exhibit loss in coverage. MPs could then be allocated with the aim to increase the connectivity and to cope with the loss in coverage.

Scheduling based applications: layers in CLI are managed separately by the associated MPs. Applications that require a sort of scheduling mechanism could take advantage of CLI duty activities. For example, the whole layer or part of a layer could be turned to sleep mode whenever possible to reduce congestion or energy consumption.

Applications based on Multi-granularities communications: different communication levels, ranging from communications via fraction of access points, go through all access points to all the nodes, allowing for efficient multi-resolution/granularities applications to be developed. For example, in applications that require obtaining the network image (current status at each node), CLI provides different granularity levels for obtaining this image. This granularity ranges from a partial network image where the MPs directly offload the status from subset/all of the access points, to a full image where all the nodes transmit their status.

5.7.1 Interface to upper layers applications

We suggest that the interface to the upper layer application be an attribute-based data structure that we call a *communication plan CP*. The aim of the CP is to fulfill the different levels of communications required by different upper layer applications. The plan describes the service requests that should be implemented by the probes. It should indicate (i) the probes that would participate in the processing, (ii) timing information to specify whether the probes should visit the APs periodically or conditionally and (iii) the type of required communication. This plan should be submitted to all the probes and should specify whether *all* or *subset* of the probes is in the plan. It should indicate whether the underlying communication is specific or generic. For example, a routing request requires a specific communication in which the end points of the process should be specified. Other applications, such as data collection where data items are equally collected from all nodes, require generic communication. Some applications target only a sub-topology of the network. So values such as *sub_layer*, *cross-layer* and *multi-layer* could be used to indicate such topological information. All the communications required by the service are translated into LIP communications. The complete design of this plan is left for future work.

5.8 Summary

In this Chapter, a localized LIP protocol (Layered Infrastructure Protocol) for building a generic, efficient infrastructure for sensor networks is described—the proposed protocol builds a physically circular layered infrastructure. The resulting infrastructure CLI (Circular Layered Infrastructure) is able to provide a high degree of reliability to the upper layer applications while reducing the overall energy consumption. The ability of the resulting infrastructure to cope with failures makes it a good candidate for sensor networks; the CLI infrastructure could efficiently support different upper layer applications due to its ability to provide varieties of communication configurations. The sample applications mentioned in this paper provide insight into the flexibility of the proposed infrastructure. This flexibility provides much room for optimization, and the organized layered structure of CLI could also help achieve effective scheduling, which would minimize congestion and boost security.

Chapter 6

Conclusions

In this thesis, we are concerned with the development of efficient communication protocols that aim to maximize the WSN lifetime and maintain the network connectivity. Considering that the crucial task of WSN is the data gathering, we have proposed an efficient data delivery protocol. The philosophy behind our protocol's design is leveraging the positive effect of including sink mobility and multiplicity into WSN. In order to leverage the development of multiple upper layer protocols, we have proposed a localized protocol that decouples the data communication primitives from the communication primitives. Section 6.1 concludes the proposed research work. Section 6.2 discusses the open research issues and the future research directions.

6.1 Research Findings

In this thesis, we contribute research on communication protocols for irregular wireless sensor networks that involve sink mobility and multiplicity. As a result of the work in this thesis, three protocols have been designed, developed, and evaluated.

Firstly, the proposed boundary-peeling data collection protocol and the associated sink-deployment model are able to maximize the lifetime of the network while maintaining a high degree of network connectivity. The protocol shows how sinks could be deployed in order to move and collect data so that the sink-to-network connectivity is highly maintained, the network resources are best utilized, and the energy balancing is ensured. We show that the proposed movement of sinks following the direction of the progressive peeling highly retains the sink-to-network connectivity. The proposed data forwarding strategy balances the load among the nodes and hence leverages the energy balancing. The protocol dynamically adapts to the topological changes in an efficient and dynamic manner with no overhead to exchange topological updates or messages. The protocol is highly reliable. It is able to retain the network connectivity as long as the coverage percentage is tolerated by the application. This allows for better utilization of the network resources. We also show that by constructing a guard region we can customize the longevity of sensor nodes in the core area of interest.

Secondly, the Topology-based Rendezvous Data Dissemination (TRDD) approach is proposed for data dissemination in tiered irregular sensor networks. The protocol overcomes the challenges that hinder the wide deployment of rendezvous schemes. The protocol dynamically extracts and adapts to the topological features while data is collected. The even construction of the query region over the network perimeter allows for the even propagation of events through the network. This, in turn, spreads the traffic and results in a high degree of load balancing. The synchronization of the underlying phases results in efficient on-demand communication that reduces resource wasting. The cooperation between the multiple sinks reduces the impact of failure. This results in a robust mechanism that reduces the communication cost of data gathering and maintains a high degree of reliability against network fluctuations.

Finally, the LIP protocol is proposed for building generic efficient infrastructure for sensor networks. The proposed protocol builds a physically circular layered infrastructure. The resulting infrastructure CLI is able to provide a high degree of reliability to the upper layer applications while reducing the overall energy consumption. The outstanding ability of the resulting infrastructure to cope with failures makes it an excellent candidate for sensor networks. The CLI infrastructure could efficiently support different upper layer applications due to its ability to provide varieties of communication configurations. It has the ability of to trade mobility overhead versus communication overhead. This makes CLI a rich environment for developing efficient upper layer protocols. CLI could effectively support both multi-hop communications and mobility-based communications.

The sample applications mentioned in Chapter 5 provide insight into the flexibility of the proposed infrastructure. This flexibility provides a lot of room for optimization. The organized layered structure of CLI could also help achieve effective scheduling, which would minimize congestion and boost security.

6.2 Open Research Issues

As mentioned in Chapter 2, aggregations mechanisms could further reduce the communication cost of data gathering protocols. Currently, the only aggregation we consider is duplicate suppression (where nodes do not retransmit queries and events packets that they have already transmitted). Our work could further proceed towards the investigation of the impact of involving other aggregation schemes available in the literature.

The concept of adaptive querying behaviour, which is introduced in Chapter 4, promises movement towards an efficient adaptive information discovery system. More investigation and implementation of different query types based on this concept could also be considered.

The CLI resulting from LIP provides a rich environment to support both multi-hop and data mule like communication regimes. Utilizing CLI to develop a variety of communication protocols is a promising future research direction. For example, effective scheduling mechanisms based on CLI could be developed to minimize congestion and boost security. Another promising research area is the design of a complete communication plan that abstracts the different communication requirements of upper layer protocols and utilizes the communication primitives of CLI.

Since all the proposed protocols have been evaluated through simulations and analysis, an extension of our work would be to implement and evaluate the protocols in real world wireless sensor networks.

Bibliography

[1] J. Li, P. Mohapatra, “An analytical model for the energy hole problem in many-to-one sensor networks,” in Proc. of IEEE VTC, September 2005. pp. 2721- 2725.

[2] S. Olariu and I. Stojmenovic, “Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform,” in Proc. of IEEE INFOCOM, Barcelona, Spain, April 2005, pp. 1-12.

[3] J. Park, S. Sahni, “An online heuristic for maximum lifetime routing in wireless sensor networks,” IEEE Trans. Computers. Vol. 55, No. 8, pp. 1048-1056, August 2006.

[4] S. Soro, W. Heinzelman, “Prolonging the lifetime of wireless sensor networks via unequal clustering,” in Proc. of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS), April 2005.

[5] C. F. Li, M. Ye, G. Chen, J. Wu, “An energy-efficient unequal clustering mechanism for wireless sensor networks,” in Proc. of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), Washington, DC, Nov. 2005. 8 pp. -604.

[6] X. Wu, G. Chen, S. Das,”On the energy hole problem of non-uniform node distribution in wireless sensor networks,” in Proceedings of IEEE MASS 2006, Vancouver, BC, October 2006, pp. 180-187.

[7] K. Dantu, M. H. Rahimi, H. Shah, S. Babel, A. Dhariwal, G. S. Sukhatme, “Robomote: enabling mobility in sensor networks,” in IPSN, April 2005, pp. 404- 409.

[8] S. Diggavi, M. Grossglauser, D. N. C. Tse, "Even one-dimensional mobility increases ad hoc wireless capacity," in Proc. IEEE Int. Symp. Information Theory (ISIT), Lausanne, Switzerland, June 2002.

[9] J. Luo, J. Panchard, M. Piorkowski, M. Grossglauser, J. P. Hubaux, "Mobiroute: routing towards a mobile sink for improving lifetime in sensor networks," in IEEE DCOSS, volume LNCS 4026, 2006, pp. 480--497.

[10] J. Heidemann, F. Silva, D. Estrin, "Matching data dissemination algorithms to application requirements," in Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (Sensys '03), Los Angeles, California, November 2003. pp. 218 – 229.

[11] W. Chen, J. C. Hou, "Data Gathering and Fusion in Sensor Networks.," in Handbook of Sensor Networks, (I. Stojmenovic, ed.), Wiley, September 2005, Chapter 15.

[12] J. Burrel, T. Brooke and R. Beckwith, "Vineyard Computing: Sensor Networks in Agriculture Production", Pervasive Computing, IEEE Volume 3, Issue 1, Jan.-March 2004, pp.38 - 45 .

[13] S. Yang. Redwoods go high tech: Researchers use wireless sensors to study California's state tree. *UC Berkekey News* [Online], July 2003. Available: http://www.berkeley.edu/news/media/releases/2003/07/28_redwood.shtml.

[14] A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler. "Wireless sensor networks for habitat monitoring," In ACM Workshop on Sensor Networks and Applications, Atlanta, Georgia, USA, 2002, pp. 88-97.

[15] E. Biagioni and K. Bridges, "The application of remote sensor technology to assist the recovery of rare and endangered species," In Special issue on Distributed Sensor Networks for the International Journal of High Performance Computing Applications, Vol. 16, No. (3), august 2002

[16] I. Stojmenovic, S. Olariu, "Data-centric protocols for wireless sensor networks," in Handbook of Sensor Networks: Algorithms and Architectures (I. Stojmenovic, ed.), Wiley, 2005, pp. 417-456.

[17] X. Zhu, R. Sarkar, J. Gao, "Shape Segmentation and Applications in Sensor Networks," InfoCom, Alaska, USA, May 2007, pp. 1838-1846.

[18] F. Ye, H. Luo, J. Cheng, S. Lu, L. Zhang, "A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks," in MOBICOM'02, Atlanta, Georgia, USA, August 2002, pp. 148 - 159.

[19] I. f. Akyildiz, et al. "Wireless sensor networks: a survey," Computer Networks, Vol. 38, No. 4, March 2002, pp. 393- 422.

[20] Jian Wu, Reliable routing protocols for dynamic wireless ad hoc and sensor networks, Ph.D. thesis, University of Twente, Enschede, the Netherlands. 2007.

[21] A. Abbasi, M. Younis, "A survey on clustering algorithms for wireless sensor networks," Computer Communications, Volume 30, Issue 14-15 , pp. 2826-2841, October 2007.

[22] M. Ali, K. Langendoen, "A case for peer-to-peer network overlays in sensor networks," in Int. Workshop on Wireless Sensor Network Architecture (WWSNA'07), Cambridge, MA, April 24, 2007.

[23] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, I. Stoica, "A unifying link abstraction for wireless sensor networks," in Proceedings of the 3rd international conference on Embedded networked sensor systems, San Diego, California, USA, 2005. pp. 76-89.

[24] S. Olariu, A. Wadaa, L. Wilson, M. Eltoweissy, "Wireless Sensor Networks-Leveraging the Virtual Infrastructure," IEEE Network, August 2004, pp. 51-56.

[25] R. Shah et al., "Data mules: Modeling a Three-Tier Architecture for Sparse Sensor Networks," in Proc. IEEE Workshop Sensor Network Protocols and Applications, 2003. pp. 30-41.

[26] S. Palchaudhuri, R. Kumar, R. Baraniuk, D. Johnson, "Design of adaptive overlays for multi-scale communication in sensor network," in Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS, 2005, pp. 173—190.

[27] Edgar H. Callaway, Wireless Sensor Networks: Architectures and Protocols, CRC Press, Inc., Boca Raton, FL, 2003.

[28] http://www.djstein.com/projects/WirelessSensorNetworkSimulator.html.

[29] M. Tubaishat, S. Madria, "Sensor Networks: an Overview," IEEE Potentials, Vol. 22, No. 2, pp. 20-23, April 2003.

[30] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, "Next century challenges: scalable coordination in sensor networks," in MOBICOM, August 1999, pp. 263—270.

[31] G. Khanna, S. Bagchi, Y. S. Wu, "Fault tolerant energy aware data dissemination protocol in sensor networks," in Dependable Systems and Networks (DSN), Florence, Italy, 2004. pp. 795-804.

[32] B.M. Sadler, "Fundamentals of energy-constrained sensor network systems," in IEEE Aerospace and Electronic Systems Magazine, Tutorial Supplement, Vol. 20, No. 8, pp. 17-35, August 2005.

[33] T. Voigt, H. Ritter , J. Schiller, "Solar-aware routing in wireless sensor networks," in International Workshop on Personal Wireless Communications (PWC), 2003, pp. 847-852.

[34] K. Sohrabi, J. Gao, V. Ailawadhi, G.J. Pottie, "Protocols for self-organization of a wireless sensor network," IEEE Personal Communications, Vol 7, No. 5, pp. 16-27, October 2000.

[35] S. Madden, M.J. Franklin, J.M. Hellerstein,W. Hong, "TAG: a Tiny AGgregation Service for ad-hoc sensor networks," in 5th Symposium on Operating System Design and Implementation (OSDI 2002), December 2002.

[36] P. Bonnet, J. Gehrke, P. Seshadri, "Querying the physical world," IEEE Personal Communications, Vol. 7, No. 5, pp. 10–15, October 2000.

[37] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, D. Estrin, "Data-centric storage in sensornets," ACM SIGCOMM Computer Communication Review , Volume 33 , Issue 1 , pp. 137 - 142 , Jan. 2003.

[38] B. Sheng, Q. Li, W. Mao, "Data storage placement in sensor networks," in Proceedings of the seventh ACM international symposium on mobile ad hoc networking and computing, Florence, Italy , May 2006, pp. 344 – 355.

[39] Y. Yao, J. Gehrke, "Query processing in sensor networks," in Proceedings of the First Biennial Conference on Innovative Data Systems Research (CIDR), 2003.

[40] N. Shrivastava, C. Buragohain, D. Agrawal and S. Suri, "Medians and Beyond: New Aggregation Techniques for Sensor Networks," in Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004), August 2004, pp. 239-249.

[41] Rocky Dunlap, “In-network aggregation in wireless Sensor Networks”, Georgia Institute of Technology, Report. www.cc.gatech.edu/~rocky/docs/rocky_netagg.doc.

[42] T. He, B. M. Blum, J. A. Stankovic, T. Abdelzaher, “AIDA: Adaptive Application-Independent Data Aggregation in Wireless Sensor networks,” ACM Transactions on Embedded Computing Systems, Vol. 3, No .2, pp. 426-457, May 2004.

[43] I. Solis and K. Obraczka, “The Impact of timing in data aggregation for sensor networks,” in IEEE International Conference on Communications, June 2004, pp. 3640-3645 Vol.6

[44] B. Krishnamachari, D. Estrin, S. Wicker, “The impact of data aggregation in wireless sensor networks,” in International Workshop on Distributed Event-Based Systems, (DEBS '02), July 2002, pp. 575- 578.

[45] F. Ye, H. Luo, S. Lu, L. Zhang, “Dissemination Protocols for Large Sensor Networks,” in Wireless Sensor Networks, Kluwer Academic Publishers, Norwell, MA, USA, 2004, pp. 109-128.

[46] Chee-Yee Chong; Kumar, S.P., "Sensor networks: Evolution, opportunities, and challenges," in Proceedings of the IEEE, Volume: 91, No. 8, pp. 1247—1256, August 2003.

[47] S. Tilak, N. Abu-Ghazaleh, W. Heinzelman, “A Taxonomy of Wireless Micro-Sensor Network Models,” ACM Mobile Computing and Communications Review (MC2R), Vol 6, No. 2, pp. 28-36, 2002.

[48] W. R. Heinzelman, J. Kulik, H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” in Proc. MobiCom, Seattle, WA, 1999, pp. 174–185.

[49] C. Intanagonwiwat, R. Govindan, D. Estrin. “Directed Diffusion: a scalable and robust communication paradigm for sensor networks,” in Proceedings of the Sixth Annual International Conference on Mobile Computing and Networks (MobiCOM), August 2000, pp. 56-67.

[50] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks," in Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS '00), January 2000, pp. 3005 - 3014.

[51] M. A. Sharaf, J. Beaver, A. Labrinidis, P. K. Chrysanthis, “TiNA: a scheme for temporal coherency-aware in-network aggregation,” in Proceedings of the 3rd ACM international workshop on data engineering for wireless and mobile access, San Diego, CA, USA , 2003, pp. 69 – 76.

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

[53] N. Sadagopan et al., “The ACQUIRE mechanism for efficient querying in sensor networks,” in the Proceedings of the First International Workshop on Sensor Network Protocol and Applications, Anchorage, Alaska, May 2003, pp. 149-155.

[54] R. C. Shah, J. Rabaey, “Energy aware routing for low energy ad hoc sensor networks," in IEEE Wireless Communications and Networking Conference (WCNC), Orlando, FL. March 17-21, 2002, pp. 350- 355 vol.1.

[55] F. Ye, A. Chen, S. Liu, L. Zhang, “A scalable solution to minimum cost forwarding in large sensor networks," in Proceedings of the tenth International Conference on Computer Communications and Networks (ICCCN), 2001, pp. 304-309.

[56] B. Deb, S. Bhatnagar, B. Nath, “ReInForM: Reliable Information Forwarding using Multiple Paths in Sensor Networks,” in 28th Annual IEEE Conference on Local Computer Networks, 2003, pp. 406- 415.

[57] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks," in Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS '00), January 2000, pp. 3005 - 3014.

[58] T. J. Kwon, Y. Yi, M. Gerla, “Efficient Flooding with Passive Clustering – an Overhead-Free Selective Forward Mechanism for Ad Hoc/ Sensor Networks,” in Proc. of IEEE, 2003. pp. 1210-1220.

[59] M. Gerla, T. J. Kwon; G. Pei,” On-demand routing in large ad hoc wireless networks with passive clustering,” IEEE Wireless Communications and Networking Conference, WCNC , Vol. 1, 2000, pp. 100 – 105.

[60] V. Handziski, A. Köpke , H. Karl, C. Frank, W. Drytkiewicz, “Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering”, in Proc of the 1st European Workshop on Wireless Sensor Networks (EWSN), LNCS 2920, Berlin, Germany, 2004, pp. 172-187.

[61] C. Wan , S. B. Eisenman , A. T. Campbell , J. Crowcroft, “Siphon: overload traffic management using multi-radio virtual sinks in sensor networks”, in Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys’ 05), San Diego, California, USA, November 02-04, 2005, pp. 116-129.

[62] W.K.G. Seah, “Multipath virtual sink architecture for underwater sensor networks,” in Proceedings of the OCEANS2006 Asia Pacific Conference, Singapore, 16–19 May, 2006, pp. 1-6.

[63] S. Giordano , I. Stojmenovic, “Position based routing algorithms for ad hoc networks: A taxonomy,” in Ad Hoc Wireless Networking, X. Cheng, X. Huang and D.Z. Du (eds.), Kluwer, 2004.

[64] Y. Xu, J. Heidemann, D. Estrin, ”Geography-informed Energy Conservation for Ad-hoc Routing," in Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking, 2001, pp. 70-84. 55

[65] F. Kuhn, R. Wattenhofer, A. Zollinger, “Worst-Case optimal and average-case efficient geometric ad-hoc routing," in Proceedings of the 4th ACM International Conference on Mobile Computing and Networking, 2003, pp. 267-278.

[66] I. Stojmenovic, X. Lin,”GEDIR: Loop-Free Location Based Routing in Wireless Networks," in International Conference on Parallel and Distributed Computing and Systems, Boston, MA, USA, Nov. 3-6, 1999.

[67] B. Karp, H. T. Kung, “GPSR: Greedy perimeter stateless routing for wireless sensor networks,” in the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00), Boston, MA, August 2000, pp. 243-254.

[68] Y. Yu, D. Estrin, R. Govindan, “Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks," UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, May 2001.

[69] H. S. Kim, T. F. Abdelzaher, W. H. Kwon, "Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks," in ACM SenSys, 2003. pp. 193-204.

[70] D. Braginsky, D. Estrin, "Rumor routing algorithm for sensor networks," in the First Workshop on Sensor Networks and Applications (WSNA'02), Atlanta, Georgia, USA ,October 2002, pp. 22 - 31.

[71] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, F. Yu., "Data-centric storage in sensornets with GHT, a geographic hash table," Mobile and Networks Application., Vol. 8, No. 4, pp. 427–442, 2003.

[72] X. Li, Y. J. Kim, R. Govidan, and W. Hong, "Multi-dimensional range queries in sensor networks," in Proc. of ACM SenSys, 2003.

[73] Mohamed Aly, Kirk Pruhs, Panos K. Chrysanthis, "KDDCS: a load-balanced in-network data-centric storage scheme for sensor networks," in ACM Fifteenth Conference on Information and Knowledge Management (CIKM2006) November 6-11, 2006. pp. 317-326.

[74] X. Liu, Q. Huang, Y. Zhang, "Combs, needles, haystacks: balancing push and pull for discovery in large-scale sensor networks," in ACM Sensys, Baltimore, MD, USA, November 2004, pp. 122-133.

[75] C. Tian Ee, S. Ratnasamy, S. Shenker, "Practical data-centric storage," in the Third 3rd Symposium on Networked Systems Design and Implementation (NSDI 2006), May 2006.

[76] Q. Fang, J. Gao, L. Guibas, "Landmark-based information brokerage in wireless sensor networks," in 25th Conference of the IEEE Communications Society (InfoCom), Barcelona, Spain, April 2006. pp. 1-12.

[77] Q. Fang, J. Gao, L. J. Guibas, V. de Silva, L. Zhang, "GLIDER: Gradient Landmark-Based Distributed Routing for Sensor Networks," in 24th Conference of the IEEE Communications Society (InfoCom), 2005, pp. 339- 350 vol. 1.

[78] E. B. Hamida, G. Chelius, "A line-based data dissemination protocol for wireless sensor networks with mobile sink," in Proceedings of the IEEE International Conference on Communications (ICC 2008), Beijing, China, May 2008, pp. 2201-2205.

[79] J. H. Shin, J. Kim, K. Park, D. Park, "Railroad: virtual infrastructure for data dissemination in wireless sensor networks," in Proceedings of the 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN'05), Oct. 2005, pp. 168–174.

[80] Roy Friedman, Gabriel Kliot, "Location Services in Wireless Ad Hoc and Hybrid Networks: A Survey," Technical Report TRCS- 2006-10, Technion Computer Science, April 2006.

[81] D. Liu, X. Jia, I. Stojmenovic, "Quorum and connected dominating sets based location service in wireless ad hoc and sensor networks," Computer Communications, vol. 30, pp. 3627–3643, 2007.

[82] I. Stojmenovic, "A routing strategy and quorum based location update scheme for ad hoc wireless networks," Technical Report TR-99-09, SITE, University of Ottawa, September 1999.

[83] D. Liu, I. Stojmenovic, X. Jia, “A scalable quorum based location service in ad hoc and sensor networks,” in Proc. IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Vancouver, BC, Oct. 2006, pp. 489-492.

[84] R. Zhang, H. Zhao, M.A. Labrador, “The Anchor Location Service (ALS) protocol for large-scale wireless sensor networks,” in Proceedings of InterSense 2006, Nice, France, May 2006.

[85] Jeon H., Park K., Hwang D.-J., Choo H., “Sink-oriented Dynamic Location Service Protocol for Mobile Sinks with an Energy Efficient Grid-Based Approach,” Sensors, 9(3):1433-1453, 2009.

[86] Y. H. S. Das, H. Pucha, “Performance comparison of scalable location services for geographic ad hoc routing,” in Proceedings of IEEE INFOCOM 2005, Miami, FL, March 13-17 2005, pp. 1228–1239.

[87] P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, “Routing with guaranteed delivery in ad hoc wireless networks,” in Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Seattle, August 20, 1999, pp. 48–55, ACM/Kluwer Wireless Networks, vol. 7, no. 6, 609–616, November 2001.

[88] Q. Fang and J. Gao and L. Guibas, “Locating and Bypassing Routing Holes in Sensor Networks,” 23rd Conference of the IEEE Communications Society (InfoCom), 2004.

[89] Q. Fang, J. Gao, L. Guibas, “Locating and Bypassing Routing Holes in Sensor Networks,” in 23rd Conference of the IEEE Communications Society (InfoCom), 2004, Mobile Network Application, vol. 11, no. 2, pp. 187-200, April 2006.

[90] Stefan Funke, "Topological hole detection in wireless sensor networks and its applications," in Proceedings of the 2005 joint workshop on Foundations of mobile computing, Germany, 2005. pp: 44 – 53.

[91] A. Kröller, S.P. Fekete, D. Pfisterer, S. Fischer, "Deterministic boundary recognition and topology extraction for large sensor networks," in Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 06), Miami, Florida 2006, pp. 1000-1009.

[92] Y. Wu, X. Chen, L. Zhang, Z. Niu, "Interest Dissemination with Directional Antenna for Wireless Sensor networks with Mobile Sinks," in ACM Sensys, Boulder, Colorado, USA, November 2006. pp. 99-111.

[93] M. Grossglauser, D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," IEEE/ACM Transactions on Networking (TON), vol. 10, no. 4, p.477-486, August 2002.

[94] A. Labrinidis, A. Stefanidis, "Panel on mobility in sensor networks," In Proceedings of the 6th international conference on Mobile data management, Ayia Napa, Cyprus, 2005, pp. 333 – 334.

[95] Y. Tirta, B. Lau, N. Malhotra, S. Bagchi, Z. Li , Y. H. Lu, IEEE Monograph on Sensor Network Operations, in: S. Phoha and T. F. La Porta (eds.), chapter "Controlled Mobility for Efficient Data Gathering in Sensor Networks with Passively Mobile Nodes" IEEE Press and John Wiley and Sons, Inc., Piscataway, NJ (December 2, 2005).

[96] E. Ekici, Y. Gu, D. Bozdag, "Mobility-based communication in wireless sensor networks," IEEE Communications Magazine, 44(7), pp. 56-62, 2006.

[97] M.A. Batalin, G.S. Sukhatme, "Coverage, Exploration, and Deployment by a Mobile Robot and Communication Network," Telecommunication Systems, Volume 26, Numbers 2-4, pp. 181-196, June, 2004.

[98] B. Liu, P. Brass, and O. Dousse, "Mobility Improves Coverage of Sensor Networks," in ACM MobiHoc, Urbana-Champaign, IL, USA , March 2005, pp. 300-308.

[99] G. Wang, G. Cao, T. L. Porta, W. Zhang, "Sensor relocation in mobile sensor networks," in IEEE Infocom, Miami, FL, USA, March 2005, pp. 2302-2312 vol. 4.

[100] O. Younis, M. Krunz, S. Ramasubramanian, "A Framework for Resilient Online Coverage in Sensor Networks," in IEEE Secon, San Diego, CA,, June 2007, pp. 540-549.

[101] A. Howard, M.J. Mataric¢, G.S. Sukhatme, "An Incremental Self-Deployment Algorithm for Mobile Sensor Networks," Autonomous Robots, Special Issue on Intelligent Embedded Systems, vol. 13, no. 2, 113-126 , September2002.

[102] N. Heo, P.K. Varshney, "An intelligent deployment and clustering algorithm for a distributed mobile sensor network," in IEEE International Conference on Systems, Man and Cybernetics, October 2003, pp. 4576- 4581 vol.5.

[103] G. Wang, G. Cao, T.L. Porta, "Movement-Assisted Sensor Deployment," in INFOCOM'04, March 2004, IEEE Transactions on Mobile Computing , vol. 5, no. 6, pp. 640-652, 2006.

[104] Guiling Wang, Guohong Cao, Tom La Porta, "Proxy-Based Sensor Deployment for Mobile Sensor Networks," in IEEE MASS, October 2004, pp. 493- 502.

[105] J. Wu, S. Yang, “SMART: a scan-based movement-assisted sensor deployment method in wireless sensor networks,” in IEEE InfoCom, March 2005, pp. 2313- 2324 vol. 4.

[106] A. Kashyap, S. Khuller, M. Shayman, “Relay Placement for Higher Order Connectivity in Wireless Sensor Networks,” in IEEE InfoCom, Barcelona, Spain April 2006. pp. 1-12.

[107] X. Bai, S. Kuma, D. Xua, Z. Yun, T. H. La, “Deploying wireless sensors to achieve both coverage and connectivity,” in Proceedings of the seventh ACM international symposium on mobile ad hoc networking and computing , Italy, May 2006. pp. 131-142.

[108] S. Das, H. Liu, A. Nayak, I. Stojmenovic, “A localized algorithm for bi-connectivity of connected mobile robots,” in 3rd MSN 2007, Beijing, China, Lecture Notes in Computer Science, Vol. 4864, 12-14 December 2007, pp. 19-31.

[109] P. Basu, J. Redi, “Movement Control Algorithms for Realization of Fault-Tolerant Ad Hoc Robot Networks,” IEEE Networks, vol. 18, no 4, pp. 2-10, July/August 2004.

[110] T. Spyropoulos, K. Psounis, CS Raghavendra, “Performance analysis of mobility-assisted routing,” in Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing , Italy, May 2006. pp. 49-60.

[111] M. Younis, M. Bangad, K. Akkaya, “Base-Station Repositioning For Optimized Performance of Sensor Networks,” in Proc. IEEE Vehicular Technology Conference (VTC-Fall’03), Orlando, Florida, October 2003, pp. 2956- 2960 Vol.5.

[112] A. Chakrabarti, A. Sabharwal, B. Aazhang, “Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks,” in The second International Workshop on Information Processing in Sensor Networks (IPSN), 2003, pp. 129-145.

[113] A. A. Somasundara, A. Ramamoorthy, M. B. Srivastava, “Mobile Element Scheduling for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines,” in Proceedings of the 25th IEEE International Real-Time Systems Symposium, 2004, pp.296 – 305.

[114] A. Kansal et al., “Intelligent Fluid Infrastructure for Embedded Networks,” in Proc. 2nd International Conference Mobile System, Applications, and Services, Boston, MA, USA, 2004, pp. 111-124.

[115] D. Jea, A. A. Somasundara, M. B. Srivastava, “Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks,” in Proc. IEEE/ACM International Conference Distributed Comp., in Sensor Sys. (DCOSS), 2005, pp. 244-257.

[116] Y. Gu et al., “Partitioning-Based Mobile Element Scheduling in Wireless Sensor Networks,” in Proc. IEEE ComSoc Conference Sensor and Ad Hoc Communications and Networks, 2005, pp. 386-395.

[117] I. Athanasios, J. Rolim, “Fast and Energy Efficient Sensor Data Collection by Multiple Mobile Sinks,” in ACM MobiWac 2007, Crete, Greece, October 22, 2007, pp. 25-32.

[118] S. R. Gandham et al., “Energy Efficient Schemes for Wireless Sensor Networks With Multiple Mobile Base Stations,” in Proc. IEEE GlobeCom, 2003, pp. 377-381 (22).

[119] Z. M. Wang et al., “Exploiting Sink Mobility for Maximizing Sensor Networks Lifetime,” in Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005, pp. 287a- 287a.

[120] J. Luo , J. P. Hubaux, “Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks,” in Proc. IEEE InfoCom, 2005, pp. 1735- 1746 vol. 3.

[121] J. English, M. Wiacek, M. Younis, "CORE: Coordinated Relocation of Sink Nodes in Wireless Sensor Networks," in Proc. of the 23rd IEEE Biennial Symposium on Communications (QBSC'06), Kingston, Ontario, Canada, May 2006, pp. 320-323.

[122] K. Akkaya, M. Younis, "Coverage and Latency aware Actor Placement for Wireless Sensor and Actor Networks," International Journal of Sensor Networks, vol. 3, no. 3, pp. 152-164, 2008.

[123] Eiko Yoneki, Jean Bacon Se., "Mobility-Based Communication in Wireless Sensor Networks, A survey of Wireless Sensor Network technologies: research trends and middleware's role" Technical Report UCAM-CL-TR646, University of Cambridge, 2005.

[124] D. Culler, P. Dutta, C. T. Ee, R. Fonseca, J. Hui, P. Levis, J. Polastre, S. Shenker, I. Stoica, G. Tolle, J. Zhao. "Towards a sensor net architecture: Lowering the waistline," In HotOS, Santa Fe, NM, 2005, pp. 24 – 24.

[125] Cheng, R. Fonseca, Kim, Moon, Tavakoli, D. Culler, S. Shenker, I. Stoica, "A modular network layer for sensornets," in OSDI'06, Seattle, WA, USA ,November 6-8, 2006, pp. 249-262.

[126] O. Gnawali, B. Greenstein, K-Y. Jang, A. Joki, J. Paek, M. Vieira, D. Estrin, R. Govindan, E. Ko, "The Tenet Architecture for Tiered Sensor Networks," in ACM Sensys, Boulder, Colorado, USA , November 2006, pp. 153-166.

[127] J. H. Saltzer, D. P. Reed, D. D. Clark, "End-to-end arguments in system design," ACM Transactions on Computer Systems, 2(4):277, Nov. 1984.

[128] S. Olariu, A. Wadaa, Q. Xu, I. Stojmenovic, “A virtual infrastructure for wireless sensor networks,” in Handbook of Sensor Networks: Algorithms and Architectures (I. Stojmenovic, ed.), Wiley, 2005, 107-140.

[129] M. Caesar, M. Castro, E. Nightingale, A. Rowstron, ”Virtual ring routing: network routing inspired by DHTs,” in ACM SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, Vol. 36, No. 4. (2006), pp. 351-362.

[130] M. Welsh, G. Mainland, “Programming sensor networks using abstract regions,” in NSDI. 2004, pp. 29-42.

[131] K. Whitehouse, C. Sharp, E. Brewer, D. Culler, “Hood: a neighborhood abstraction for sensor networks,” in MobiSYS '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services, ACM Press, 2004 , pp. 99-110.

[132] J. N. Al-Karaki, A. E. Kamal, “Routing techniques in wireless sensor networks: a survey,” IEEE Wireless Communications, vol. 11, no. 6, pp. 6–28, 2004.

[133] Y. Chen, C. Chuan, Q. Zhao, “Sensor Placement for Maximizing Lifetime per Unit Cost in Wireless Sensor Networks,” in Proceedings of the IEEE Military Communication Conference (MILCOM’05), Atlantic City, NJ, October, 2005.

[134] Y. Shi, Y. Thomas Hou, “Approximation Algorithm for Base Station Placement in Wireless Sensor Networks,” in Proceedings of the IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON), San Diego, CA, June 18–21,2007. pp. 512–519.

[135] W. Wang, V. Srinivasan, K. Chua, "Using mobile relays to prolong the lifetime of wireless sensor networks," in Proceedings of the ACM MobiCom, Cologne, Germany ,August 28–September 2, 2005.

[136] A. Kinalis, S. Nikoletseas, "Scalable data collection protocols for wireless sensor networks with multiple mobile sinks," in Proceedings of the 40th ANSS, March 2007, pp. 60–69.

[137] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in ACM SenSys'03, Los Angeles, California, USA , 2003. pp. 28 - 39.

[138] F. Xue, R. Kumar, "The Number of Neighbors Needed for Connectivity of Wireless Networks," Wireless Networks, Vol. 10, No. 2, pp. 169-181, Mar 2004.

[139] A. Savvides, C. Han, M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in Proceedings of ACM MobiCom'01, Rome, Italy, July 2001, pp. 166–179.

[140] J. Albowicz, A. Chen, L. Zhang, "Recursive position estimation in sensor networks," in Ninth International Conference on Network Protocols, November 2001, pp. 35-41.

[141] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, J. Zhao., "Habitat Monitoring: Application Driver for Wireless Communications Technology," in ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, San Jose, Costa Rica, April 2001.

[142] R. Szewczyk, A. Mainwaring, J. Polastre, D. Culler, "An Analysis of a Large Scale Habitat Monitoring Application," in ACM Sensys, Baltimore, MD, USA, November 2004, pp. 214 - 226.

[143] A. Kansal, M. Rahimi, W. J. Kaiser, M. B. Srivastava, G. J. Pottie, D. Estrin, "Controlled mobility for sustainable wireless networks," in Proc. IEEE Sensor and Ad Hoc Communications and Networks (SECON), Santa Clara, CA, Oct. 2004. pp. 1-6.

[144] S. Tilak, N.B. Abu-Ghazaleh, and W. Heinzelman, "Infrastructure Tradeoffs for Sensor Networks," in WSNA, Georgia, USA, September 2002, pp. 49-58.

[145] M. I. Shamos, F. P. Preparata, Computational Geometry. Springer-Verlag, Berlin, 1985.

[146] G. Xing, C. Lu, R. Pless, Q. Huang, "Impact of sensing coverage on greedy geographic routing algorithms," IEEE Transaction on Parallel and Distributed Systems Vol. 17, No. 4, pp. 348–360, April 2006.

[147] I. Stojmenovic, X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," IEEE Transactions on Parallel and Distributed Systems, Vol. 12, No. 10, pp. 1023 – 1032, October 2001.

[148] M. Cardei, J. Wu, Coverage in wireless sensor networks. In Handbook of Sensor Networks, M. Ilyas, Ed. CRC Press, West Palm Beach, FL. July 2003.

[149] I. Dietrich, F. Dressler, "On the lifetime of wireless sensor networks," ACM Transactions on Sensor Networks (TOSN), Volume 5, Issue 1, February 2009.

[150] I. Stojmenovic , D. Liu , X. Jia, "A scalable quorum based location service in ad hoc and sensor networks," International Journal of Communication Networks and Distributed Systems, invited paper, Vol. 1, No. 1, pp. 71-94. 2008,

[151] R. Kumar, V. Tsiatsis, M. Srivastava, "Computation hierarchy for in-network processing," ACM Mobile Networks and Applications, Volume 10 , No. 4 , pp. 505 - 518 , 2005.

[152] M. Yarvis et al. “Exploiting heterogeneity in sensor networks,” in Proc. 24th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Infocom 2005), Mar. 2005, pp. 878- 890 vol. 2.

[153] M. Soini, J. Greunen, J. Rabaey, L. Sydanheimo, “Beyond Sensor Networks: ZUMA Middleware,” IEEE Wireless Communication and Networking Conference, Hong Kong, China, 2007, pp. 4318-4323.

[154] Y. Diao, D. Ganesan, G. Mathur , P. Shenoy, “Re-thinking Data Management for Storage-centric Sensor Networks,” in Proceedings of the Third Biennial Conference on Innovative Data Systems Research (CIDR), Asilomar, CA, January 2007, pp. 22-31.

[155] N. Bulusu, J. Heidemann and D. Estrin, “GPS-less Low Cost Outdoor Localization For Very Small Devices,” IEEE Personal Communications, Special Issue on "Smart Spaces and Environments", Vol. 7, No. 5, pp. 28-34, October 2000.

[156] D. Niculescu, B. Nath, “Trajectory-based forwarding and its applications,” in MobiCom, Brazil, 2003, pp. 260–272.

[157] A. Meliou, D. Chu, C. Guestrin, J. Hellerstein, W. Hong , “Data Gathering Tours in Sensor Networks,” n the fifth International Conference on Information Processing in Sensor Networks (IPSN'06), USA, April 2006.

[158] A. Ghosh, S. Das, “Coverage and connectivity issues in wireless sensor networks: A survey,” Elsevier Pervasive and Mobile Computing, Vol. 4, No. 3, pp. 303-334, June 2008.

[159] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," in

ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001), July 16-21, 2001.

[160] L. Feeney, M. Nilsson," Investigating the Energy Consumption of a Wireless Network Interface in an Ad hoc Networking Environment," in proceedings of IEEE InfoCom, Anchorage, AK, USA, 2001, pp. 1548-1557 vol.3.

[161] Sonia Hashish, Ahmed Karmouch, "Deployment-Based Solution for Prolonging Lifetime in Sensor Networks with Multiple Mobile Sinks," Ad Hoc & Sensor Wireless Networks Journal (AHSWN), Special Issue Wireless Sensor and Actor Networks, Vol. 7, (1-2), PP. 23-49, 2009.

[162] Sonia Hashish, Ahmed Karmouch," Deployment-based Solution for Prolonging Network Lifetime in Sensor Networks," in IFIP Conference on Wireless Sensors and Actor Networks. Ottawa, Canada, July 2008. PP. 85-96. (Best student paper award)

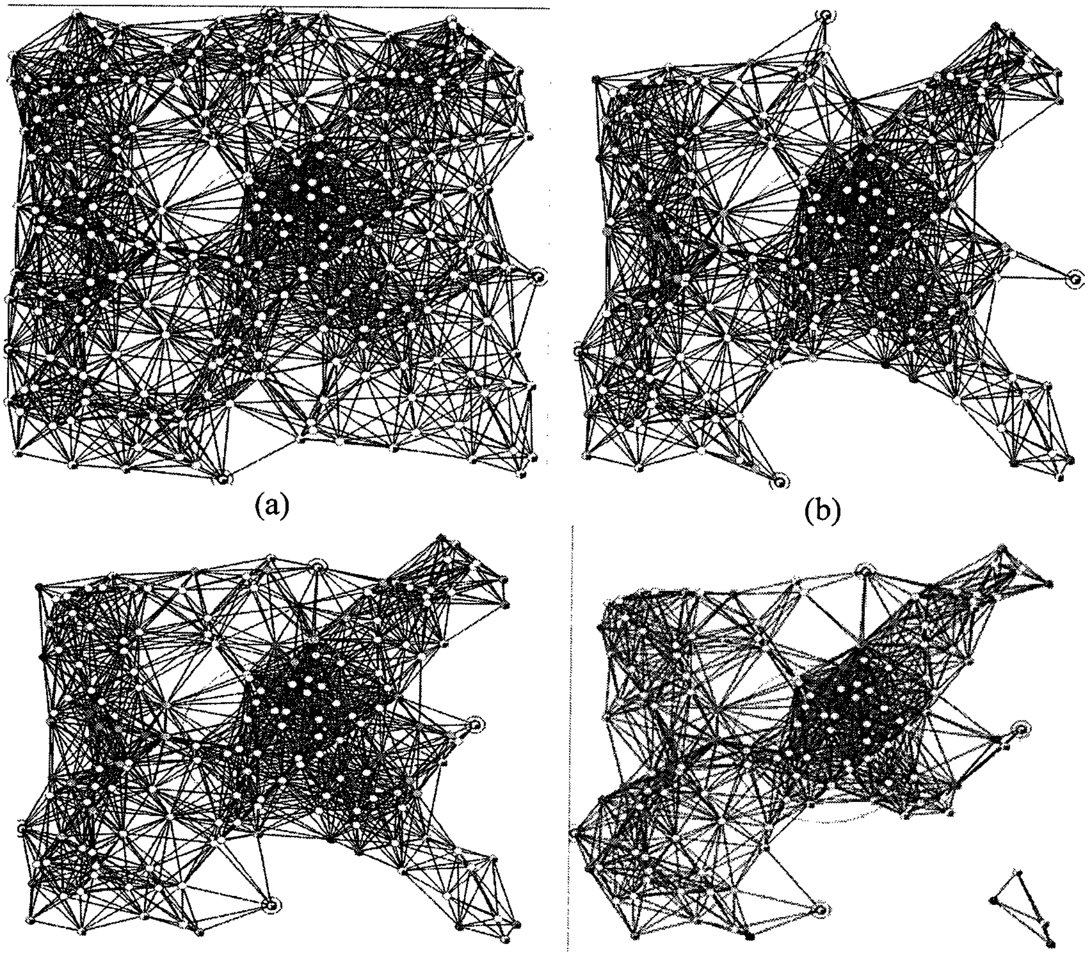
[163] Sonia Hashish, Ahmed Karmouch, "Hybrid data dissemination approaches for sensor network," in IEEE Consumer Communications and Networking Conference CCNC'07-Sensor Networks Track, Las Vegas, NV, USA,, Jan. 2007, PP. 838-843.

[164] Sonia A. Hashish, Ahmed Karmouch, "Topology-based on-board data dissemination approach for sensor network," in ACM MobiWac, Crete, Greece, Oct. 2007. PP. 33-41.

[165] Sonia Hashish, Ahmed Karmouch, "An Adaptive Rendezvous Data Dissemination for Irregular Sensor Networks with Multiple Sinks," Elsevier Computer Communications (COMCOM), Vol 33, Issue (2), pp. 176-189. February 2010.

[166] Sonia Hashish, Ahmed Karmouch, "Mobility-based generic infrastructure for large scale sensor network architecture," in IEEE ICC '09, Dresden, Germany, 13-18 June 2009.

Appendix A: Boundary Peeling Data Collection Simulation Snapshots



(a) (b)

(c) (d)

Fig. A.1 Boundary Peeling Data Collection simulation snapshots at different simulation time, (a) the initial topology, 4 sinks are deployed at the network peripheral (nodes with double circles), perimeter nodes are recognized and each node is marked with small dark rectangular, (b) Pre-movement status: network shrinking; three sinks lost 50% of their initial one-hop neighbors, (c) Sink movement, (d) The network partition (some nodes form an isolated island- the simulation stops at this point).