Energy-efficient Reliable Wireless Sensor Networks

ZHOU Yangfan

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

August 2006

Abstract of thesis entitled:

Energy-efficient Reliable Wireless Sensor Networks

Submitted by ZHOU Yangfan

for the degree of Master of Philosophy

at The Chinese University of Hong Kong in August 2006

In the recent decade, advances in Micro Electro-Mechanical Systems (MEMS) have made in-situ sensing with wireless sensor networks (WSNs) a promising technique. As wireless integrated network sensors are powered with a small battery and they usually work in an unattended manner, the main constraint of a sensor node is that its energy resource is limited. To enable in-situ sensing, sensor nodes and WSNs should function in an energy-efficient manner. Energy optimization techniques must be performed in every level of the design of a sensor network system. The work described in this thesis investigates various aspects of power saving approaches to achieve energy-efficient and reliable WSNs.

We first study routing issues and data transportation issues. Based on the features of WSNs, we discard the common layering network-protocol principle by coupling data transport protocol and applications: let the applications solve an optimization problem and feed back required reporting rates of sources. Based on this consideration, we propose PORT, a Price-Oriented Reliable Transport protocol for wireless sensor networks to reliably and energy-efficiently convey sensor information to the sink.

In our second work, we examine the problem of transmitter power control for energy-efficient sensor-to-sink communi-

cations. We model this problem based on the network and application features of WSNs. An intuitive implementation to solve this problem, namely BOU (Broadcast-On-Update), is presented. We identify the broadcast explosion problem in BOU, and then improve BOU by allowing a waiting period before each broadcasting. We show that the waiting time should be proportional to the probability that a node would find a more energy-efficient path to the sink, and present an efficient approximation algorithm to calculate the probability.

In the last work presented in this thesis, we propose ι, a novel index for evaluation of point-distribution. ι is the minimum distance between each pair of points normalized by the average distance between each pair of points. We find that a set of points that achieve a maximum value of ι result in a honeycomb structure. We propose that ι can serve as a good index to evaluate the distribution of the points, which can be employed in coverage-related problems in WSNs. We set out to validate this idea by employing ι to a sensor-grouping problem. We formulate a general sensor-grouping problem for WSNs and provide a general sensing model. With an algorithm called Maximizing-ι Node-Deduction (MIND), we show that maximizing ι at sensor nodes is a good approach to solve this problem.

In conclusion, this thesis studies various energy-efficient approaches to achieve reliable and energy-efficient wireless sensor networks. Simulation studies demonstrate the effectiveness of these approaches.

学位论文摘要

学位论文题目：高效节能的可靠无线传感器网络
提交人：周扬帆
学位：哲学硕士，
香港中文大学，二零零六年八月

近十年来，微型电子机械系统的飞速发展使得应用无线传感器网络的现场监测成为一种很有发展前景的技术。微型传感器节点用一块小电池做电源，而这些节点一般而言需要无人值守工作；这使得电源成为微型传感器节点的主要瓶颈所在。为使这种现场监测技术有可行的应用，微型传感器节点乃至整个无线传感器网络都必须很节能地工作。整个无线传感系统设计的各个层面，都必须应用各种优化能耗的技术。本学位论文研究无线传感器网络各方各面的节能方法，以实现高效节能的可靠无线传感器网络。

首先，我们研究路由和数据传输层面的问题。根据无线传感器网络的特点，我们摒弃了传统网络协议严格的分层设计，而让数据传输层和应用层耦合在一起；我们让应用层解决一个优化问题，并把结果反馈给传输层，通过传输层来控制源节点的发包率。基于这样的思路，我们设计了一个名为 PORT（Price-Oriented Reliable Transport）的协议。此协议可让无线传感器网络可靠节能地将传感器节点的现场监测数据传送给数据收集点。

在我们的第二个工作中，我们研究了微型传感器节点无线电发射器的发送功率控制问题来实现节能的传感器节点到数据收集点的数据通讯。我们根据无线传感器网络极其应用的特点来为这个问题建模。接着，我们提出名为 BOU 的这个问题解法的一种直接实现。我们指出这个实现会导致广播风暴问题，而改进方法是在每个传感器节点每次广播之前，等待一段时间再进行广播。这个等待时间应与该传感器节点能在以后找到一条比其已知的最节能的到数据收集点的路径更加节能的路径的概率成正比。我们进而提出了一种近似方法来计算这个概率。

在本学位论文述及的第三个研究工作中，我们提出了一个新的点分布情况的评价参数（命名为ι）。此参数是这些点的两两距离的最小值比上它们两两距离的平均值。我们发现，二维空间中的点，要达到上述参数最大，它们的位置构成的Vonoroi结构是蜂窝结构。我们据此认为，这个参数可作为一个很好的点分布情况的评价参数，并可应用于解决和无线传感器网络的覆盖相关的问题。为验证这个想法，我们研究此参数如何应用于解决无线传感器节点的分组问题。我们提出了一个无线传感器节点的分组问题的普适模型和一个传感器节点的感应监测模型。我们设计了一个名为MIND（Maximizing-ι Node-Deduction）的算法，通过对这个算法的性能研究，我们指出优化无线传感器节点的这个参数，是无线传感器网络分组问题一个很好的解决方法。

综上所述，此学位论文研究了各种节能技术，以实现高效节能的可靠无线传感器网络。仿真实验结果表明这些方法是有效的。

學位論文摘要

學位論文題目：高效節能的可靠無線傳感器網絡
提交人：周揚帆
學位：哲學碩士，

香港中文大學，二零零六年八月

近十年來，微型電子機械系統的飛速發展使得應用無線傳感器網絡的現場監測成爲一種很有發展前景的技術。微型傳感器節點用一塊小電池做電源，而這些節點一般而言需要無人值守工作；這使得電源成爲微型傳感器節點的主要瓶頸所在。爲使這種現場監測技術有可行的應用，微型傳感器節點乃至整個無線傳感器網絡都必須很節能地工作。整個無線傳感系統設計的各個層面，都必須應用各種優化能耗的技術。本學位論文研究無線傳感器網絡各方各面的節能方法，以實現高效節能的可靠無線傳感器網絡。

首先，我們研究路由和數據傳輸層面的問題。根據無線傳感器網絡的特點，我們摒棄了傳統網絡協議嚴格的分層設計，而讓數據傳輸層和應用層耦合在一起；我們讓應用層解決一個優化問題，並把結果反饋給傳輸層，通過傳輸層來控制源節點的發包率。基於這樣的思路，我們設計了一個名爲 PORT（Price-Oriented Reliable Transport）的協議。此協議可讓無線傳感器網絡可靠節能地將傳感器節點的現場監測數據傳送給數據收集點。

在我們的第二個工作中，我們研究了微型傳感器節點無線電發射器的發送功率控制問題來實現節能的傳感器節點到數據收集點的數據通訊。我們根據無線傳感器網絡極其應用的特點來爲這個問題建模。接著，我們提出名爲 BOU 的這個問題解法的一種直接實現。我們指出這個實現會導致廣播風暴問題，而改進方法是在每個傳感器節點每次廣播之前，等待一段時間再進行廣播。這個等待時間應與該傳感器節點能在以後找到一條比其已知的最節能的到數據收集點的路徑更加節能的路徑的概率成正比。我們進而提出了一種近似方法來計算這個概率。

在本學位論文述及的第三個研究工作中，我們提出了一個新的點分佈情況的評價參數（命名爲ι）。此參數是這些點的兩兩距離的最小值比上它們兩兩距離的平均值。我們發現，二維空間中的點，要達到上述參數最大，它們的位置構成的Vonoroi結構是蜂窩結構。我們據此認爲，這個參數可作爲一個很好的點分佈情況的評價參數，並可應用於解決和無線傳感器網絡的覆蓋相關的問題。爲驗證這個想法，我們研究此參數如何應用於解決無線傳感器節點的分組問題。我們提出了一個無線傳感器節點的分組問題的普適模型和一個傳感器節點的感應監測模型。我們設計了一個名爲MIND（Maximizing-ι Node-Deduction）的算法，通過對這個算法的性能研究，我們指出優化無線傳感器節點的這個參數，是無線傳感器網絡分組問題一個很好的解決方法。

綜上所述，此學位論文研究了各種節能技術，以實現高效節能的可靠無線傳感器網絡。仿真實驗結果表明這些方法是有效的。

Acknowledgement

First of all, I feel very grateful to my parents and my sister. In my life, they are always endless source of warm support and love.

I wish to thank my supervisors, Prof. Michael R. Lyu and Prof. Jiangchuan Liu (currently, he is with Simon Fraser University), for their generous guidance and patience given to me during my postgraduate study. Their support, encouragement, and most importantly, their advice, are extremely essential and valuable in my research work. They not only shape the work described in this thesis, but also shape my academic research styles and directions. Moreover, I will never forget what Michael has told me: "Everyone deserves a second chance". It is this very second chance he gave me that makes my academic career possible.

I would like to thank Prof. Soung-Chiang Liew in Information Engineering department, Prof. Man Hon Wong in Computer Science and Engineering department, and Prof. Bo Li at the Hong Kong University of Science and Technology for their precious time to serve as my research markers during my postgraduate study.

Dr. Kun Zhang provided some constructive comments in the work described in Chapter 4. Dr. Evan Young suggested an algorithm in the work described in Chapter 5. Dr. Haixuan Yang, Ms. Edith Ngai, and Ms. Hui Wang have conducted research work with me during my postgraduate study. Working with them was very happy experience. Specially, Dr. Haixuan Yang taught me much on mathematics and machine learning: his insightful ideas often embeds in my research work. Dr. Jiqiang Song taught me much on how to conduct academic research work. He is the one who brought me into the world of academic research. I would like to express my grateful thanks to all of them.

Last but not least, my research work described in this thesis was substantially supported by two grants, RGC Project No. CUHK4205/04E and UGC Project No. AoE/E-01/99, of the Hong Kong Special Administrative Region, China.

Contents

Abstract **i**

Acknowledgement **v**

1 Introduction and Background Study **1**

1.1 Wireless Sensor Networks 1

1.1.1 Wireless Integrated Network Sensors . . . 1

1.1.2 Main Challenge of In-situ Sensing with Sensor Nodes: Limited Energy Resource . 3

1.1.3 Networking the Sensor Nodes 4

1.2 Applications of Wireless Sensor Networks 4

1.3 Characteristics of Wireless Sensor Networks: A Summary 6

1.4 Energy-Efficient and Reliable Wireless Sensor Networks 9

2 PORT: A Price-Oriented Reliable Transport Protocol **12**

2.1 Reliable Sensor-to-Sink Data Communications in Wireless Sensor Networks 14

2.2 Related Work 17

2.3 Protocol Requirements 20

2.4 Design Considerations 25

2.4.1 The concept of node price 25

2.4.2 Link-loss rate estimation 28

2.4.3 Routing scheme 29
2.5 Protocol Description 31
2.5.1 Task initialization 31
2.5.2 Feedback of newly desired source reporting rates 32
2.5.3 Feedback of wireless communication condition 32
2.5.4 Fault tolerance and scalability considerations 33
2.6 Protocol Evaluation: A Case Study 34
2.6.1 Simulation model 34
2.6.2 Energy consumption comparison 36
2.6.3 The impact of reporting sensors' uncertainty distribution 39
2.7 Conclusion 40

3 Setting Up Energy-Efficient Paths 41
3.1 Transmitter Power Setting for Energy-Efficient Sensor-to-Sink Data Communications 46
3.1.1 Network, communication, and energy consumption models 46
3.1.2 Transmitter power setting problem for energy-efficient sensor-to-sink data communications 49
3.2 Setting Up the Transmitter Power Levels for Sensor-to-Sink Traffic 51
3.2.1 BOU: the basic algorithm 52
3.2.2 Packet implosion of BOU: the challenge . . 53
3.2.3 Determining the waiting time before broadcasting 56
3.2.4 BOU-WA: an approximation approach . . 60
3.3 Simulation Results 62
3.3.1 The comparisons of BOU and BOU-WA . 63
3.3.2 The approximation of BOU-WA 65

3.4 Related Work 67
3.5 Conclusion Remarks and Future Work 69

4 Solving the Sensor-Grouping Problem 71
4.1 Introduction 73
4.2 The Normalized Minimum Distance ι: A Point-Distribution Index 74
4.3 The Sensor-Grouping Problem 77
4.3.1 Problem Formulation 80
4.3.2 A General Sensing Model 81
4.4 Maximizing-ι Node-Deduction Algorithm for Sensor-Grouping Problem 84
4.4.1 Maximizing-ι Node-Deduction Algorithm . 84
4.4.2 Incremental Coverage Quality Algorithm: A Benchmark for MIND 86
4.5 Simulation Results 87
4.5.1 Number of Groups Formed by MIND and ICQA 88
4.5.2 The Performance of the Resulting Groups 89
4.6 Conclusion 90

5 Conclusion 92

A List of Research Conducted 96

B Algorithms in Chapter 3 and Chapter 4 98

Bibliography 102

List of Figures

1.1 A typical architecture of a sensor node, the MIT μAMPS sensor node. (Fig. 1 of [44]) 2

2.1 A scenario of the WSN. 22
2.2 An example showing oscillation 29
2.3 The simulation network. 36
2.4 Energy consumption comparisons. 38
2.5 Energy consumption comparisons: different phenomenon positions in a grid 39

3.1 A scenario of a network 54
3.2 The estimated probability distribution of node cost 59
3.3 Total number of broadcasts 64
3.4 The energy consumption overhead 65
3.5 The converging time 66
3.6 The average estimation error of ρ' 67
3.7 The average number of broadcasts by each node . 68

4.1 Node Number = 20, $\iota = 0.435376$ 76
4.2 The number of groups found by MIND and ICQA 88
4.3 The failure numbers of MIND and ICQA 90

List of Tables

2.1 The descriptions of the symbols 26
2.2 Simulation network settings. 37

3.1 The setting of the simulation network 62

4.1 The settings of the simulation networks 87
4.2 The grouping results of five networks with $n = 1500$ 89

Chapter 1

Introduction and Background Study

1.1 Wireless Sensor Networks

In recent years, advances in Micro Electro-Mechanical Systems (MEMS) have made it possible to integrate signal processing environmental data sensing, and wireless communication modules in one single small circuit board. Such technological developments make small, low-cost, low-power devices capable of sensing the environmental/physical data, collecting and processing such data, and communicating with wireless technology, a reality [20, 21, 41, 42]. Such devices are referred as wireless integrated network sensors (WINS) [5, 48], or in short, sensor nodes.

1.1.1 Wireless Integrated Network Sensors

Typical implementations of sensor nodes include the μAMPS sensor nodes developed by Massachusetts Institute of Technology (MIT) [44], the WINS sensor nodes developed by the University of California at Los Angeles (UCLA) [5, 48], and the MICA2 Mote sensor nodes developed by the University of California at Berkeley (Berkeley) [29].

Figure 1.1 demonstrates the architectural overview of the

MIT μAMPS sensor node. It can be viewed as a typical architecture of a sensor node.

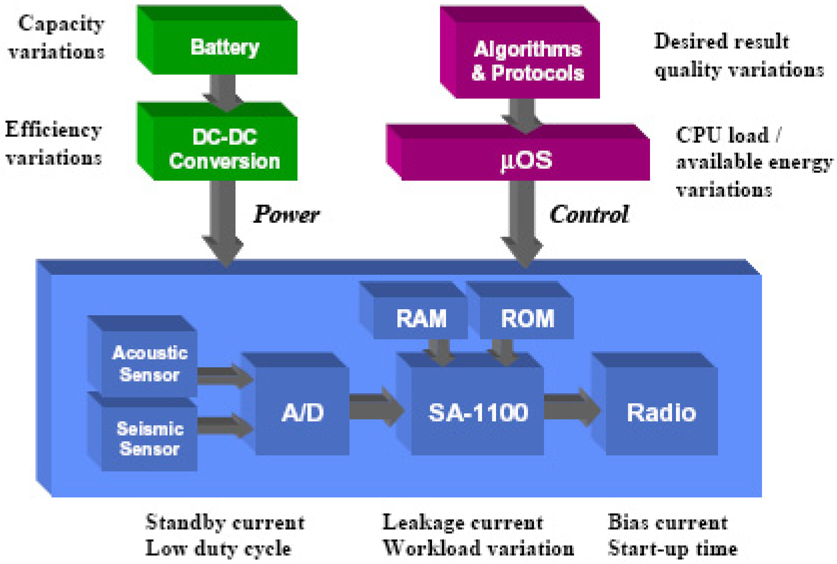

Figure 1.1: A typical architecture of a sensor node, the MIT μAMPS sensor node. (Fig. 1 of [44])

A typical sensor node consists of the following units [5, 29, 44, 48].

- A power supply unit, which consists of a small battery with DC-DC conversion to the appropriate voltages required by the electronic system of a sensor node.
- One or more physical sensors. Examples of sensors include thermoelectrical sensors to measure environmental temperature, acoustic sensors to monitor the sounds of interest, infrared sensors to capture the existence of the objects of interest, and seismic sensors to measure earth vibrations. The types of sensors employed are determined by the applications of the in-situ sensing tasks.
- An A/D convertor to convert the analog signals captured by a sensor to digital signals.

- A RF radio communication unit. Usually the transmitter power of the radio unit can be shifted to different levels.
- One or more microchip computer systems to process the digital signals from the A/D convertor, to control and monitor the behaviors of the sensor node, and to run other required software such as communication protocols.

1.1.2 Main Challenge of In-situ Sensing with Sensor Nodes: Limited Energy Resource

Sensor nodes are physically deployed in an area to collect data about some physical phenomena of interest. Such in-situ sensing approach facilitate monitoring and controlling of physical environments with better accuracy [21]. And also, sensor nodes are cheap. In-situ sensing is more economic, comparing to remote sensing approaches with devices such as radars, sonars, and satellites. These merits arouse great research interests among academic people [2, 3, 33, 37, 52, 67, 77].

A sensor node is typically powered by a small battery. It suffers from limited energy resource. What makes the situation worse is that recharging the sensor nodes manually is usually not facile, and hence costs highly, because sensor nodes usually operate in an unattended manner. Even in some application case that sensor nodes may work in a hostile environment, recharging them is impractical.

To prolong the entire lifetime of an in-situ sensing system to typical requirements which are on the order of months to years, energy efficiency becomes a vital hardware and software design consideration of sensor nodes [5, 29, 44, 45, 48]. It is a major challenge to enable environmental sensing with in-situ sensor nodes [22, 32].

1.1.3 Networking the Sensor Nodes

Among the great research efforts people have put in to enable environmental sensing with in-situ sensor nodes, a very important issue is how to network such sensor nodes. As individual sensor nodes are with low sensing and processing capabilities due to their commonly low-cost implementation, networking of a large number of sensor nodes can enhance the amount, as well as the accuracy, of the information obtained by the sensor nodes [21, 22].

In proposed implementations of environmental sensing with in-situ sensor nodes, a number of in-situ sensor nodes are deployed to collect data about some physical phenomena of interest. The sensor nodes form an ad hoc multi-hops wireless network through which the collected data are conveyed to collectors, *e.g.*, hand-held devices such as PDAs and laptop computers. Such data collectors are called sink nodes. Sinks are usually with higher computational capabilities and power. They might also be able to communicate with other computers, *e.g.*, they are connected to the Internet. Sinks are where the outside world obtain the data from the sensor nodes and where the outside world control the behaviors of the sensor nodes. Such networks are referred as wireless sensor networks (WSNs).

1.2 Applications of Wireless Sensor Networks

The self-organized nature of WSNs facilitates their deployment, and also increases their fault-tolerance and hence robustness. In addition, the low-cost implementation of sensor nodes make it possible for high-density and large-scale deployment, which not only enhances the robustness of WSNs, but also increases the possible scale of the sensing area. These features make WSNs attractive in military applications, such as battlefield surveillance and enemy tracking. That's what the research on WSNs is initially driven by.

Besides military applications, which are outside the interests of us, there are a great variety of proposed civil applications of WSNs such as environmental surveillance, habitat monitoring, motor-traffic tracking, surveillance, *etc.* [2, 3].

In the work by A. Mainwaring *et al* [40], a specific habitat monitoring application with WSNs is studied. This application case is a good representative of the civil application domain of WSNs.

In this experimental application, a sensor network composed by MICA Mote [29] sensor nodes are deployed on Great Duck Island, Maine, to monitor the behavior of storm petrels. Temperature, photosensitive, barometrical, humidity and thermopile sensors are employed in this project. Sensor nodes are placed at the habitat of storm petrels (*e.g.* inside a burrows). These sensor nodes are grouped into sensor patches, and transmit sensor reading to a sink that is responsible for forwarding the data from the sensor patch to a remote base station through a local transmit network. The base station then copies the data every 15 minutes to a database server in Berkeley over satellite link. Users can can access the data from the database server and can also control the behaviors of the network such as adjusting the sampling rates.

In environmental surveillance, Automated Local Evaluation in Real-Time (ALERT) [4] is a project that provides real-time rainfall and water level information to evaluate the possibility of potential flooding. WSNs are also proposed to perform Structure health monitoring to detect possible damage and predict the residual life of structures [47, 74]. A biomedical application of WSNs is studied in [59] aiming to turn artificial retina into reality. Also, work in [62] presents a "Smart Kindergarten" to introduce WSNs into childhood education.

Although most of the proposed applications of WSNs are only in experimental phase and they just serve as test-bed to help researchers find out challenges and verify existing ideas, real-world applications of WSNs are expected to have a great impact in our life in a foreseeable future. It is very promising to study the WSNs so as to turn this prevision into reality.

1.3 Characteristics of Wireless Sensor Networks: A Summary

Wireless Sensor networks are similar to Mobile Ad Hoc Networks (MANETs) as they are both wireless networks and they are both involved in multi-hops wireless communications. However, as one can see from the above discussion, WSNs are very different from traditional data networks including MANETs. The characteristics of WSNs are summarized in this section, together with some comparisons with MANETs.

- WSNs are application-driven networks. Different WSNs have different task-specific requirements. It is quite different to traditional general-purpose networks. This means that the protocols for WSNs can be totally new. It is not required to tailor the protocol design for WSNs in order to achieve compatibility to existing protocols of traditional general-purpose networks. For example, unlike in MANETs, it is not required to build up an IP address mechanism over WSNs. Also, the design of protocol stack do not have to confine to traditional layering thought.

- WSNs are self-organized networks. They are usually with large scale and high node density. The number of sensor nodes in a network might be several hundred or even reach over a thousand. Such a large number of sensor nodes are usually left unattended after they are deployed. This means that a WSN should function in a completely autonomous manner.

- WSNs suffer from limited energy supply. Protocols for WSNs should be energy-efficient. Sensor node is battery-powered. Recharging a sensor node is expensive, if not impossible, which means that it is impractical to revive a sensor node after its battery energy is drained. Energy-efficiency must be a main consideration of protocol design for WSNs.

- WSNs are instable. First, low-cost implementation of a sensor node make it easy to fail. Also, due to limited energy resource, the power of a sensor node is easily drained, which results in permanent disfunction of the sensor node. Second, WSNs are usually working in bad or even hostile environments. The wireless links between sensor nodes are fragile and subject to failures. These two situations make a given wireless communication path between two sensor nodes instable and even easy to be permanently damaged. Therefore, protocols for WSNs should adapt to such network instability.

- The mobility feature of WSNs is different to MANETs. In MANETs, nodes are laptop computers, PDAs carried by people. Nodes are mobile. However, in WSNs, nodes are sensor devices. Usually they are not mobile after deployment. But the locations of the interesting phenomena might be mobile, *e.g.*, when the network is tracking a motor vehicle. Also, the sink of a WSN may be mobile. A sink may be a hand-held device such as a PDA or a laptop computer. It may be carried to the network area to collect the sensor data hourly, and each time its location may be different.

- The network traffic feature of WSNs is different to MANETs. In MANETs, network traffic is like the traditional wired networks. It might usually be unicasting. Every node may require to communicate with all the others ones: traffic is usually in a peer-to-peer manner. Broadcasting traffic and multicasting traffic are also possible in MANETs. However, in WSNs, usually network traffic is in a many-to-one manner. Many nodes send data packets to a single node, *i.e.*, the sink.

- A global-identity-based addressing mechanism which is required by traditional data networks including MANETs may be unnecessary for WSNs. The application of WSNs is mainly on collecting data of some phenomena of interest. Depending on specific task requirement, it may care where a particular phenomenon takes place or whether a particular phenomenon happens, rather than which particular node is currently report data.

- In-network data processing may be required in WSNs. WSNs are deployed to collect data of some phenomena of interest. Usually, there would be many sensor nodes that collect the data of a particular phenomenon and report the data to the sink. Naturally, these data packets reported by each source sensor nodes could have some redundancy. This redundancy can be further exploited by in-network data processing approaches such as data aggregation or data fusion.

- Finally, WSNs suffer more computational constraints than MANETs. Comparing to a node of a MANET, a sensor node has by far lower memory capacity and its computational speed is much slower due to its low-cost design, small size and low power. One cannot expect to save large volume of data in a sensor node or program it to do complex computations.

These features are the reasons that many network protocols for traditional data networks including MANETs are not suitable for WSNs. We have to employ another family of network protocols for data communication and network organization in WSNs which take account of the above network characteristics.

1.4 Energy-Efficient and Reliable Wireless Sensor Networks

As discussed above, the main constraint of sensor node is that its energy resource is limited. To enable in-situ sensing, sensor nodes and WSNs should function in an energy-efficient manner.

Besides conventional low-power design techniques in electronic level [14], energy optimization techniques must also be performed in every level of the design of a sensor node system. Considerations include energy efficient computing and data processing, power-controlled wireless communication, and energy efficiency protocols (*i.e.*, [28, 49]), which can be classified into the following categories (Cardei *et al* have similar classification. See [10, 12]).

- Route data packets via energy-efficient path, *i.e.*, route data packets so that the energy required for the data transport is minimized.
- Exploit the redundancy of data packets through some techniques such as in-network data aggregation or data fusion, and source reporting rate control.
- Adjust the transmitter power level of a sensor node to let it communicate with its intended receivers in an energy-efficient way.

- Avoid useless packets, *i.e.*, minimize protocol overheads.
- Schedule the in-network sensor nodes so that they can work in sleep mode to save energy when they are not required to perform sensing tasks or communication tasks.

The work described in this thesis studies various aspects of power saving approaches to achieve energy-efficient and reliable WSNs [1].

Chapter 2 studies how to minimize energy consumption of the communications for sensor reporting traffic, which falls into class 1 and 2 in the above classification. Chapter 3 studies the transmitter power control problem and propose an low-overhead approach to set up energy-efficient communication paths, which falls into class 1 and 3 of the above classification. Both studies also take "minimizing protocol overheads" (class 4) as an important design considerations. Chapter 4 studies how to schedule sensor nodes to divide the nodes into as many as possible disjoint groups. Each of group can maintain the sensing-task requirements. It falls into class 5 of the above classification.

[1] For a complete list of the work conducted during the postgraduate study, please refer to the publication list in Appendix A

□ **End of chapter.**

Chapter 2

PORT: A Price-Oriented Reliable Transport Protocol

Summary

In wireless sensor networks, to obtain reliability and minimize energy consumption, a dynamic rate-control and congestion-avoidance transport scheme is very important. We notice that reporting packets may contribute to the sink's fidelity of its knowledge on the phenomenon of interest to different extents. Thus, reliability cannot simply be measured by the sink's total incoming packet rate as considered in current schemes. Also, communication costs between sources and the sink may be different and may change dynamically. Based on these considerations, we propose PORT (Price-Oriented Reliable Transport protocol) to facilitate the sink to achieve reliability. Under the constraint that the sink must obtain enough fidelity for reliability purpose, PORT minimizes energy consumption with two schemes. One is based on the sink's application-based optimization approach that feeds back the optimal reporting rates. The other is a locally optimal routing scheme according to the feedback of downstream communication conditions. PORT can adapt well to the communication conditions for energy saving while maintaining the necessary level of reliability. Simulation results in an application case study demonstrate the effectiveness of PORT. This chapter is based on the work presented in [79].

2.1 Reliable Sensor-to-Sink Data Communications in Wireless Sensor Networks

Although different WSNs have different task-specific requirements, they all require a sensor-to-sink data transport scheme[1] to take account of two important issues. The first is reliability assurance, which means we must guarantee that the sink can obtain enough information about the phenomenon of interest. The second is energy-efficiency, as recharging the sensor nodes is usually impractical [22]. Therefore, a sensor-to-sink data transport scheme should aim to minimize energy consumption under the constraint that the sink can collect enough information on the phenomenon of interest.

The notion of reliability on sensor-to-sink communication was first introduced in [54], where the authors notice that, unlike existing WSN transport schemes (*e.g.*, PSFQ [68] and RMST [63]) that focus on end-to-end reliable data transferring, absolute end-to-end reliable data transport is usually not needed when transmitting sensor reporting packets. Packet loss within a certain limit can usually be well tolerated in most application scenarios. This notion is important to the design of a reliable sensor-to-sink data transport protocol; however, there are still several unconsidered problems in current approaches.

First, we notice that the packets from different sources may make a different contribution to improve the sink's information on the phenomenon of interest. We regard the contribution of a source node as being how much it reduces the sink's uncertainty on the data about the phenomenon. Thus reliability cannot simply be measured by the total incoming packet rate, as consid-

[1]The sensor-to-sink data transport scheme refers to the data transport scheme that transfer the desired information collected by the in-situ sensors to the sink.

ered in current approaches, *e.g.*, ESRT (Event-to-Sink Reliable Transport) [54]. Instead, it should be assured with the cooperation of a reliable sensor-to-sink data transport scheme and network applications.

Second, to achieve reliability, ESRT adjusts the report rates of sources in an undifferentiated manner. But, as the communication cost from different sources to the sink may be different and may change dynamically, and also the contributions of packets from different sources are also different, adjusting the report rates of the sensor nodes in an undifferentiated manner is not the most energy-efficient way to increase the knowledge of the phenomenon. It is therefore necessary to bias the reporting rates of the sources.

Third, to minimize energy consumption, we must avoid links with high communication costs. Congestion always results in an increase in communication cost, and so congestion control is vital to minimize energy consumption. ESRT proposes to avoid congestion in an end-to-end manner by reducing reporting rates. CODA [69] also proposes to avoid congestion by slowing down sending rates. However, slowing down sending rates may cause the sink to receive fewer packets, which may yield in insufficient information on the phenomenon of interest. In this case, the sink will ask for higher reporting rates and that may cause congestion again if a reliability control mechanism like what ESRT proposes is employed. Therefore, besides an end-to-end congestion-avoidance mechanism, an in-network congestion-avoidance mechanism is also necessary.

In this chapter, we aim to address these problems by providing a Price-Oriented Reliable Transport protocol (PORT). PORT is based on the following assumptions.

- The sensor reporting traffic lasts for a considerable duration.

- The sink is aware of the sources of the data packets; *i.e.*, the sink can identify where a packet originates.
- The sink is aware of the information a packet carries.

The first assumption means source sensor nodes would keep reporting data on the phenomenon of interest for a long period of time. It is generally valid because in most application scenarios such as environmental monitoring, object tracking, surveillance, *etc.*, WSNs are employed to provide continuous data streaming about the phenomenon of interest.

The second assumption is also reasonable in most application scenarios. This is because of two reasons. First, it is usually necessary for the sink to know the physical location of the phenomenon. Where a packet originates provides information on where the phenomenon of interest is taking place. Second, the sink should usually fuse the data packets it has received. Each source node should be identified in order to provide information on how to fuse the packets. Note that PORT does not require a heavy-weighted address-based approach. It only requires the sink can identify different sources which are reporting data on the same phenomenon. This can be achieved, for example, by randomly generating an identifier and embedding it in reporting packets when a node is sensing and reporting the phenomenon of interest.

The third assumption means that the sink knows how a packet can improve its knowledge on the phenomenon of interest. It is true as the sink is where data packets are interpreted.

PORT employs *node price*, which is defined as the total number of transmission attempts across the network needed to achieve successful packet delivery from a node [2] to the sink, to measure the communication cost from a node to the sink.

[2] Successful packet delivery from a node means that the packet from the node arrives at the sink successfully.

Under the constraint that the sink must obtain enough information, PORT dynamically feeds back the optimal reporting rate to each source according to the current contribution of the packets from each source and the node price of each source.

Based on the neighboring nodes' feedback of their node prices and the loss rates of the links between the neighbors and the node, an in-network node dynamically allocates its outgoing traffic to avoid high loss rate paths (which are probably caused by congestion). PORT, in this way, alleviate congestion in an in-network manner. Also, congestion will increase the node price of the sources. The source reporting rate control mechanism of PORT is aware of node prices of the sources, and can decide to adjust the source reporting rates (it might slow down one with a high node price and speed up one with a low node price) with a guarantee that the sink can still obtain enough information. Hence, with this in-network congestion-avoidance mechanism and this end-to-end reporting-rate adjustment mechanism, PORT provides a good congestion-avoidance mechanism.

The rest of this chapter is organized as follows. Section 2.2 briefly surveys the related work. Section 2.3 discusses the requirements that a reliable sensor-to-sink data transport scheme should fulfill. In Section 2.4, we provide the design considerations of PORT. In Section 2.5, we elaborate the implementation of PORT. Section 2.6 evaluates our mechanism with NS-2 [23] in an application case. We conclude the chapter in Section 2.7.

2.2 Related Work

In traditional TCP/IP networks, data transport mechanism is implemented with an address-based end-to-end data communication concept. But, this is not appropriate for WSNs not only because of the need to simplify the implementation, but also to save energy [30].

As a WSN is mainly employed to convoy information on some particular phenomena of interest, one should emphasize more on the information obtained rather than who provides the information. For example, one might be interested in where a particular phenomenon takes place or whether a particular phenomenon happens rather than which particular node is currently report data. Hence, the addressing mechanism in WSNs is not of interest. Furthermore, in-network data processing is desirable. Usually, there would be many sensor nodes that collects the data of a particular phenomenon and report the data to the sink. The information reported by each of the sensor nodes contains redundancy. This redundancy should be further exploited by in-network data processing approaches such as data aggregation or data fusion in order to reduce the amount of data reported to the sink and thus save energy.

This consideration has led to the development of a new concept: data-centric communication [1, 30]. Data-centric concept is proposed in early work on WSNs [27, 30]. It has got well accepted in the literature and become important design notions when people design sensor-to-sink communication protocols.

Data-centric schemes deliver sensor data throughout the network with an application specific naming scheme for the data. Routing paths are constructed on-demand based on the specific task and the data packets are routed according to the data they carry. A representative example of data-centric routing is directed diffusion [30]. In directed diffusion, the sink requests its data of interest by broadcasting its *interests*. The interest packets are flooded throughout the network and the nodes set up *gradients* to save the data-centric routing information. Using the gradient filter, directed diffusion conveys sensor reporting data to the sink. The reporting packets might be delivered to the sink along multiple paths. The sink determines the best path and increases the desired reporting rate along this path.

This process is called path *reinforcement*. Directed diffusion can periodically reinitiate the reinforcement process to find the new best path. Many other schemes (*e.g.*, [8, 17, 24, 58, 60]) have been proposed based on directed diffusion or similar concepts (work in [1] provides a good survey on routing protocols for WSNs). But, most of the work does not consider the notion of *reliability* of sensor-to-sink communication within the design of the protocols.

PSFQ [68] and RMST [63] are concerned with reliable data transport protocol over WSNs. However, they aim at providing 100% reliable data transport for WSNs. In [54], the authors argue that absolute end-to-end reliable data transport is usually not needed for transmitting the sensor reporting packets. They propose ESRT (Event-to-Sink Reliable Transport) to address the reliable sensor-to-sink communication problem. They measure the reliability of the event features achieved in terms of total packet receiving rate. The communication is considered to be reliable if the number of the received packets is not less than the desired number of packets per unit time. ESRT ensures that the total incoming packet rate of the sink stays within the desired range by providing a mechanism to feed back the required reporting rate directly to the source nodes. But the reporting rate of each source is adjusted in an unbiased manner.

Congestion control for WSNs is studied in [54] and [69]. In [69], congestion is detected by sampling wireless channel utilization. In [54], congestion is detected according to the buffer utilization of the in-network nodes. These studies avoid congestion by slowing down the sending rate, regardless of what the node is reporting.

There are also other kinds of communications required in WSNs. For example, one may need to reprogramme sensor nodes [18, 36, 46, 64] or adjust the behaviors of sensor nodes such as data sampling rate [30].

2.3 Protocol Requirements

Because WSNs are employed to sense and convey information of some physical phenomenon of interest, the reliability of sensor-to-sink data transport should be considered as the fidelity[3] of the knowledge obtained by the sink on the physical phenomenon. Based on this notion, we define that a sensor-to-sink data transport is *reliable* when the transport mechanism can assure that the sink is able to collect enough information; *i.e.*, *the sink can obtain enough fidelity of the knowledge on the phenomenon of interest.*

Specifically, we consider the sensor-to-sink data transport is *reliable* when the following inequation holds.

$$u = f(t_1, t_2, ..., t_m) \geq u', \tag{2.1}$$

where m denotes the number of the sources, u' denotes the required minimum fidelity on the phenomenon of interest and u denotes the current fidelity obtained, which is a function of the incoming packet rate t_i ($i = 1, 2, ..., m$) from each of the sources. Note that we adopt incoming packet rates of the sink rather than the reporting rates of the sources, as packet loss along the sensor-to-sink paths would cause that the reporting rates do not well indicate the fidelity obtained by the sink [54].

As each data packet sent by the source sensor node obviously contains some information of the phenomenon of interest and therefore contributes to the sink's fidelity on the phenomenon of interest, u is an increasing function of the incoming packet rate t_i ($i = 1, 2, ..., m$), *i.e.*,

$$f(t_1, ..., t_i + 1, ..., t_m) > f(t_1, ..., t_i, ..., t_m) \qquad \forall i = 1, 2, ...m. \tag{2.2}$$

[3] *Fidelity* means how certain the phenomenon value obtained by the sink is. We also use the word 'uncertainty' as its opposite.

In case that the incoming packet rate from each source is t_j ($j = 1, 2, ..., m$), if we increase the reporting of the ith source by one, the additional fidelity obtained, denoted by δ_i, is computed as follows:

$$\delta_i = f(t_1, ..., t_i + 1, ..., t_m) - f(t_1, ..., t_i, ..., t_m) \quad \forall i = 1, 2, ...m. \tag{2.3}$$

In existing work (*e.g.*, [54]), reliability (*i.e.*, the fidelity on the phenomenon of interest) is often measured in terms of the ratio of the achieved total incoming packet rate to the desired incoming packet rate regardless of the sources of the incoming packets, which can be modeled in terms of $f(t_1, t_2, ..., t_m)$ as follows:

$$f(t_1, t_2, ..., t_m) = \gamma \sum_{i=1}^{m} t_i, \tag{2.4}$$

where γ is a constant.

But this consideration is not adequate. Total incoming packet rate is not a good indicator of how reliable the sensor-to-sink data transport is. Take the scenario in Figure 2.1 as an example. The sink is interested in the physical phenomenon P. The sensor nodes A, B and C, which can detect P, are instructed to collect information about P and report data on P to the sink. In this scenario, node A is nearer to the physical phenomenon than node B and node C. In most application cases, measurement error is a monotonically increasing function of the distance between the sensor and the phenomenon. Node A may thereby measure the phenomenon data with less error and provide higher certainty of the phenomenon value than node B and node C. In this scenario, if the sink receives a given number of packets, its fidelity on the phenomenon is related to the proportions of the packets sent by different sources.

Moreover, according to Equation (2.3) and Equation (2.4), δ_i is considered constant, which is not true in most application

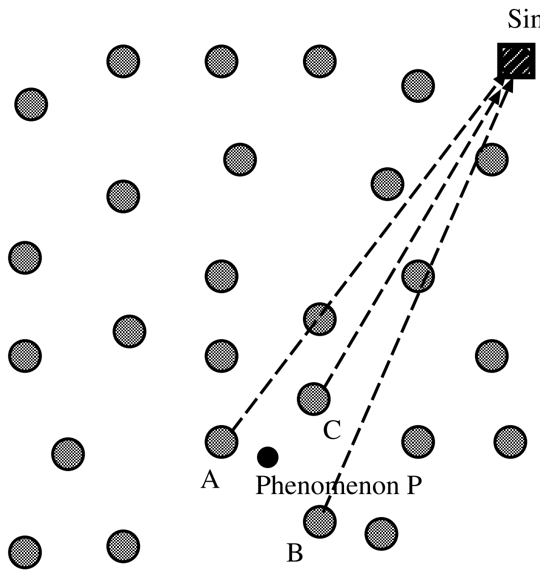

Figure 2.1: A scenario of the WSN.

cases. δ_i is usually a decreasing function of t_i. The reporting packets from source i in one time unit contain redundant information on the measuring phenomenon. The higher t_i is, the higher the source reporting rate is required, and as a result, the more the redundant information packets from source i contain, which consequently causes δ_i to decrease.

According to the above considerations, obviously, to save energy, it is better for WSNs to bias packet reporting rates of source sensor nodes according to their current contributions to improve the sink's fidelity on the phenomenon of interest. Therefore, a reliable sensor-to-sink data transport scheme should provide the sink with a mechanism to adjust the reporting rate of each data source dynamically in a discriminative manner.

But, is the contribution of each source node the only factor that influences the decision about source reporting rates? Again, consider the example scenario in Figure 2.1. If the current fidelity u of the phenomenon is lower than the acceptable fidelity u', we should increase the source reporting rates so that the sink can obtain higher fidelity.

Assume increasing the packet reporting rate of node A by r_1 or increasing the packet reporting rate of node C by r_2 ($r_2 > r_1$) can make the fidelity higher than the acceptable fidelity. Although the sink needs to increase the packet rate from node A by less than that from node C to make the fidelity acceptable (because, say, packets from A have higher contribution to decrease uncertainty), increasing node A's reporting rate to reduce uncertainty may not be a better solution in terms of minimizing energy consumption. This is because increasing the reporting rate of node A may counter-intuitively require more energy consumption than increasing the reporting rate of node C if the communication cost from node A to the sink is much higher than that from node C to the sink. Especially when the path from node A to the sink suffers from high packet loss rate, *e.g.*, due to congestion, much energy will be consumed to convey packets along this path.

We propose that source-reporting rates should be decided based on an optimization approach. This sink should determine the reporting rates of sources so that the energy consumption of the WSN is minimized, subject to the constraint that the fidelity of the phenomenon knowledge cannot exceed a given tolerable minimum value. Formally,

$$\begin{aligned} &\mathit{minimize} \quad && \sum_{i=1}^{m} (t_i \times p_i) \\ &\mathit{subject\ to} \quad && u = f(t_1, t_2, ..., t_m) \geq u' \end{aligned} \tag{2.5}$$

where p_i is the communication cost (*i.e.*, the energy consumed to successfully deliver a packet) for each source i to the sink.

As $f(\cdot)$ is application-related, how to determine it and how to solve the optimization problem are beyond the scope of our protocol design. Note that only the sink (*i.e.*, where the application runs) is required to solve this optimization problem. It would not cause any energy overhead at in-network sensor nodes.

Although solving the above optimization problem is the task of applications, it is vital for a reliable sensor-to-sink data transport protocol to provide information about the communication cost p_i from each source to the sink, so that the sink can properly decide the reporting rates.

Another important merit of providing end-to-end communication cost is that it can offer a congestion control mechanism. As congestion causes high communication costs, it can be alleviated with a discriminative source-rate control mechanism provided by a reliable sensor-to-sink data transport scheme. The sink can slow down sources that cause congestion and speed up sources with lower communication costs. In the meanwhile, enough fidelity can still be obtained based on the optimization approach discussed above.

In summary, to assure that the sink can obtain enough fidelity of the knowledge on the phenomenon of interest and achieve energy-efficiency, it is necessary for a reliable sensor-to-sink data transport protocol to provide two mechanisms which is listed as follows:

- A dynamic and discriminative source reporting rate feedback mechanism, allowing the sink to adjust the reporting rate of each data source.

- A mechanism to provide the sink with the current end-to-end communication cost from each source to the sink.

Note that we intend to violate the common *layering* network-protocol principle by somewhat coupling data transport protocol and applications (*i.e.*, let applications solve an optimization problem and feed back required reporting rates of sources). This is based on the features of WSNs. A WSN is usually employed to conduct one or a few specific tasks; *i.e.*, only one or a few specific applications are running at the sink. Traditional layering concept aims at general purpose protocol design. In transport layer design, it aims to provide data transport service for various applications. However, strict layering is not necessary in WSN because applications of a network are always deterministic before a network is set up (whereas, they are not deterministic in traditional networks). Moreover, it is even worse as it would cause much protocol overhead. Violating layering principle and utilizing application information in data transport protocol design can let the application facilitate the data transport protocol to save energy, which is an important merit of this work.

2.4 Design Considerations

2.4.1 The concept of node price

As wireless communication consumes most of the energy in WSNs, the energy consumption of local computation at each node can be ignored [22]. Also, even though the packet size of each packet may be dynamic, the inevitable large overhead of the physical layer implementation of traditional wireless communication schemes makes the energy consumption of each packet transmission attempt nearly constant. So, we consider the total number of transmission attempts of the nodes required to successfully deliver a packet as the metric to evaluate the energy cost of the communication. The formal definition is as follows.

The price of a node n is, the total number of transmission attempts all in-network nodes have made to successfully deliver a packet from node n.

We denote the node price of node n as $NP(n)$. Obviously, node price is determined by the price of its downstream neighbors, the link-loss rate between the node and its downstream neighbors, the end-to-end packet loss rate from its downstream neighbors to the sink, and the proportion of the outgoing traffic allocated to each downstream neighbor. Table 2.1 describes the symbols employed during our following discussions.

n_i	The ith downstream neighbor of node n
$NP(n_i)$	The node price of node n_i
$\omega(n, n_i)$	The proportion of node n's outgoing traffic that is routed to its downstream node n_i
$p(n)$	End-to-end packet loss rate from node n to the sink
$p(n_i)$	End-to-end packet loss rate from node n_i to the sink
$h(n_i, n)$	Link packet loss rate from node n to its downstream node n_i

Table 2.1: The descriptions of the symbols

Now we derive the node price of each in-network node in a recursive way. Consider node n sends out $\mathcal{N}$ packets via its downstream neighbors to the sink. The number of packets that can successful reach neighbor n_i is:

$$\mathcal{N} \cdot \omega(n, n_i) \cdot (1 - h(n_i, n)), \tag{2.6}$$

in which the number of packets that can successfully reach the sink, denoted by $\mathcal{N}_i$, is:

$$\mathcal{N}_i = \mathcal{N} \cdot \omega(n, n_i) \cdot (1 - h(n_i, n)) \cdot (1 - p(n_i)). \tag{2.7}$$

Therefore, according to the definition of the node price, the total number of transmission attempts that all in-network nodes have made to successfully deliver $\mathcal{N}_i$ packets from node n via the path along node n_i is:

$$\mathcal{N}_i \cdot NP(n_i) + \mathcal{N} \cdot \omega(n, n_i). \tag{2.8}$$

The total number of packets that can successfully reach the sink is:

$$\sum_{\forall i} \mathcal{N}_i = \sum_{\forall i} \{\mathcal{N} \cdot \omega(n, n_i) \cdot (1 - h(n_i, n)) \cdot (1 - p(n_i))\}. \tag{2.9}$$

The total number of transmission attempts that all in-network nodes have made to successfully deliver $\sum_{\forall i} \mathcal{N}_i$ packets is:

$$\sum_{\forall i} [\mathcal{N}_i \cdot NP(n_i) + \mathcal{N} \cdot \omega(n, n_i)]. \tag{2.10}$$

According to Equations (2.6)−(2.10), we can calculate $NP(n)$ as follows:

$$NP(n) = \frac{\sum_{\forall i} [\mathcal{N}_i NP(n_i) + \mathcal{N}\omega(n, n_i)]}{\sum_{\forall i} \mathcal{N}_i} \tag{2.11}$$

$$= \frac{\sum_{\forall i} \{\omega(n, n_i)[(1 - p(n_i))(1 - h(n_i, n))NP(n_i) + 1]\}}{\sum_{\forall i} [\omega(n, n_i)(1 - p(n_i))(1 - h(n_i, n))]}. \tag{2.12}$$

The end-to-end loss rate from node n to the sink $p(n)$ is:

$$p(n) = 1 - \frac{\sum_{\forall i} \mathcal{N}_i}{\mathcal{N}}$$

$$= 1 - \sum_{\forall i} \{\omega(n, n_i) \times [(1 - p(n_i)) \cdot (1 - h(n_i, n))]\} \tag{2.13}$$

As the traffic ends at a sink, the sink always has $NP(sink) = 0$ and $p(sink) = 0$.

If the link packet loss rates along all the paths of the sensor-to-sink traffic can be obtained with a hop-by-hop feedback mechanism along the reverse direction of the sensor-to-sink traffic, any node n along the path can calculate its $NP(n)$ and $p(n)$ according to Equation (2.12) and Equation (2.13) based on $NP(n_i)$ and $p(n_i)$ fed back by its downstream nodes n_i and its outgoing traffic allocation scheme $\omega(n, n_i)$.

Because of the dynamic nature of the WSN traffic, the link-loss rate is a dynamic variable. Accurate and up-to-date hop-by-hop loss rate estimation is necessary to ensure that the price of a node represents the real downstream communication conditions. We will discuss how to obtain the link-loss rate in Subsection 2.4.2 and the routing scheme that determines $\omega(n, n_i)$ in Subsection 2.4.3.

2.4.2 Link-loss rate estimation

There are three situations in which the communication load may change. The first one is that a new task is assigned and the responsible sensor nodes begin to report packets. The second one is that the sink requests the source nodes to change their reporting rates. The last one is that some in-network nodes decide to change their routing scheme, *e.g.*, a node may begin to send more packets to a downstream neighbor when it finds that the price of the neighbor has become smaller. It is reasonable to estimate link-loss rate based on an EWMA (Exponentially Weighted Moving Average) approach.

We base our link-loss rate measurements upon the sequence of the arrival packets' serial numbers (SN). Every node n sends packets to each downstream neighbor n_i with consecutively increasing SN. The receiver, *i.e.*, node n_i can measure the link-loss rate according to the missing SN. Then we can calculate the link-loss rate with an EWMA approach. Formally,

$$h(n_i, n) = \alpha \times h_{-1}(n_i, n) + (1 - \alpha) \times h'(n_i, n) \tag{2.14}$$

where $h_{-1}(n_i, n)$ is the previous estimate of link-loss rate; $h'(n_i, n)$ is the link-loss rate according to current sampling result; and α is a weighting factor, whose value is selected empirically according to the traffic features of WSNs. Generally speaking, α should be set close to 1 when we have *a priori* knowledge that the traffic of the WSN is stable.

As congestion will increase packet loss rate, the link-loss rate gives a good indication of the congestion condition. The communication cost metric, *i.e.*, node price, calculated from the packet loss rates, is therefore also influenced by congestion. As a result, reporting rate control and routing based on node price can provide a good congestion avoidance mechanism.

2.4.3 Routing scheme

As the price of a node determines the energy efficiency of the communication between the node and the sink, the nodes can make a local optimal decision on where to route packets to minimize their prices. If the in-network node finds that its outgoing traffic is not fully allocated to the current best downstream path (*i.e.*, the traffic is not 100% sent to the preferred downstream neighbor to achieve the smallest local NP), the node will shift the outgoing traffic that is currently allocated to the other downstream neighbors to the best downstream neighbor. Obviously, such an optimization approach always allocates all the traffic to one best path.

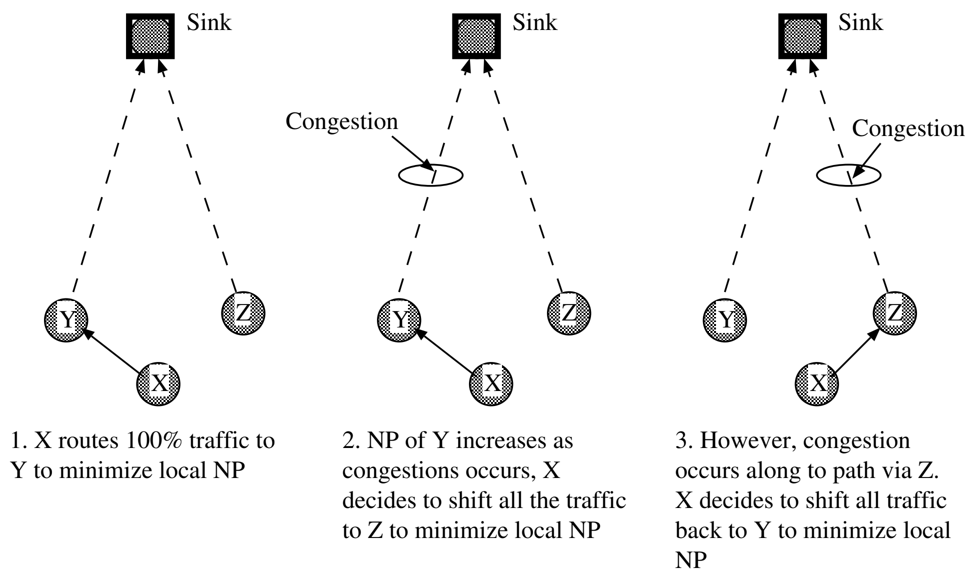

Figure 2.2: An example showing oscillation

However, the price of a node's downstream neighbors might vary with the change of the node's outgoing traffic allocation.

In the worst case, the dynamics of the downstream neighbors' price caused by the node's outgoing traffic allocation change will result in fast routing oscillation. An example is shown in Figure 2.2. When node X routes all traffic via its neighbor Y (scenario 1 in the figure), the path to the sink via Y may get congested. The packet loss rate along the path will increase and so the price of Y will increase (scenario 2 in the figure). Node X, to minimize its own price, will then shift all the outgoing traffic to Node Z. As a result, the path via Z will get congested and the price of Z will increase (scenario 3 in the figure). Node X has to shift all the traffic back to node Y to achieve minimal price. Oscillation is inevitable in this example scenario.

Such oscillation caused by interaction of traffic loads and path cost (in our protocol, it is node price) is a notorious routing problem in data networks [7]. Fortunately, since we can have more than one outgoing path at a time, we can avoid such a fast oscillation by shifting the traffic to the new detected best downstream path in a gradual manner. Let us denote the current proportion of outgoing traffic allocated to the bad downstream node (a 'bad' downstream neighbor means that routing through it causes high node price comparing to routing through a 'good' one) with a higher price NP_{high} as $\omega(n, high)$, and denote the price of the good downstream node as NP_{low}. The proportion of traffic that will be shifted from the bad node to the good node in each decision interval is:

$$\omega(n, high) \times \frac{NP_{high} - NP_{low}}{NP_{high}} \tag{2.15}$$

This scheme assures that the more the difference between the prices of the downstream nodes, the more traffic would be shifted each time. The network can thus adapt to the communication condition changes and avoid fast oscillation with a proper decision period.

If congestion of one selected path occurs, the node price of the neighbor in that path will increase. The node will gradually shift outgoing traffic to a new best path. This scheme could result in an increase of the node's price. If the new best path never gets congested because of the traffic shift, the node will locally avoid congestion by eventually allocating all traffic to the new best path. Otherwise, because the price of the node will eventually influence the node price of the source that sends packets via this node, the sink can decide to slow down the source that keeps sending packets to the congested path and speed up another source, using the rate control scheme provided by PORT.

2.5 Protocol Description

When a new task is assigned, PORT employs a similar routing information establishment mechanism to directed diffusion [30] by flooding the task description packet (called *interest* in [30]) to achieve the in-network nodes' neighborhood information. After the task assignment phase, the nodes in the WSN begin to report data packets to the sink if the physical phenomenon of interest can be sensed. The outgoing traffic allocation of a node can be dynamically adjusted during the reporting period according to the feedback about downstream communication conditions sent by its downstream neighbors. The sink also feeds back new reporting rate requirements to source nodes. We elaborate our detailed protocol implementation as follows.

2.5.1 Task initialization

We employ a reactive routing approach: the sink initiates a task by flooding its interest on some physical phenomenon. The nodes' neighborhood information is initialized as the interest packet travels throughout the network. A node's price is initially set to be the hop number between the node and the sink, and all the loss rates are considered to be zero. The nodes that are responsible for reporting data begin to report at the desired rate described in the interest packet. In order to ensure that the traffic pattern is changed in a gradual manner, the initial desired reporting rate is cautiously set to a very small value in the interest packet. After initialization, further adjustment will be conducted by the sink as described in the following subsection.

2.5.2 Feedback of newly desired source reporting rates

A source node encapsulates its node price in its reporting data packets. In this way PORT provides the node price of a data source to the application. If the application at the sink finds that the packets received per unit time provide more or less information on the physical phenomenon of interest than it desires, it will adjust the reporting rates based on an optimization approach. The new desired reporting rate of each source node is fed back to PORT by applications.

The feedback information is sent to the sources by PORT along the reverse path of the sensor-to-sink traffic. The rate control packets are inserted at the head of the sender nodes' queues and sent out with the highest priority. Such rate control packets can also be sent back directly to individual source nodes as implemented in ESRT [54] if the wireless interface of the sink is powerful enough.

2.5.3 Feedback of wireless communication condition

The sink, and the in-network nodes that are conveying the sensor-to-sink packets, estimate the link-loss rate from each of their upstream neighbors to themselves. The link-loss rate and their prices, as well as their end-to-end path loss rates (from them to the sink), are checked in a given time interval. If they find that these values have changed, the new values are fed back to their upstream neighbors. These feedback packets are inserted to the head of the nodes' queues and sent out with the highest priority.

Upon receiving a communication condition feedback packet from a downstream neighbor, a node will re-allocate its outgoing traffic as discussed in Subsection 2.4.3 if it finds that the current traffic allocation cannot achieve the local lowest price. The new price and path loss rate are calculated according to Equation (2.12) and Equation (2.13).

2.5.4 Fault tolerance and scalability considerations

In the case that a node dies (silently quitting the task), its upstream neighbor should shift the traffic routed via this node to other nodes immediately. We employ a timer on each node to detect the quitting of its downstream nodes. For each timeout occurrence, if a node fails to receive any feedback information from a downstream neighbor, it considers the downstream neighbor has failed and set the price of the neighbor as infinite to avoid routing packets to it.

If a new node (which could be a newly awakened node, a newly deployed node, or a node that recovered from a previous failure) detects an ongoing task, it might decide to join the routing task. In this case, the node will broadcast to its neighbor nodes to inform them that it is up. Neighboring nodes will send it their prices. The node selects some nodes with the lowest prices as its downstream neighbors and sends its own calculated price and the path loss rate to those neighboring nodes with larger prices. Upon receiving this information, those neighbors with larger prices will consider the node as a possible downstream neighbor. In this way, the new node joins the routing task.

2.6 Protocol Evaluation: A Case Study

To verify PORT, we code it over NS-2 [23]. As discussed above, PORT is employed to facilitate the sink to achieve reliability. To perform simulations, an application model should be specified. Although we verify PORT in a given application case, note that more sophisticated models could be employed in real world applications. The performance of PORT is surely influenced by the application, as it is the application that determines the reporting rates of the sources. The aim of our simulations is to show that with a proper decision on source reporting rates, PORT can effectively facilitate the sink to achieve energy-efficiency and maintain reliability.

Without loss of generality, PORT can be applied in many application scenarios for energy saving. The prerequisite is that the application should determine reporting rates of sources dynamically according to the data reported by the sources and the communication cost reported by PORT.

2.6.1 Simulation model

In our application scenario when conducting simulations, the sink is interested in a phenomenon with physical position (x, y). m nodes that are close to the phenomenon measure the physical value of that phenomenon and report each measurement value with a packet sent to the sink. For simplicity, assume that the jth measurement value of node $i(i = 1, 2, ..., m)$, denoted by $s_{i,j}$, is one-dimensional. The measurement model is

$$s_{i,j} = X + e_{i,j} \tag{2.16}$$

where X is the true value of the phenomenon parameter; $e_{i,j}$ is the error of the jth measurement of node i. Assume $e_{i,j}$ $(j = 1, 2, ...)$ are Gaussian-distributed with zero mean and with standard deviation v_i. v_i is related to the physical distance d

between node i and (x, y). For simplicity, we set it as follows, which means that the uncertainty of each measurement is directly proportional to the square of the distance d.

$$v_i = 0.0001 \times d^2 \tag{2.17}$$

The sink fuses the data received from node i in one second by calculating the mean of them (we denote the incoming packet rate from node i as t_i). The sink then calculates the average of the fused result of each node as the value of the phenomenon.

$$\frac{1}{m} \cdot \sum_{i=1}^{m} \left(\frac{1}{t_i} \cdot \sum_{j=1}^{t_i} s_{i,j}\right) \tag{2.18}$$

Thus, the sink's uncertainty υ on the value of the phenomenon is calculated as the standard deviation of the error:

$$\upsilon = \sqrt{\frac{1}{m^2} \cdot \sum_{i=1}^{m} \frac{\theta_i^2}{t_i}} \tag{2.19}$$

where θ_i is the standard deviation of t_i measurements (*i.e.*, $s_{i,j}$, $\forall j = 1, 2, ..., t_i$) of source i, obtained statistically.

In our simulations, we compare two sensor-to-sink data communication protocols: one is a directed-diffusion-based shortest path routing scheme with an ESRT-like unbiased report rate control approach (denoted as scheme 1); the other is PORT (denoted as scheme 2).

The original locations of the sensor nodes are in a grid-like way shown in Figure 2.3. For each simulation (*i.e.*, for each location of the phenomenon point), we change the location of each sensor node (except the source nodes and the sink) *randomly in a uniform manner* in a $100 \times 100m$ square which centers on its original location shown in Figure 2.3 for 20 times. We average the simulation results for each settings of node locations.

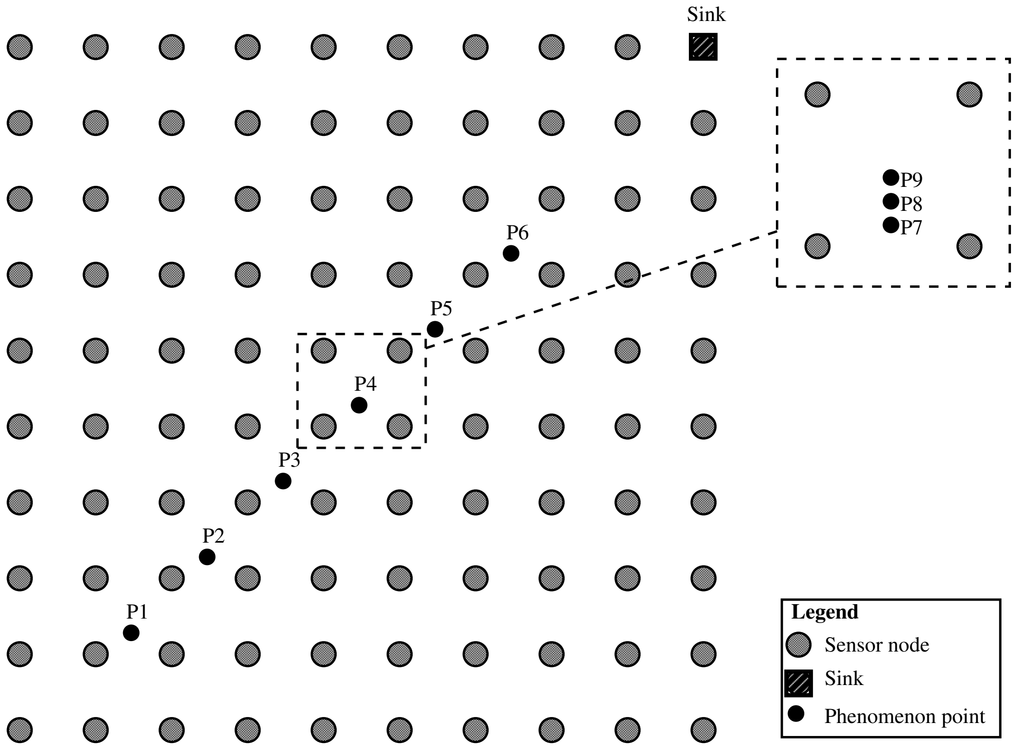

Figure 2.3: The simulation network.

The sink is in the top-right corner of the network. The wireless parameter settings are the same as the study of directed diffusion [30]. Detailed settings of the simulation network are shown in Table 2.2.

2.6.2 Energy consumption comparison

To study the total energy consumptions, we set the phenomenon at six different positions marked by $P1 - P6$ in Figure 2.3. For each setting, a set of different uncertainty values are required by the sink. For each uncertainty requirement, the four nearest nodes report their measurements to the sink in 500 seconds. Figure 2.4 shows the total energy consumptions of the whole network under these two protocols given different uncertainty requirements when the phenomenon points are at different po-

Area of sensor field	1350m*1350m
Number of sensor nodes	100
MAC	IEEE 802.11 without CTS/RTS and ACK
Radio power	0.2818 W
Packet length	36 bytes
Transmit power	0.660 W
Receive power	0.395 W
IFQ length	50 packets
Simulation time at each setting	500 seconds
Feedback / decision period	1 second

Table 2.2: Simulation network settings.

sitions.

The results show that PORT can save 10% to 30% of the energy consumption, compared to an existing scheme which employs unbiased source reporting rate control. This is not surprising, as PORT biases the reporting rates of the sources according to their contributions to reduce the uncertainty of the phenomenon value and their prices, which is a more energy-efficient approach.

PORT saves more energy when a smaller uncertainty is required. This is because, when a small uncertainty is required, large source reporting rates are needed. As a result, traffic load is high. Packet loss rate along the sensor-to-sink path is then also high. PORT can allocate traffic to alleviate congestion. In this case, PORT saves much more energy than the existing scheme.

Moreover, the results show that PORT can satisfy a smaller uncertainty requirement (uncertainty requirements less than 0.12 in Figure 2.4(a) and Figure 2.4(b), 0.11 in Figure 2.4(c), 0.10 in Figure 2.4(d) and Figure 2.4(e), and 0.09 in Figure 2.4(f)). In the very small uncertainty requirement cases, large source

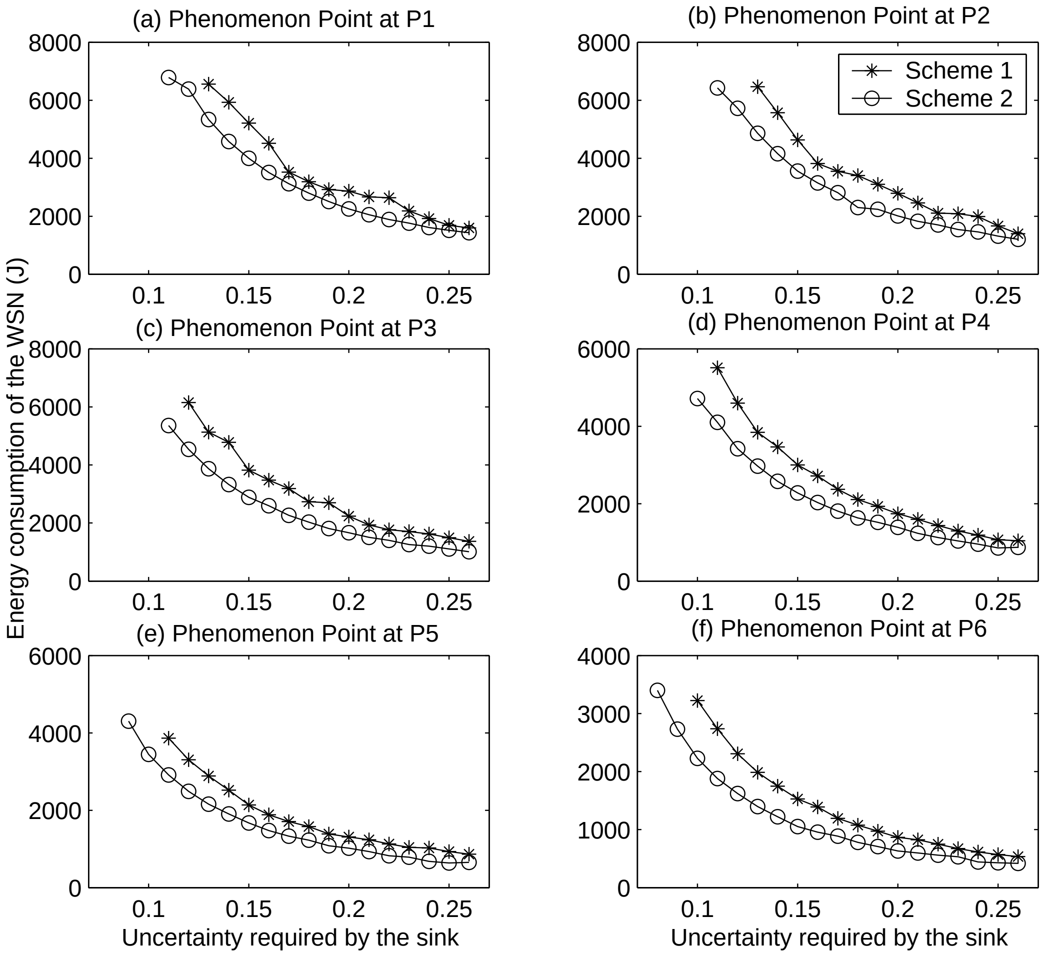

Figure 2.4: Energy consumption comparisons.

reporting rates of the sources overload the network capacity. The network severely congests and thus cannot provide the sink with enough packets. The uncertainty requirement cannot then be fulfilled. As PORT can alleviate congestion by routing via different paths, it allows higher reporting rates than existing schemes and hence it can fulfill a smaller uncertainty requirements. It shows that PORT provides a better congestion avoidance scheme.

2.6.3 The impact of reporting sensors' uncertainty distribution

To study the impact of the reporting sensors' uncertainty distribution, we set the phenomenon point at three different places in the network grid marked by $P7 - P9$ in Figure 2.3. Also, four nodes in the corners of the grid are reporting their measurements. Note that the closer the phenomenon point to the center of the grid, the more similar are the contributions of the four sources. Figure 2.5 shows the total energy consumptions of the whole network under these two protocols given different uncertainty requirements and different phenomenon positions.

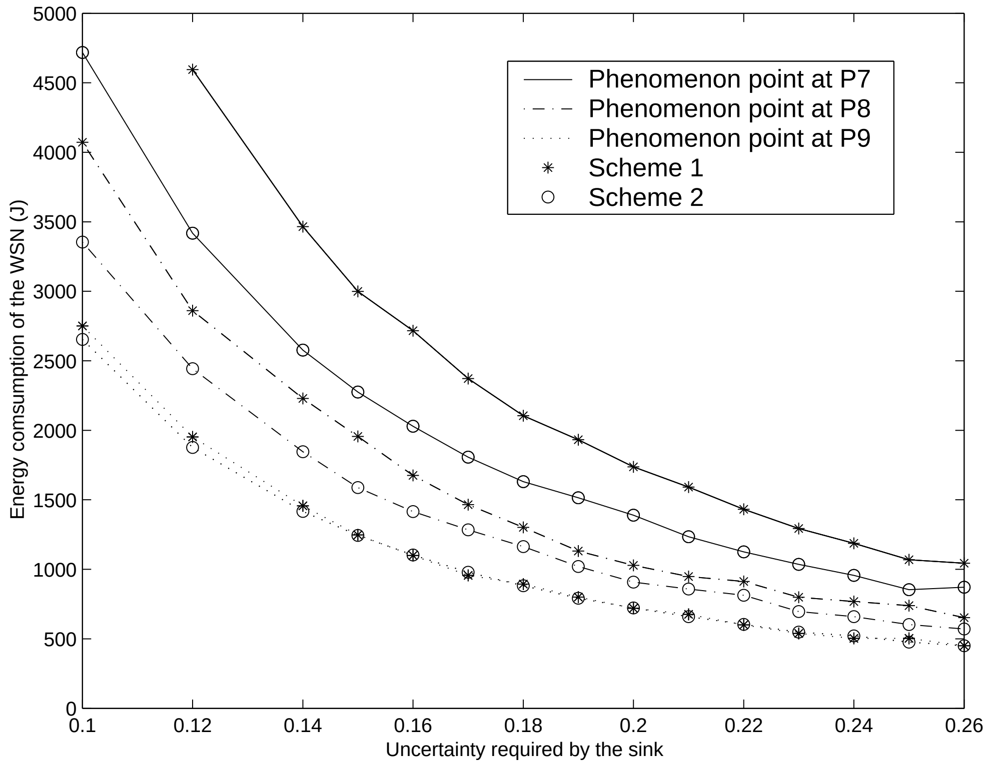

Figure 2.5: Energy consumption comparisons: different phenomenon positions in a grid

These results show that PORT can save more energy if the contributions of the sources are more different. PORT achieves little improvement when the sources have the same contributions. This is not surprising, as PORT biases the reporting rates of sources according to the sources' contributions to reduce the sink's uncertain of the phenomenon value. When the sources' contributions are almost the same, PORT will adjust the reporting in an almost unbiased manner, like existing schemes. Their energy consumptions, as a result, are almost the same.

2.7 Conclusion

This chapter proposes PORT, a price-oriented sensor-to-sink data transport protocol for wireless sensor networks. Under the constraint that the sink must obtain reliable information on the phenomenon of interest, PORT minimizes the energy consumptions using two schemes. One is based on the sink's application-based optimization approach that feeds back the optimal reporting rate of each source according to the contribution of the sources and the energy consumption of the sensor-to-sink communication from each source to the sink. The other is a locally optimal routing scheme for in-network nodes according to feedback of downstream communication conditions. The communication conditions estimation is based on an estimation of link-loss rate along the sensor-to-sink traffic path. PORT can obtain the sensor-to-sink communication condition such as congestion and weak link which cause packet loss, and thus it adapts well to network dynamics caused by these factors.

We code PORT on the NS-2 network simulation tool. Simulation results in an application case study demonstrate that PORT is an effective transport protocol for reducing energy consumption comparing to existing schemes. Thus, it can prolong the life time and reliability of wireless sensor networks.

□ **End of chapter.**

Chapter 3

Setting Up Energy-Efficient Paths

Summary

Energy-efficiency is an important design consideration of communication schemes for wireless sensor networks (WSNs). In this chapter, we investigate the problem of energy-minimized sensor-to-sink communications with adaptive transmitter power settings. We devise a novel network- and application-aware model for this problem, and present a Broadcast-On-Update (BOU) solution. However, BOU suffers from the high overhead due to explosive broadcasting in path setup. We then show a waiting scheme, BOU-WA, that effectively mitigates the broadcast explosion. In BOU-WA, the waiting time before each broadcast is proportional to the probability that a node could find a more energy-efficient path to the sink. We provide an efficient approximation algorithm to calculate this probability. The performance of BOU-WA is evaluated under diverse network configurations, and the results demonstrate its superiority in conserving energy. This chapter is based on the work presented in [78].

Although WSNs have diverse task-specific requirements, many of them rely on a sensor-to-sink communication scheme to transfer the information that is collected by the sensors to a sink node.

In general, the battery in a sensor node is limited and not rechargeable [22]. Since wireless communications consume most of the energy in typical WSN applications [19], an energy-efficient data communication scheme is greatly desired. To be energy-efficient, the data communication scheme to convey the desired information on the event of interest through the established sensor-to-sink paths should cost as low energy as possible.

One important approach to save communication energy consumption is to perform transmitter power control (which is also called topology control). Obviously, setting the wireless transmitter power of each sensor node in different levels will result in different network topologies, as the neighboring nodes that a node can directly reach are determined by the node's transmitter power setting.

A topology control scheme enables each node to set its power level to a minimum value under the constraint that the packet sent by this node could just reach its intended neighboring node. The energy consumption of data communication can thus be reduced. Transmitter power control is an important technique to save the energy consumptions of sensor nodes and prolong the lifetime of a network.

The prerequisite of transmitter power setting scheme is that each sensor node can set its own wireless transmitter power level. This is true in typical sensor node implementations. For example, the Berkeley Mica Mote [29] provides such program interfaces.

The notion of transmitter power control (topology control) has been extensively studied in wireless mobile ad hoc networks (MANETs). [55] is a good bibliography in the field. Although much work is done (*e.g.*, [15, 50, 53, 56, 57, 71]) on transmitter power control in MANETs and WSNs, researchers mostly focus on network connectivity analysis, network lifetime (a network is alive if it is 'somehow' connected) analysis. They usually propose transmitter power setting schemes for energy-efficient communications between an arbitrary node pair and for energy-efficient broadcasting and multicasting.

Our objective of transmitter power setting is to achieve energy-efficient sensor-to-sink data communications. We model the transmitter power setting problem based on this consideration and the network and application features of WSNs. With this analysis and modeling work, we investigate implementation issues and analyze the proposed schemes to solve the transmitter power setting problem.

The contributions of our work are twofold. First, transmitter power setting problem is studied in this chapter for achieving energy-efficient sensor-to-sink data communications. We do not emphasize to construct an energy-efficient communication path between an arbitrary node pair. This is the main consideration of the transmitter power setting problem in MANETs, as the MANET traffic is mainly unicasting peer-to-peer traffic. Instead, we aim at finding an energy-efficient communication path between an arbitrary node and a given node, *i.e.*, the sink. This can greatly simplify the complexity of the problem. We show that this problem is tractable .

Second, we tailor the solution of the problem to adapt to the features of WSNs. We investigate the implementation issues for setting up the energy-efficient paths for sensor-to-sink traffic. Although high node density and large network scale of WSNs are major challenges for algorithms that set up each node's transmitter power level, we provide a low-overhead algorithm to address the transmitter power setting problem.

The rest of this chapter is organized as follows. In Section

3.1, we model our transmitter power setting problem according to the network and application features of WSNs. Section 3.2 investigates the implementation issues and analyzes the algorithm that solve the transmitter power setting problem. In Section 3.3, we present our simulation results. Section 3.4 discusses the related work of this research. Section 3.5 provides conclusion remarks and our future directions.

3.1 Transmitter Power Setting for Energy-Efficient Sensor-to-Sink Data Communications

3.1.1 Network, communication, and energy consumption models

The wireless signal fading models investigated in the literature [51] give the condition that packets transmitted from node u can be successfully received by the destination node v if the transmitter power setting of node u satisfies the following inequation:

$$Pr(u) \geq c \cdot (D(u, v))^n \tag{3.1}$$

Here c is a constant whose value is related to the system parameters such as the wavelength of the wireless signal, the antenna gains, and the threshold that a signal can be successfully detected in the destination node. n is the signal fading factor whose value is typically in the interval $(2, 5)$ in an application environment. $D(u, v)$ is defined as the physical (Euclidian) distance between node u and node v,

$$D(u, v) = \|X(v) - X(u)\|, \tag{3.2}$$

where $X(\cdot)$ denotes the physical location of a node.

Our work is based on this model: If node u knows the locations of itself and its one-hop destination node v, the optimal transmitter power setting for node u to send a packet to node v is computed as:

$$Pr(u) = c \cdot (D(u,v))^n = c \cdot \|X(v) - X(u)\|^n \tag{3.3}$$

We assume that each sensor node can know its approximate physical location with which we can calculate the transmitter power setting. The approximate location information is achievable if each sensor node carries a GPS receiver or if some localization algorithms are employed (*e.g.*, [9]).

We model the network as a graph. Let $G(V, E)$ be the graph constructed by the sensor nodes in a d-dimensional space where V is the set of vertices which are the sensor nodes[1] and E is the set of edges that are the wireless links connected by the pairs of the sensor nodes which can communicate with each other at the maximum power setting Pr_{max}. Denote s ($s \in V$) as the sink node which is the final destination of the sensor-to-sink traffic.

Let $P(V) = \{Pr(u)$: for each $u \in V\}$ be the transmitter power setting scheme for the sensor-to-sink communications. $P(V)$ should assure that each node can send packets (possibly, in a multi-hop manner) to the sink s.

Denote edge set E' as the set of the wireless links under the transmitter power setting scheme $P(V)$. Obviously, each $Pr'(u)$ in $P(V)$ is not larger than Pr_{max}. Therefore, the graph $G'(V, E')$ is the subgraph of $G(V, E)$.

Note that $G'(V, E')$ is a directed graph. With the transmitter power setting scheme $P(V)$, node u can send packets to node v if $Pr(u)$ satisfies Equation (3.1) and thus $\overrightarrow{e'}(u, v)$ is formed. But $Pr(v)$ may not necessarily satisfy similar requirements and thus $\overrightarrow{e'}(v, u)$ may not be formed. This consideration is because

[1]The terms 'vertex' and 'node' both refer to a sensor device. In the rest of this chapter, they are used interchangeably.

the power level of a node's downstream neighbor is not necessarily larger than that of the node to let the downstream neighbor respond acknowledgement (ACK) packets to the node, as hop-by-hop packet ACK mechanism (*i.e.*, the packet ACK mechanism in MAC layer) is usually not employed for energy saving. Note that if a hop-by-hop packet ACK mechanism has to be employed, we can simply adjust a node's transmitter power to different levels: One for sending sensor-to-sink packets to its downstream neighbor; the other for sending ACK packets to the upstream neighbor. The technique is trivial and we do not discuss it in the rest of this chapter.

Let $\overrightarrow{\ell}(u_1, s)$ (where u_1 is the source node and s is the destination node) be the path in $G'(V, E')$ along u_1, u_2, ..., u_i, s $(u_1, u_2, ..., u_i \in V)$. Consider the sensor-to-sink traffic path $\overrightarrow{\ell}$. We assume all sensor-to-sink packets are of the same size. The energy consumption of a sensor-to-sink packet delivery along this path is modeled as:

$$\begin{aligned}&\sum_{n=1}^{i}(\gamma Pr(u_n)) + \sum_{n=2}^{i}(\gamma Rr(u_n) + Ps(u_n))\\&+\gamma Rr(s) + Ps(s),\end{aligned} \tag{3.4}$$

where γ is a constant related to the packet size, $Rr(\cdot)$ denotes the receiver power of a node, and $Ps(\cdot)$ denotes the energy consumption to process this packet. We assume that the energy consumed to receive and process a packet of each node is the same. Equation (3.4) can then be written as:

$$\gamma \sum_{n=1}^{i}(Pr(u_n) + \beta) \tag{3.5}$$

where β is a constant related to the power consumed to receive and process a packet.

We define Equation (3.5) the *path cost* of the path $\overrightarrow{\ell}$, denoted by $\omega(\overrightarrow{\ell})$. $\omega(\overrightarrow{\ell})$ reflects the energy consumption of the communication along the path $\overrightarrow{\ell}$. We define the *node cost* of a node u ($u \in V$) as the minimum value of the path costs of the known possible paths from node u to the sink s. We denote $\eta(u)$ as the node cost of node u. $\eta(u)$ reflects the known minimum energy required to transfer a packet from node u to the sink.

3.1.2 Transmitter power setting problem for energy-efficient sensor-to-sink data communications

Since WSNs are employed to sense and convey the phenomenal data of interest, sensor-to-sink traffic dominates the traffic of the networks. In typical WSN applications, usually traffic sources are a set of sensor nodes responsible for reporting the data of some nearby phenomena of interest and the traffic destination is a given sink.

If no data fusion/aggregation approaches are employed, the sensor-to-sink traffic of the network is simply many-to-one traffic. The transmitter power setting scheme for the network, in this case, should aim to minimize the energy consumption of the communication between an arbitrary node and a given sink.

If some data fusion/aggregation approaches are employed, without loss of generality, the transmitter power setting scheme for the network should still aim to minimize the energy consumption of the communication between an arbitrary node and the given sink. The reasons are as follows.

First, in most application scenarios, we could simply consider the data fusion/aggregation center as the single data source that is reporting data packet to the sink in the transmitter power setting problem. Usually, the data fusion/aggregation center should be a sensor node located near the set of the source sensor nodes sensing and reporting the data on the physical phenomenon of interest. We can simply leave the consideration of how to report the data to the data fusion/aggregation center. Also, as data fusion/aggregation center consumes more energy than the other source sensor nodes, in practical applications, to avoid quickly draining the data fusion/aggregation center node, data fusion/aggregation center node should be selected in a rotational basis. (Details on how to select a data fusion/aggregation center and how to report data to the center are beyond the scope of this work.) This means that in a long term point of view, each sensor node could be voted as the center node. Therefore, minimizing the energy consumption of the communication between an arbitrary node to a given sink is desired in this case.

Second, if we cannot consider the data fusion/aggregation center as the single data source, then for a given set of the source nodes, the optimal transmitter power setting problem (i.e., how to minimize the total energy consumptions of the sensor-to-sink communications) is very hard to solve. It is a minimum Steiner tree problem [26] which is NP-Hard. An approximation algorithm is to minimize the energy consumption of the communication between each source node to the sink. The packets are fused/aggregated only at the nodes in which the paths from the source sensor nodes to the sink intersect [35]. In this case, the transmitter power setting scheme should still minimize the energy consumption of the communication between an arbitrary node to a given sink.

Based on the above discussion, to save energy consumption of the wireless communications, we do not have to consider how to minimize the sum of the energy consumed for the communication along a path between any arbitrary node pair. What we should consider, instead, is how to minimize the energy consumption of the communication between an arbitrary node to the sink. This consideration can greatly simplify the transmitter power setting problem. We model the transmitter power setting problem for energy-efficient sensor-to-sink data communications as follows.

Problem 1 *Given graph G(V, E) and a sink s ($s \in V$), compute $P(V)$ such that in the resulting subgraph $G'(V, E')$, there exists at least one path $\overrightarrow{\ell}(u, s)$ from each node u ($u \in V$) to the sink s and $\eta(u)$ is minimized.* ■

Denote $cost(e)$ and $cost(\overrightarrow{e})$ as the cost functions of edge $e(u, v)$ and $\overrightarrow{e'}(u, v)$ ($e \in E$, $e' \in E'$, and $u, v \in V$), respectively. $cost(e)$ and $cost(\overrightarrow{e})$ are defined as follows.

$$cost(e) = cost(\overrightarrow{e}) = \gamma(c \cdot (D(u, v))^n + \beta) \tag{3.6}$$

With the cost function, the shortest path from each node u ($u \in V$) to a given node s in the graph $G(V, E)$ can be found. The solution of Problem 1 is simply setting each node's transmitter power to the value with which it can just send packets to the downstream node along the shortest path.

With the information of the physical location of each node, $D(u, v)$ in Equation (3.6) can be calculated and thus $cost(e)$ can be obtained. Theoretically, the shortest paths can easily be found, for example, with the Dijkstra algorithm [25].

3.2 Setting Up the Transmitter Power Levels for Sensor-to-Sink Traffic

Although a theoretical algorithm to set up the transmitter power levels for energy-efficient sensor-to-sink data communications is simply based on the modeling work in Section 3.1, there are many practical implementation issues in WSNs that should be carefully considered.

Usually, the scale of WSNs is very large containing hundreds to thousands of sensor nodes. In order to obtain high reliability, the networks are usually with high density, *i.e.*, the number of each node's neighbors is large. Moreover, sensor nodes are usually deployed in a non-deterministic manner, which means the location of each node is not known *a priori*. It is therefore not feasible for each sensor node to achieve a global picture of the network (*i.e.*, graph G(V, E) and the location X(u) of each node u in set V) because exchanging the location information of every node is very expensive.

As each of the nodes does not have a global picture of the whole network, the shortest paths should be constructed with only localized information. We have to implement a solution to the problem in a completely distributed way, *i.e.*, each node u should determine who its downstream neighbor in the shortest path is with only localized information. Here a node's localized information means the information that can be obtained by a node from its one-hop neighbors (*i.e.*, its adjacent nodes in graph $G(V, E)$).

How to find out the downstream neighbor in the shortest path in an energy-efficient way (*i.e.*, exchange as small number of packets as possible) is a challenging implementation issue. We analyze and solve this problem in this section.

3.2.1 BOU: the basic algorithm

A direct way to set up the transmitter power level is broadcasting. Broadcasting is performed by setting the power level to the maximal value in order to reach all possible one-hop neighbors. We call the broadcast packets which carry the information to set up in-network nodes' transmitter power level the *configuration packets*. A configuration packet describes the location, the identity, and the node cost η of the node that sends the configuration packet.

The sink first broadcasts a configuration packet. The node cost of the sink is obviously set to zero. Upon receiving a configuration packet, an in-network node may update the node cost of itself and broadcast another configuration packet with the updated node cost.

Each node that receives a configuration packet computes its own wireless transmitter power setting with which it can reach the node where it receives the configuration packet according to Equation (3.3). Then the cost of the edge (the wireless link) from this node to the neighbor is calculated with Equation (3.6). The sum of the edge cost and the node cost of this neighbor is the path cost of the path from the node via this neighbor to the sink. If the node has not received any configuration packet before, this path cost is saved as its node cost. Otherwise, this path cost is compared with the current node cost. If the current node cost is smaller, the packet is simply dropped. Otherwise, the node cost is replaced with this path cost and this updated node cost is encapsulated in a configuration packet together with the node's location. The node then broadcasts the configuration packet.

It is straightforward to show that this process will finally converge and each node can know the location of its downstream neighbor through which the path to the sink is the shortest path. This process builds up a spanning tree rooted at the sink that initially sends out a configuration packet with node cost equal to zero. The path from each node to the sink in the spanning tree is the shortest path in graph $G(V, E)$ given the cost function of each edge e ($e \in E$) described in Equation (3.6). We call this approach *broadcast on update* (BOU) and formulate it in Algorithm 1 in Appendix B.

3.2.2 Packet implosion of BOU: the challenge

However, as mentioned before, a typical WSN is with high node density and with large number of nodes. In addition, the power level needed to communicate with a neighboring node is linearly related to the nth power of the physical distance to the neighboring node according to Equation (3.3). The BOU approach is surely not efficient. A major challenge encountered in this distributed implementation is that it might cause explosive broadcasting in the network. Let us take Figure 3.1 as an example. Note that to simplify our discussion, we adopt the ideal free space transmission model [51] (*i.e.*, $n = 2$ in Equation (3.6)), and we ignore the energy consumptions of receiving and processing a packet at the node inside area ϕ, where area ϕ is a circular area with diameter AC (AC is the segment from node A to node C).

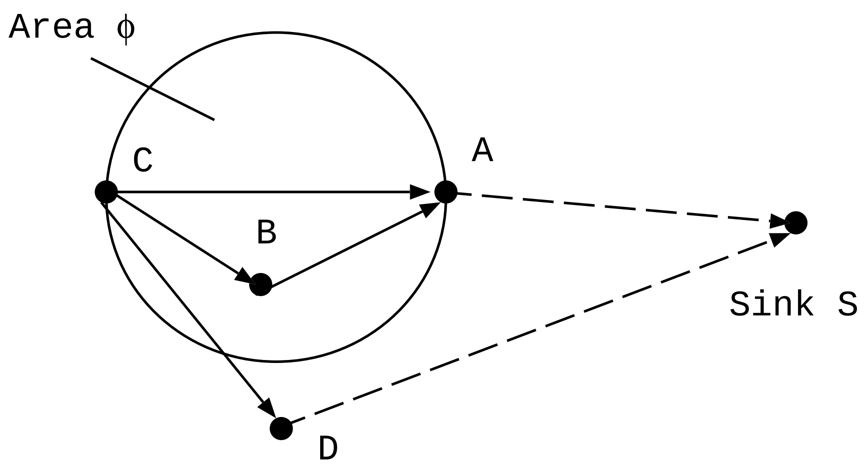

Figure 3.1: A scenario of a network

Suppose node A in Figure 3.1 broadcasts a configuration packet. Node B and node C will approximately receive the packet at the same time. Normally, the processing time of the packet in node B and node C is almost the same. If node B and node C both find out that the paths along node A to the sink are the current shortest path to the sink, node B and node C will broadcast their configuration packets almost at the same time with their node costs.

In the next step, node C will receive the configuration packet from node B and notice that the path cost of the path through node B to the sink is lower than that of the path directly through A to the sink (because $\|AB\|^2 + \|BC\|^2 < \|AC\|^2$). Node C thus has to update its node cost computed with the node cost received from node B. Therefore, node C has to broadcast again a configuration packet.

If the network is with high node density and large scale, similar scenarios would cause severe problems. An in-network node might have to update its node cost for many times. Broadcasting has to be performed upon each update of the node cost. This will cause explosive broadcasting of the network because the updated information is propagated in a tree-like manner to all upstream nodes of the nodes which have updated their node costs and broadcasted their configure packets.

Such kind of packet explosion should by all means be avoided to save the energy spent in configuring the transmitter power settings of the in-network nodes. Moreover, work in [38] enlightens us and strengthens our motivation to address this problem. In [38], the authors show that an *energy hole* around the sink is very likely to happen if the sink is fixed. We believe one easy way to avoid such energy hole is that we change the location of a sink frequently in the network area. In this case or in other application scenarios, when the location of the sink is not fixed during the network lifetime, the process to configure the optimal power setting of each node needs to be started each time the location of the sink changes. The efficiency of the implementation of the process therefore becomes a more critical issue. It is very desirable to address the aforementioned packet explosion problem.

One way to avoid the broadcast packet explosion problem is for each node to wait for a given period of time between the update of its node cost and the broadcasting of a configuration packet. An important research issue is therefore to determine this waiting time, which is investigated in the following subsection.

3.2.3 Determining the waiting time before broadcasting

The case that a node broadcasts a configure packet for more than one times happens only when the node needs to update its node cost after the first time it broadcasts a configuration packet. The reason of the update is that the node receives another configuration packet from a neighbor, which causes the change of its node cost. Therefore, if the waiting time before each in-network node broadcasts a configuration packet can be long enough, the node could have collected the configuration packet which indicates the actual shortest path from the node to the sink. It could avoid broadcasting for another time.

An intuitive solution is that each in-network node waits for the same period of time. But unfortunately, this idea does not work and the explosion situation still exists. For example, in the scenario described in Figure 1, node B and node C waits for the same time after they receive the packet from A. Node C will still receive the configuration packet from node B after it broadcasts the configuration packet with a node cost based on the node cost of node A. Then node C will update its node cost and it has to broadcast another configuration packet with the updated node cost.

We propose that the sophisticated waiting time should be proportional to the probability that a node will update the node cost in the future. Suppose that a node receives a configuration packet announcing a path $\overrightarrow{\ell}$ to the sink. It calculates the path cost $\omega(\overrightarrow{\ell})$ of this path. If $\omega(\overrightarrow{\ell})$ is smaller than the current node cost (if the node has not received any configuration packet before, the node cost is set to $+\infty$), the current node cost is updated to $\omega(\overrightarrow{\ell})$. The node should derive the probability that there exists another path $\overrightarrow{\ell'}$ to the sink whose path cost $\omega(\overrightarrow{\ell'})$ is lower than $\omega(\overrightarrow{\ell})$, and waits for a period of time that is proportional to this probability. Now the problem left is how to calculate this probability.

Although a node cannot have a global picture of the network, if the node deployment scheme of the network is known, the probability distribution of a node's location can be modeled. For example, we can model this distribution as a uniform distribution if the sensor nodes are deployed randomly in a uniform way. Furthermore, the location $X(u)$ of each node u can be regarded as independent and identically distributed random variable if the deployment scheme of each in-network node is identical and independent of the others. In our following discussions, we regard $X(u)$ as independent and identically distributed random variable with probability density function $P_x(X)$.

The problem of computing the probability that a node will update the node cost in the future is formulated as follows.

Problem 2 *Given*

- *A graph G(V, E), a sink s ($s \in V$), and the cost function of an edge of the graph described in Equation (3.6);*
- *The probability density function $P_x(X)$ of the location of each node u ($u \in V$) where X is the possible physical location;*
- *The deterministic location x of a node m ($m \in V$, $m \neq s$) and the deterministic location y of the sink s;*
- *The cost $\omega(\overrightarrow{\ell})$ of a path $\overrightarrow{\ell}$ from node m to the sink s;*

Compute the probability ρ that there exists a path $\overrightarrow{\ell'}$ from the node m to the sink s other than $\overrightarrow{\ell}$ such that the cost of $\overrightarrow{\ell'}$, $\omega(\overrightarrow{\ell'})$, satisfies $\omega(\overrightarrow{\ell'}) < \omega(\overrightarrow{\ell})$.■

With the solution of Problem 2, an improvement of BOU is to update line 16 of Algorithm 1 to Algorithm 2. We call this improved approach BOU-W (*wait before broadcast, on update*).

Here α is a constant whose value can be determined empirically.

Now we discuss the solution to Problem 2. Obviously, this problem is equivalent to computing the probability that the known path $\overrightarrow{\ell}$ is not the shortest path from the node m to the sink s. Therefore, in order to solve the problem, we should know the probability distribution of the node cost of a *given* node. However, it is very difficult to derive this probability distribution mathematically. But, we can perform Monte Carlo method to find this probability distribution. We discuss this approach in the following example.

Consider a network that contains 30 sensor nodes and a sink. The sensor nodes are deployed uniformly in a $30m \times 30m$ square and the sink is at the center of the square.

We fix the physical distance between one sensor node (denoted by node m) and the sink. This distance is denoted by d. Then we randomly generate the locations of the other 29 sensor nodes in a uniform way for h times and thus we get h graphs. For each graph, we perform the Dijkstra algorithm to find the shortest path from node m to the sink given the edge cost described in Equation (3.6) and record the node cost $\eta(m)$. Thus we get h results of $\eta(m)$. Let each number of series $N_i(i = 1, 2, ...)$ be the number of results which is in interval $(0, i \cdot \tau]$, where τ is a constant. Obviously, if h is large enough and τ is small enough, N_i/h reflects the probability distribution of $\eta(m)$.

We gradually change the distance d with a step size equal to δ and perform the above process. In this way, we can get the probability distributions of $\eta(m)$ with different distances between node m and the sink. Figure 3.2 shows part of the results of the statistical data, in which δ is 3, h is 10^5, and τ is

5. The probabilities is calculated with N_i/h.

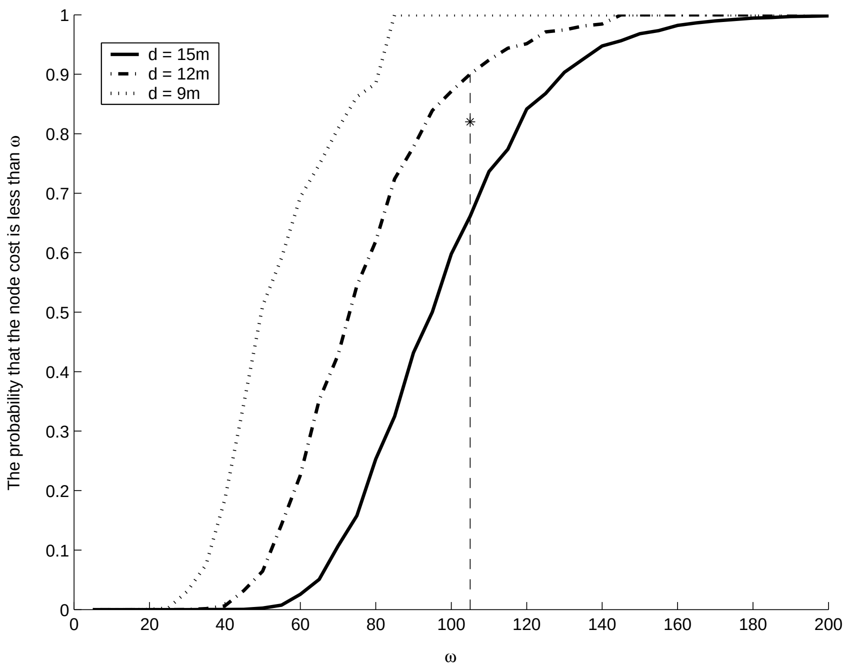

Figure 3.2: The estimated probability distribution of node cost

With the statistical data achieved in the above approach, Problem 2 can be solved approximately. For example, when $\omega(\overrightarrow{\ell}) = 105$ and the distance $\|x - y\|$ is 13, the probability ρ is 0.82, which is approximately estimated with simple linear interpolation technique according to the data shown in Figure 3.2.

Note that the above approach to achieve the solution of Problem 2 requires only localized information. The cost $\omega(\overrightarrow{\ell})$ can be calculated as the sum of the node cost of the neighbor from which the node receives a configuration packet and the cost of edge between the node and the neighbor according to Equation (3.6). The location of the sink can be found in the configuration packet and the statistical data can be achieved with emulations before the sensor nodes are deployed and saved in the memory of each sensor node. The complexity to calculate the waiting time is negligible if we employ the statistical estimation approach discussed above to solve Problem 2.

3.2.4 BOU-WA: an approximation approach

In subsection 3.2.3, we discuss how to determine the waiting time before a node broadcasts a configuration packet on the update of its node cost. However, the mathematical solution of Problem 2 is not easy to be derived. Although Monte Carlo method helps to find approximation solutions, when the node deployment scheme cannot be well modeled, the statistical data cannot be achieved with emulations. Moreover, to obtain high accuracy, the above numerical solution of the Problem 2 requires that τ and δ are small. This means that huge volume of statistical data should be saved in a sensor node, which might not be practical due to the hardware constraint of the sensor node implementation [29]. In this subsection, we provide an approximation solution to determine the waiting time.

Let's still take Figure 3.1 as an example. For simplicity, we consider the space is 2-dimensional and we adopt the ideal free space transmission model [51] (*i.e.*, n=2 in Equation (3.6)). We ignore the energy consumed to receive and process a packet. Note that in actual application case, we can employ a more sophisticated model and without loss of generality, the approach proposed in our following discussions is still applicable.

We denote $\{C, A, ..., S\}$ as the path from node C via node A and some other nodes to the sink S. If there exists a node B in the area ϕ, surely the path $\{C, B, A, ..., S\}$ is shorter than the path $\{C, A, ..., S\}$ because

$$\begin{aligned}(\|X(A) - X(C)\|)^2 > (\|X(A) - X(B)\|)^2 \\ +(\|X(B) - X(C)\|)^2, \qquad (3.7)\end{aligned}$$

as node B is within the circular area with diameter AC

Note that there may exist another node D outside the area ϕ such that $\{C, D, ..., S\}$ is shorter than the path $\{C, B, A, ..., S\}$. But we can simply consider the probability that node B exists as the probability that a better path than the path $\{C, A, ..., S\}$ exists as an approximation, although this probability is smaller than the actual probability that a better path than the path $\{C, A, ..., S\}$ exists.

If we determine the waiting time according to this approximation probability (*i.e.*, the probability that node B exists), then we are waiting a shorter period of time than the BOU-W scheme. Therefore, the risk that the node will update its routing information and broadcast again is larger. In our simulation study, we will show that this risk is manageable and the approximation works well.

We name the scheme that adopts the waiting time based on this approximation the BOU-WA (*BOU-W with approximation*) scheme. BOU-WA is an improvement of BOU by updating line 16 of the BOU mechanism described in Algorithm 1 to Algorithm 3.

Suppose the deployed node number is k and the deployment area is φ. Assume the nodes are deployed uniformly in area φ. The probability ρ' that there exists at least one node in area ϕ is as follow.

$$\rho' = 1 - (1 - \frac{\phi}{\varphi})^{(k-2)}, \tag{3.8}$$

The complexity to calculate the waiting time in this scheme is negligible as a node only has to solve ρ' in Equation (3.8). Also, this scheme requires no message exchange among sensor nodes. As k and φ are known before node deployment, they can be programmed into the node beforehand. ϕ can be calculated according to $X(A)$, $X(C)$ and the cost function described in Equation (3.6). In this example that adopts the ideal free space transmission model, it is as follows.

$$\phi = \pi \cdot (\frac{\|X(A) - X(C)\|}{2})^2 \tag{3.9}$$

Our simulation work in Section 3.3 will compare this approximation probability, *i.e.*, ρ', and the probability that a better path exists, *i.e.*, ρ, which is obtained with the Monte Carlo method. How BOU-WA performs with different network scales will also be investigated.

3.3 Simulation Results

We program the BOU scheme and the BOU-WA scheme with NS-2 [23] and study the performance of these schemes with simulations.

Table 3.1: The setting of the simulation network

Area of sensor field	100m*100m
MAC Protocol	IEEE 802.11 without CTS/RTS and ACK
Transmitter Power	0.660W
Receiver Power	0.395W
Wireless Communication Model	Free Space
Packet length	36 bytes

In our simulation work, we first investigate the improvement of BOU-WA with different values of α (α is used to calculate the waiting time described in Algorithm 3 in the BOU-WA scheme) comparing with the BOU scheme in terms of energy consumption overhead to set up the transmitter power level of each in-network sensor node. The converging times of these schemes are also compared. Different network scales (*i.e.*, different node numbers of a network) are adopted in the simulations to show the scalability of these schemes.

To study how the BOU-WA scheme could approach the BOU-W scheme, we also investigate the differences between the probability ρ' calculated with Equation (3.8) and the actual probability ρ estimated with the Monte Carlo method.

Detailed settings of the simulation network are shown in Table 4.1.

3.3.1 The comparisons of BOU and BOU-WA

In the network described in Table 4.1, we randomly deploy $k+1$ nodes in a uniform manner. We randomly select a node as the sink node and the other k nodes as the in-network sensor nodes. Let the sink node initiate the algorithms (*i.e.*, it broadcasts the first configuration packet with a node cost equal to zero). We set k as 10, 20, 30, 40, 50, 60, 80, 100 and 150. For each setting of k, we set the constant α as 0.05, 0.1, 0.2, 0.4, 0.8 (the waiting time before broadcasting is $\alpha \cdot \rho'$ seconds in the BOU-WA scheme). For each setting of k, we run the simulations of the BOU scheme for 1000 times and for each setting of k and α, we also run the simulations of the BOU-WA scheme for 1000 times. We average the results of all the simulation runs in each setting.

The total number of broadcasts in setting up the transmitter power levels and the energy consumption overhead of BOU and BOU-WA are shown in Figure 3.3 and Figure 3.4. It can be found that BOU-WA greatly improves the BOU scheme, especially when the number of nodes is large. Moreover, the greater α is, the better the BOU-WA scheme performs. But when α is large (*i.e.*, $\alpha = 0.2$, 0.4 or 0.8), different values of α do not have much different effects on the energy consumption overhead of BOU-WA.

The counter-effect of BOU-WA comparing to BOU is that

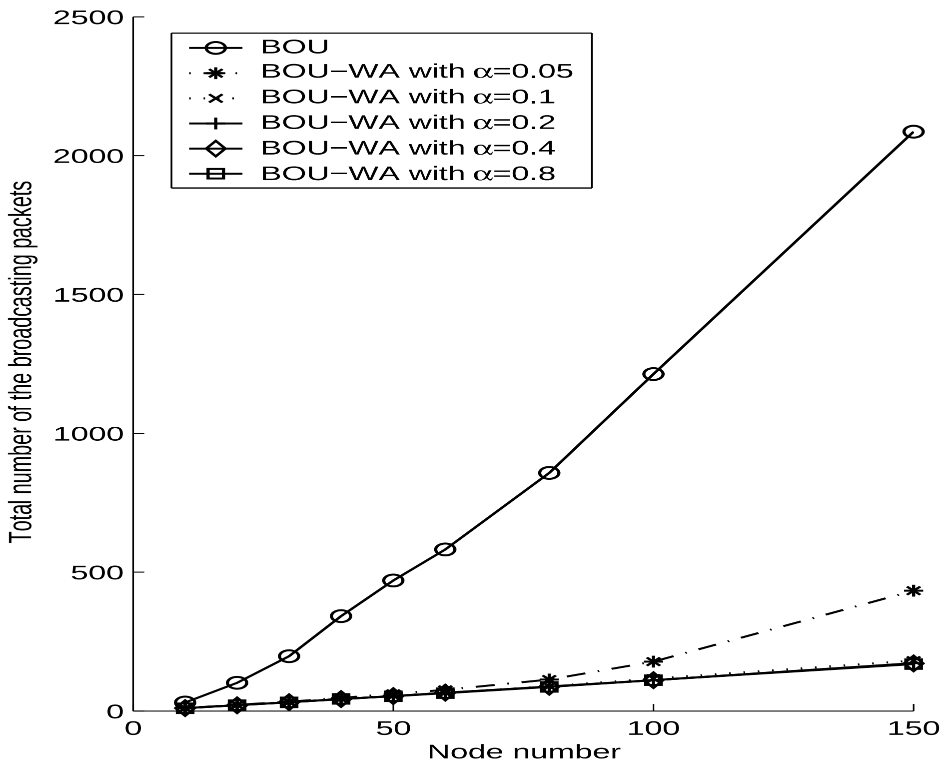

Figure 3.3: Total number of broadcasts

BOU-WA might require larger converging time. The converging times of the BOU and BOU-WA are shown in Figure 3.5. These simulation results show that the greater α is, the longer the converging time of BOU-WA scheme is. Note that when α is less than 0.2, BOU-WA has a smaller converging time than BOU. This is because the number of broadcasts in BOU-WA is much smaller than that that in BOU. As a result, the load of the wireless channel is lighter in case that BOU-WA is employed. Therefore, if a node wants to send a packet, it waits for less time until the channel is free in case that BOU-WA is employed.

These results show that BOU-WA, with a good parameter α, can perform much better than BOU.

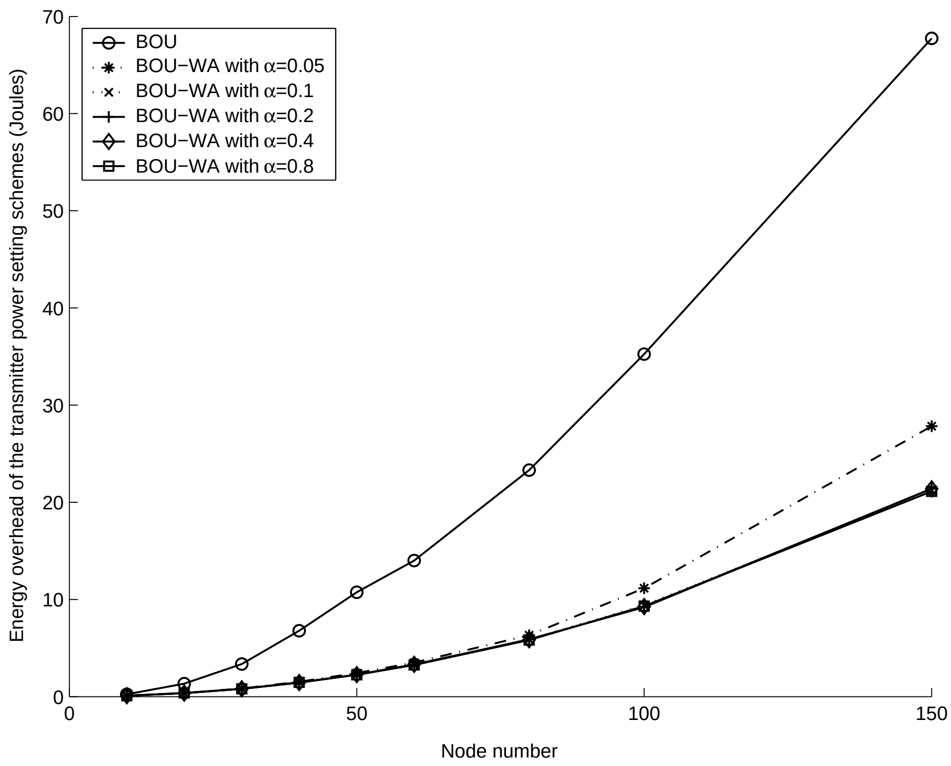

Figure 3.4: The energy consumption overhead

3.3.2 The approximation of BOU-WA

To study the estimation error of ρ' in BOU-WA, we employ the Monte Carlo method to calculate ρ.

In the network described in Table 4.1, we randomly deploy k nodes in a uniform manner and we place the sink at the corner of the square. We then randomly select two nodes in the network. One is a node that sends out a configuration packet, denoted by node s; and the other is a node that receives the configuration packet, denoted by r. With the BOU-WA scheme, ρ' is calculated. We change the locations of the other nodes, except the sink, randomly for 10000 times and count the number of instances t in which node s is not the adjacent neighbor of node r along the shortest path from node r to the sink when the algorithm converges. $t/10000$ is regarded as the probability

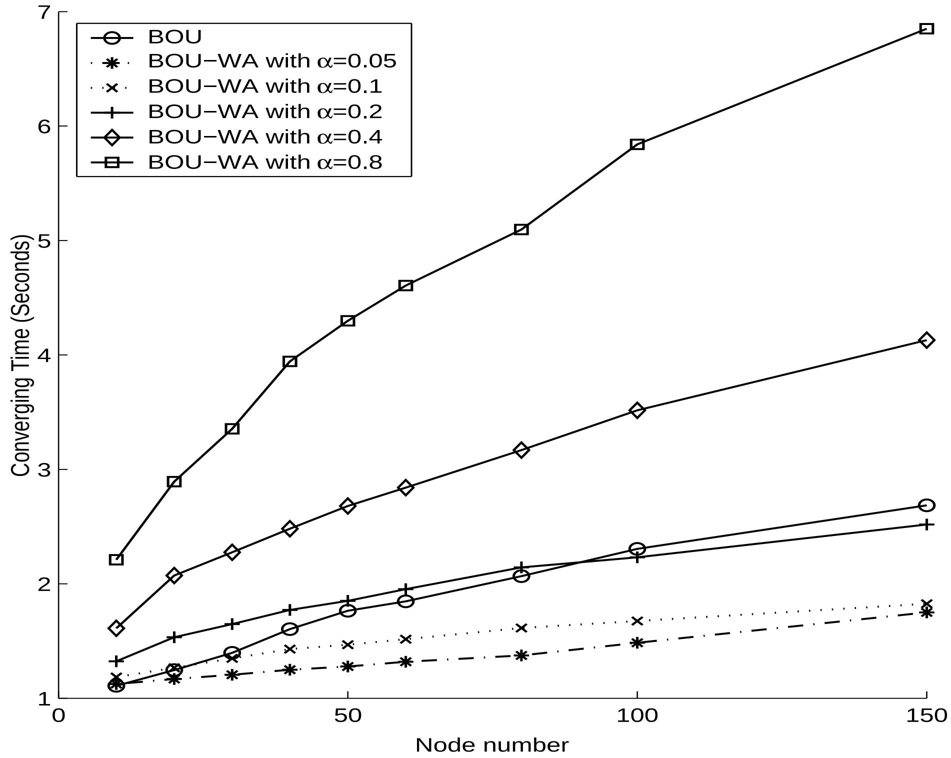

Figure 3.5: The converging time

ρ that a better path exists than the path $\{r, s, ..., sink\}$.

We perform the above process for 10000 times and the differences between ρ' and ρ are averaged. The average error can be regarded as the probability estimation error of the BOU-WA scheme.

We set k as 10, 20, 30, 40, 50, 80 and 100. We achieve the estimation error with the above method. The results are shown in Figure 3.6.

It can be seen that the estimation error is small. Moreover, the higher the node number is, the better accuracy the estimation achieves. It is worth to mention that finding a more accurate estimation is not necessary, because the average packet number that an in-network node should broadcast in the BOU-WA scheme is already close to the lower bound 1. Figure 3.7

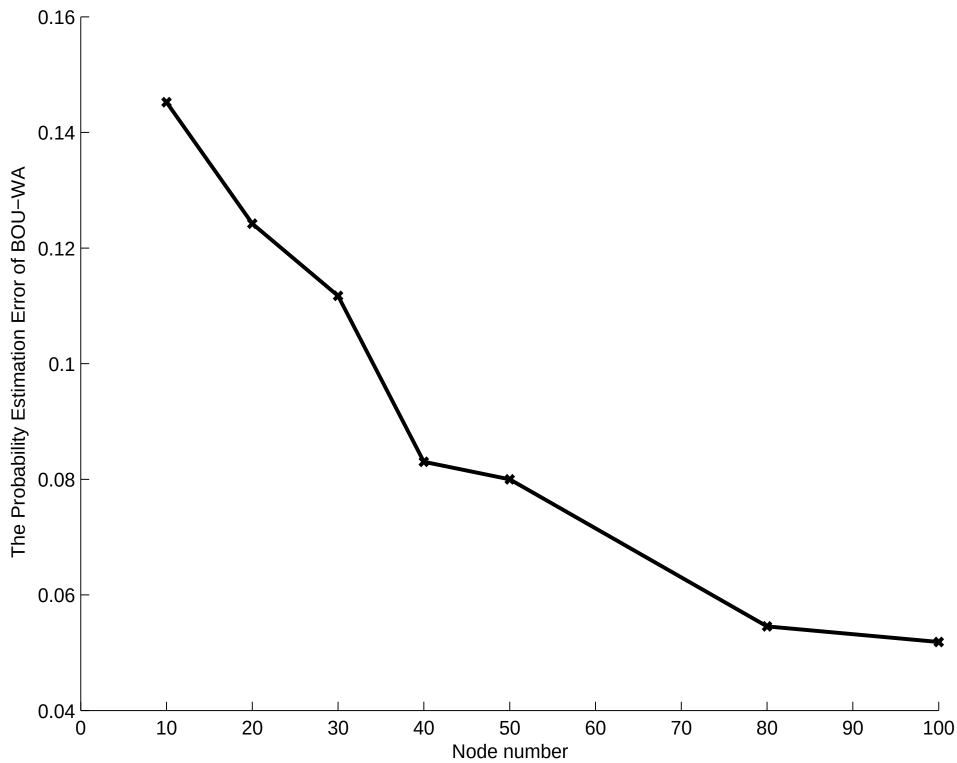

Figure 3.6: The average estimation error of ρ'

shows the average broadcast number of a node when we set $\alpha = 0.8$ in the BOU-WA scheme. The lower bound is 1 because each node obviously has to broadcast at lease once. This means that the room to further improve BOU-WA is already very small.

3.4 Related Work

Research on many aspects of energy-efficient sensor-to-sink data communication has been conducted in the literature. In the work on data routing, directed diffusion [30] introduces the data-centric notion. It proposes that sensor-to-sink packets could be pre-processed at in-network nodes with data fusion and data aggregation techniques in order to reduce the total number of

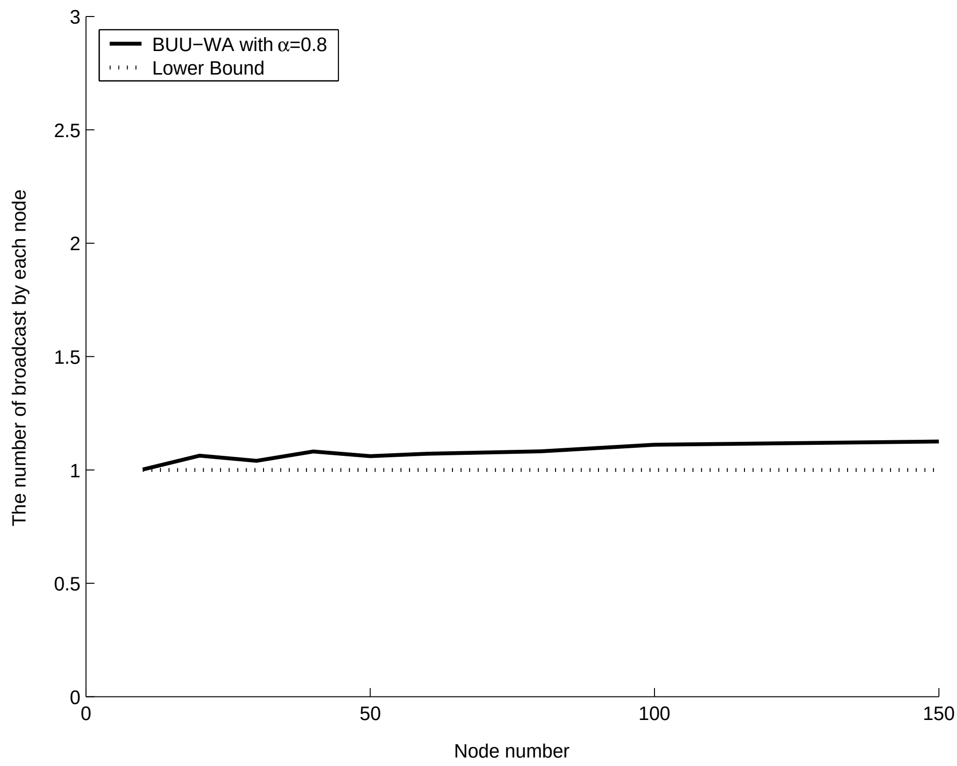

Figure 3.7: The average number of broadcasts by each node

packets needed to convey the event information. Many other data communication schemes for WSNs have been proposed [8, 17, 24, 58, 60], all of which are striving to achieve energy efficiency, while maintaining other properties of the communication such as reliability or information fidelity. [1] is a good survey of these research issues.

In addition, traffic congestion will cause high packet loss rates, which result in low energy efficiency. To this end, mechanisms for detecting and even avoiding congestion have been studied [54, 69]. In our recent work [79], we propose that the source reporting rates should be determined by the communication cost (which could be implemented to reflect the wireless communication conditions) and the importance of each source node (the metric that reflects how much information the source could provide on the event of interest) so that the communication scheme can effectively avoid congestion and provide reliable data transport.

The problem of broadcast storm has been extensively studied in multi-hop wireless networks. A series of solutions (*e.g.*, [66]) have been proposed to mitigate the storm. However, their focus is on how to efficiently send a packet to every node in a network without much duplication. In our work, we reduce the number of broadcasts which is performed by each node to report its node cost to its one-hop neighbors for setting up energy-efficient paths.

3.5 Conclusion Remarks and Future Work

This chapter has examined the problem of transmitter power control for energy-efficient sensor-to-sink communications. We have modeled this problem based on the network and application features of WSNs. An intuitive implementation to solve this problem, namely BOU, has been presented. We have identified the broadcast explosion problem in BOU, and then improved BOU by allowing a waiting period before each broadcasting. We have shown that the waiting time should be proportional to the probability that a node would find a more energy-efficient path to the sink, and presented an efficient approximation algorithm to calculate the probability. Simulations have been designed to evaluate BOU and BOU-WA. The results have validated the effectiveness of BOU-WA; specifically, it can set up energy-efficient paths for sensor-to-sink traffic with low overhead in a reasonable converging time.

There are many possible future directions for this work. We are particular interested in integrating our algorithm with existing data fusion/aggregation schemes. We also interested in practical implementations, and we expect more issues can be identified in this process.

□ **End of chapter.**

Chapter 4

Solving the Sensor-Grouping Problem

Summary

We propose ι, a novel index for evaluation of point-distribution. ι is the minimum distance between each pair of points normalized by the average distance between each pair of points. We find that a set of points that achieve a maximum value of ι result in a honeycomb structure. We propose that ι can serve as a good index to evaluate the distribution of the points, which can be employed in coverage-related problems in wireless sensor networks (WSNs). To validate this idea, we formulate a general sensor-grouping problem for WSNs and provide a general sensing model. We show that locally maximizing ι at sensor nodes is a good approach to solve this problem with an algorithm called Maximizing-ι Node-Deduction (MIND). Simulation results verify that MIND outperforms a greedy algorithm that exploits sensor-redundancy we design. This demonstrates a good application of employing ι in coverage-related problems for WSNs. This chapter is based on the work presented in [80].

4.1 Introduction

In many application scenarios, WSNs are employed to conduct surveillance tasks in adverse, or even worse, in hostile working environments. One major problem caused is that sensor nodes are subjected to failures. Therefore, fault tolerance of a WSN is critical.

One way to achieve fault tolerance is that a WSN should contain a large number of redundant nodes in order to tolerate node failures. It is vital to provide a mechanism that redundant nodes can be working in sleeping mode (*i.e.*, major power-consuming units such as the transceiver of a redundant sensor node can be shut off) to save energy, and thus to prolong the network lifetime. Redundancy should be exploited as much as possible for the set of sensors that are currently taking charge in the surveillance work of the network area [13].

We find that the minimum distance between each pair of points normalized by the average distance between each pair of points serves as a good index to evaluate the distribution of the points. We call this index, denoted by ι, the *normalized minimum distance*. If points are moveable, we find that maximizing ι results in a honeycomb structure. The honeycomb structure poses that the coverage efficiency is the best if each point represents a sensor node that is providing surveillance work. Employing ι in coverage-related problems is thus deemed promising.

This enlightens us that maximizing ι is a good approach to select a set of sensors that are currently taking charge in the surveillance work of the network area. To explore the effectiveness of employing ι in coverage-related problems, we formulate a sensor-grouping problem for high-redundancy WSNs. An algorithm called *Maximizing-ι Node-Deduction* (MIND) is proposed in which redundant sensor nodes are removed to obtain a large ι for each set of sensors that are currently taking charge in the surveillance work of the network area. We also introduce another greedy solution called *Incremental Coverage Quality Algorithm* (ICQA) for this problem, which serves as a benchmark to evaluate MIND.

The main contribution of this work is twofold. First, we introduce a novel index ι for evaluation of point-distribution. We show that maximizing ι of a WSN results in low redundancy of the network. Second, we formulate a general sensor-grouping problem for WSNs and provide a general sensing model. With the MIND algorithm we show that locally maximizing ι among each sensor node and its neighbors is a good approach to solve this problem. This demonstrates a good application of employing ι in coverage-related problems.

The rest of this chapter is organized as follows. In Section 4.2, we introduce our point-distribution index ι. We survey related work and formulate a sensor-grouping problem together with a general sensing model in Section 4.3. Section 4.4 investigates the application of ι in this grouping problem. We propose MIND for this problem and introduce ICQA as a benchmark. In Section 4.5, we present our simulation results in which MIND and ICQA are compared. Section 4.6 provides conclusion remarks.

4.2 The Normalized Minimum Distance ι: A Point-Distribution Index

Suppose there are n points in a Euclidean space Ω. The coordinates of these points are denoted by x_i ($i = 1, ..., n$).

It may be necessary to evaluate how the distribution of these points is. There are many metrics to achieve this goal. For example, the *Mean Square Error* from these points to their mean value can be employed to calculate how these points deviate from their mean (*i.e.*, their central). In resource-sharing evaluation, the *Global Fairness Index* (GFI) is often employed to measure how even the resource distributes among these points [31], when x_i represents the amount of resource that belong to point i. In WSNs, GFI is usually used to calculate how even the remaining energy of sensor nodes is.

When n is larger than 2 and the points do not all overlap (That points all overlap means $x_i = x_j$, $\forall\ i,\ j = 1, 2, ..., n$). We propose a novel index called *the normalized minimum distance*, namely ι, to evaluate the distribution of the points. ι is the minimum distance between each pair of points normalized by the average distance between each pair of points. It is calculated by:

$$\iota = \frac{\min(||x_i - x_j||)}{\mu} (\forall\, i, j = 1, 2, ..., n; and\, i \neq j) \qquad (4.1)$$

where $||x_i - x_j||$ denotes the Euclidean distance between point i and point j in Ω, the $\min(\cdot)$ function calculates the minimum distance between each pair of points, and μ is the average distance between each pair of points, which is:

$$\mu = \frac{(\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} ||x_i - x_j||)}{n(n-1)} \qquad (4.2)$$

ι measures how well the points separate from one another. Obviously, ι is in interval $[0, 1]$. ι is equal to 1 if and only if n is equal to 3 and these three points forms an equilateral triangle. ι is equal to zero if any two points overlap. ι is a very interesting value of a set of points. If we consider each x_i ($\forall i = 1, ..., n$) is a variable in Ω, how these n points would look like if ι is maximized?

An algorithm is implemented to generate the topology in which ι is locally maximized (The algorithm is presented in Algorithm 4 in Appendix B). We consider a 2-dimensional space.

We select $n = 10, 20, 30, ..., 100$ and perform this algorithm. In order to avoid that the algorithm converge to local optimum, we select different random seeds to generate the initial points for 1000 time and obtain the best one that results in the largest ι when the algorithm converges. Figure 4.1 demonstrates what the resulting topology looks like when $n = 20$ as an example.

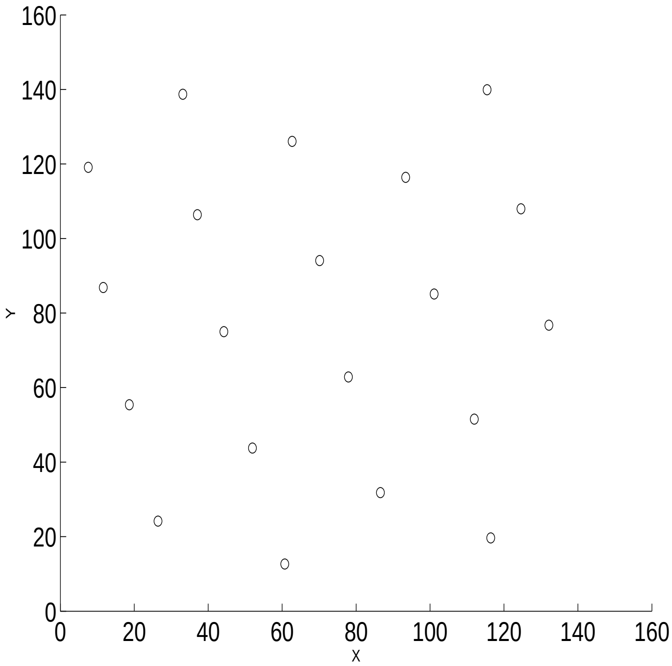

Figure 4.1: Node Number = 20, $\iota = 0.435376$

Suppose each point represents a sensor node. If the sensor coverage model is the Boolean coverage model [65, 73, 75, 76] and the coverage radius of each node is the same. It is exciting to see that this topology results in lowest redundancy because the Vonoroi diagram [6] formed by these nodes (A Vonoroi diagram formed by a set of nodes partitions a space into a set of convex polygons such that points inside a polygon are closest to only one particular node) is a honeycomb-like structure [34][1].

[1]This is how base stations of a wireless cellular network are deployed and why such a network is called a cellular one.

This enlightens us that ι may be employed to solve problems related to sensor-coverage of an area. In WSNs, it is desirable that the active sensor nodes that are performing surveillance task should separate from one another. Under the constraint that the sensing area should be covered, the more each node separates from the others, the less the redundancy of the coverage is. ι indicates the quality of such separation. It should be useful for approaches on sensor-coverage related problems.

In our following discussions, we will show the effectiveness of employing ι in sensor-grouping problem.

4.3 The Sensor-Grouping Problem

In many application scenarios, to achieve fault tolerance, a WSN contains a large number of redundant nodes in order to tolerate node failures. A node sleeping-working schedule scheme is therefore highly desired to exploit the redundancy of working sensors and let as many nodes as possible sleep.

Much work in the literature is on this issue [13]. Yan *et al* introduced a differentiated service in which a sensor node finds out its responsible working duration with cooperation of its neighbors to ensure the coverage of sampled points [75]. Ye *et al* developed PEAS in which sensor nodes wake up randomly over time, probe their neighboring nodes, and decide whether they should begin to take charge of surveillance work [76]. Xing *et al* exploited a probabilistic distributed detection model with a protocol called Coordinating Grid (Co-Grid) [72]. Wang *et al* designed an approach called Coverage Configuration Protocol (CCP) which introduced the notion that the coverage degree of intersection-points of the neighboring nodes' sensing-perimeters indicates the coverage of a convex region [70, 73]. In the recent work of our group[16], Chen *et al* also provided a sleeping configuration protocol, namely SSCP, in which sleeping eligibility of a sensor node is determined by a local Voronoi diagram. SSCP can provide different levels of redundancy to maintain different requirements of fault tolerance.

The major feature of the aforementioned protocols is that they employ online distributed and localized algorithms in which a sensor node determines its sleeping eligibility and/or sleeping time based on the coverage requirement of its sensing area with some information provided by its neighbors.

Another major approach for sensor node sleeping-working scheduling issue is to group sensor nodes. Sensor nodes in a network are divided into several disjoint sets. Each set of sensor nodes are able to maintain the required area surveillance work. The sensor nodes are scheduled according to which set they belong to. These sets work successively. Only one set of sensor nodes work at any time. We call the issue *sensor-grouping problem.*

The major advantage of this approach is that it avoids the overhead caused by the processes of coordination of sensor nodes to make decision on whether a sensor node is a candidate to sleep or work and how long it should sleep or work. Such processes should be performed from time to time during the lifetime of a network in many online distributed and localized algorithms. The large overhead caused by such processes is the main drawback of the online distributed and localized algorithms. On the contrary, roughly speaking, this approach groups sensor nodes in one time and schedules when each set of sensor nodes should be on duty. It does not require frequent decision-making on working/sleeping eligibility [2].

In [61] by Slijepcevic *et al*, the sensing area is divided into regions. Sensor nodes are grouped with the most-constrained least-constraining algorithm. It is a greedy algorithm in which

[2]Note that if some nodes die, a re-grouping process might also be performed to exploit the remaining nodes in a set of sensor nodes. How to provide this mechanism is beyond the scope of this work and yet to be explored.

the priority of selecting a given sensor is determined by how many uncovered regions this sensor covers and the redundancy caused by this sensor. In [11] by Cardei *et al*, disjoint sensor sets are modeled as disjoint dominating sets. Although maximum dominating sets computation is NP-complete, the authors proposed a graph-coloring based algorithm. Cardei *et al* also studied similar problem in the domain of covering target points in [10]. The NP-completeness of the problem is proved and a heuristic that computes the sets are proposed. These algorithms are centralized solutions of sensor-grouping problem.

However, global information (*e.g.*, the location of each in-network sensor node) of a large scale WSN is also very expensive to obtained online. Also it is usually infeasible to obtain such information before sensor nodes are deployed. For example, sensor nodes are usually deployed in a random manner and the location of each in-network sensor node is determined only after a node is deployed.

The solution of sensor-grouping problem should only base on locally obtainable information of a sensor node. That is to say, nodes should determine which group they should join in a fully distributed way. Here locally obtainable information refers to a node's local information and the information that can be directly obtained from its *adjacent nodes*, *i.e.*, nodes within its communication range.

In Subsection 4.3.1, we provide a general problem formulation of the sensor-grouping problem. Distributed-solution requirement is formulated in this problem. It is followed by discussion in Subsection 4.3.2 on a general sensing model, which serves as a given condition of the sensor-grouping problem formulation.

To facilitate our discussions, the notations in our following discussions are described as follows.

- n: The number in-network sensor nodes.
- $\mathcal{S}(j)$ $(j = 1, 2, ..., m)$: The jth set of sensor nodes where m

is the number of sets.

- $\mathcal{L}(i)$ $(i = 1, 2, ..., n)$: The physical location of node i.
- ϕ: The area monitored by the network: *i.e.,* the sensing area of the network.
- R: The sensing radius of a sensor node. We assume that a sensor node can only be responsible to monitor a circular area centered at the node with a radius equal to R. This is a usual assumption in work that addresses sensor-coverage related problems. We call this circular area the *sensing area* of a node.

4.3.1 Problem Formulation

We assume that each sensor node can know its approximate physical location. The approximate location information is obtainable if each sensor node carries a GPS receiver or if some localization algorithms are employed (*e.g.*, [9]).

Problem 3 *Given:*

- *The set of each sensor node i's sensing neighbors $\mathcal{N}(i)$ and the location of each member in $\mathcal{N}(i)$;*
- *A* sensing model *which quantitatively describes how a point P in area ϕ is covered by sensor nodes that are responsible to monitor this point. We call this quantity the* coverage quality *of P.*
- *The coverage quality requirement in ϕ, denoted by s. When the coverage of a point is larger than this threshold, we say this point is* covered.

For each sensor node i, make a decision on which group $\mathcal{S}(j)$ it should join so that:

- *Area ϕ can be covered by sensor nodes in each set $\mathcal{S}(j)$*

- *m, the number of sets $\mathcal{S}(j)$ is maximized.* ■

In this formulation, we call sensor nodes within a circular area centered at a sensor node i with a radius equal to $2 \cdot R$ the *sensing neighbors* of node i. This is because sensors nodes in this area, together with node i, may be cooperative to ensure the coverage of a point inside node i's sensing area.

We assume that the communication range of a sensor node is larger than $2 \cdot R$, which is also a general assumption in work that addresses sensor-coverage related problems. That is to say, the first given condition in Problem 3 is the information that can be obtained directly from a node's adjacent nodes. It is therefore locally obtainable information. The last two given conditions in this problem formulation can be programmed into a node before it is deployed or by a node-programming protocol (*e.g.*, [36, 46]) during network runtime. Therefore, the given conditions can all be easily obtained by a sensor-grouping scheme with fully distributed implementation.

We reify this problem with a realistic sensing model in next subsection.

4.3.2 A General Sensing Model

As WSNs are usually employed to monitor possible events in a given area, it is therefore a design requirement that an event occurring in the network area must/may be successfully detected by sensors.

This issue is usually formulated as how to ensure that an event signal omitted in an arbitrary point in the network area can be detected by sensor nodes. Obviously, a *sensing model* is required to address this problem so that how a point in the network area is covered can be modeled and quantified. Thus the coverage quality of a WSN can be evaluated.

Different applications of WSNs employ different types of sensors, which surely have widely different theoretical and physical characteristics. Therefore, to fulfill different application requirements, different sensing models should be constructed based on the characteristics of the sensors employed.

A simple theoretical sensing model is the *Boolean sensing model* [65, 73, 75, 76]. Boolean sensing model assumes that a sensor node can always detect an event occurring in its responsible sensing area. But most sensors detect events according to the signal strength sensed. Event signals usually fade in relation to the physical distance between an event and the sensor. The larger the distance, the weaker the event signals that can be sensed by the sensor, which results in a reduction of the probability that the event can be successfully detected by the sensor.

As in WSNs, event signals are usually electromagnetic, acoustic, or photic signals, they fade exponentially with the increasing of their transmit distance. Specifically, the signal strength $\mathcal{E}(d)$ of an event that is received by a sensor node satisfies:

$$\mathcal{E}(d) = \frac{\alpha}{d^{\beta}} \tag{4.3}$$

where d is the physical distance from the event to the sensor node; α is related to the signal strength omitted by the event; and β is signal fading factor which is typically a positive number larger than or equal to 2. Usually, α and β are considered as constants.

Based on this notion, to be more reasonable, researchers propose *collaborative sensing model* to capture application requirements: Area coverage can be maintained by a set of *collaborative sensor nodes*: For a point with physical location L, the point is considered covered by the collaboration of i sensors (denoted by k_1, ..., k_i) if and only if the following two equations holds [16, 39, 43].

$$\forall j = 1, ..., i;\ \|\mathcal{L}(k_j) - L\| < R. \tag{4.4}$$

$$\mathcal{C}(L) = \sum_{j=1}^{i} (\mathcal{E}(\|\mathcal{L}(k_j) - L\|) > s. \tag{4.5}$$

$\mathcal{C}(L)$ is regarded as the coverage quality of location L in the network area [16, 39, 43].

However, we notice that defining the sensibility as the sum of the sensed signal strength by each collaborative sensor implies a very special application: Applications must employ the sum of the signal strength to achieve decision-making. To capture generally realistic application requirement, we modify the definition described in Equation (4.5). The model we adopt in this work is described in details as follows.

We consider the probability $\mathcal{P}(L, k_j)$ that an event located at L can be detected by sensor k_j is related to the signal strength sensed by k_j. Formally,

$$\mathcal{P}(L, k_j) = \gamma\mathcal{E}(d) = \frac{\delta}{(\|\mathcal{L}(k_j) - L\|/\epsilon + 1)^{\beta}}, \tag{4.6}$$

where γ is a constant and $\delta = \gamma\alpha$ is a constant too. ϵ normalizes the distance to a proper scale and the "+1" item is to avoid infinite value of $\mathcal{P}(L, k_j)$.

The probability that an event located at L can be detected by any collaborative sensors that satisfied Equation (4.4) is:

$$\mathcal{P}'(L) = 1 - \prod_{j=1}^{i} (1 - \mathcal{P}(L, k_j)). \tag{4.7}$$

As the detection probability $\mathcal{P}'(L)$ reasonably determines how an event occurring at location L can be detected by the networks, it is a good measure of the coverage quality of location L in a WSN. Specifically, Equation (4.5) is modified to:

$$\mathcal{C}(L) \quad = \mathcal{P}'(L)$$

$$= 1 - \prod_{j=1}^{i}[1 - \frac{\delta}{(\|\mathcal{L}(k_j) - L\|/\epsilon + 1)^{\beta}}] > s. \quad (4.8)$$

To sum it up, we consider a point at location L is covered if Equation (4.4) and (4.8) hold.

4.4 Maximizing-ι Node-Deduction Algorithm for Sensor-Grouping Problem

Before we process to introduce algorithms to solve the sensor grouping problem, let us define the *margin* (denoted by θ) of an area ϕ monitored by the network as the band-like marginal area of ϕ and all the points on the outer perimeter of θ is ρ distance away from all the points on the inner perimeter of θ. ρ is called the *margin length*.

In a practical network, sensor nodes are usually evenly deployed in the network area. Obviously, the number of sensor nodes that can sense an event occurring in the margin of the network is smaller than the number of sensor nodes that can sense an event occurring in other area of the network. Based on this consideration, in our algorithm design, we ensure the coverage quality of the network area *except* the margin. The information on ϕ and ρ is network-based. Each in-network sensor node can be pre-programmed or on-line informed about ϕ and ρ, and thus calculate whether a point in its sensing area is in the margin or not.

4.4.1 Maximizing-ι Node-Deduction Algorithm

The node-deduction process of our Maximizing-ι Node-Deduction Algorithm (MIND) is simple. A node i greedily maximizes ι of the sub-network composed by itself, its ungrouped sensing neighbors, and the neighbors that are in the same group of itself. Under the constraint that the coverage quality of its sensing area should be ensured, node i deletes nodes in this sub-network one by one. The candidate to be pruned satisfies that:

- It is an ungrouped node.
- The deletion of the node will not result in uncovered-points inside the sensing area of i.

A candidate is deleted if the deletion of the candidate results in largest ι of the sub-network compared to the deletion of other candidates. This node-deduction process continues until no candidate can be found. Then all the ungrouped sensing neighbors that are not deleted are grouped into the same group of node i. We call the sensing neighbors that are in the same group of node i the *group sensing neighbors* of node i. We then call node i a *finished* node, meaning that it has finished the above procedure and the sensing area of the node is covered. Those nodes that have not yet finished this procedure are called *unfinished* nodes.

The above procedure initiates at a random-selected node that is not in the margin. The node is grouped to the first group. It calculates the resulting group sensing neighbors of it based on the above procedure. It informs these group sensing neighbors that they are selected in the group. Then it hands over the above procedure to an unfinished group sensing neighbors that is the farthest from itself. This group sensing neighbor continues this procedure until no unfinished neighbor can be found. Then the first group is formed.

After a group is formed, another random-selected ungrouped node begins to group itself to the second group and initiates the above procedure. In this way, groups are formed one by one. When a node that involves in this algorithm found out that the coverage quality if its sensing area, except what overlaps the network margin, cannot be ensured even if all its ungrouped sensing neighbors are grouped into the same group as itself, the algorithm stops. MIND is based on locally obtainable information of sensor nodes. It is a distributed algorithm that serves as an approximation solution of Problem 3.

4.4.2 Incremental Coverage Quality Algorithm: A Benchmark for MIND

To evaluate the effectiveness of introducing ι in the sensor-group problem, another algorithm for sensor-group problem called Incremental Coverage Quality Algorithm (ICQA) is designed. Our aim is to evaluate how an idea, *i.e.*, MIND, based on locally maximize ι performs.

In ICQA, a node-selecting process is as follows. A node i greedily selects an ungrouped sensing neighbor in the same group as itself one by one, and informs the neighbor it is selected in the group. The criterion is:

- The selected neighbor is responsible to provide surveillance work for some *uncovered* parts of node i's sensing area. (*i.e.*, the coverage quality requirement of the parts is not fulfilled if this neighbor is not selected.)
- The selected neighbor results in highest improvement of the coverage quality of the neighbor's sensing area.

The improvement of the coverage quality, mathematically, should be the integral of the the improvements of all points inside the neighbor's sensing area. A numerical approximation is employed to calculate this improvement. Details are presented in our simulation study.

This node-selecting process continues until the sensing area of node i is entirely covered. In this way, node i's *group sensing neighbors* are found. The above procedure is handed over as what MIND does and new groups are thus formed one by one. And the condition that ICQA stops is the same as MIND.

ICQA is also based on locally obtainable information of sensor nodes. ICQA is also a distributed algorithm that serves as an approximation solution of Problem 3.

4.5 Simulation Results

Table 4.1: The settings of the simulation networks

Area of sensor field	400m*400m
ρ	20m
R	80m
α, β, γ and ϵ	1.0, 2.0, 1.0 and 100.0
s	0.6

To evaluate the effectiveness of employing ι in sensor-grouping problem, we build simulation surveillance networks. We employ MIND and ICQA to group the in-network sensor nodes. We compare the grouping results with respect to how many groups both algorithms find and how the performance of the resulting groups are.

Detailed settings of the simulation networks are shown in Table 4.1. In simulation networks, sensor nodes are randomly deployed in a uniform manner in the network area.

For evaluating the coverage quality of the sensing area of a node, we divide the sensing area of a node into several regions and regard the coverage quality of the central point in each region as a representative of the coverage quality of the region. This is a numerical approximation. Larger number of such regions results in better approximation. As sensor nodes are with low computational capacity, there is a tradeoff between the number of such regions and the precision of the resulting coverage quality of the sensing area of a node. In our simulation study, we set this number 12. For evaluating the improvement of coverage quality in ICQA, we sum up all the improvements at each region-center as the total improvement.

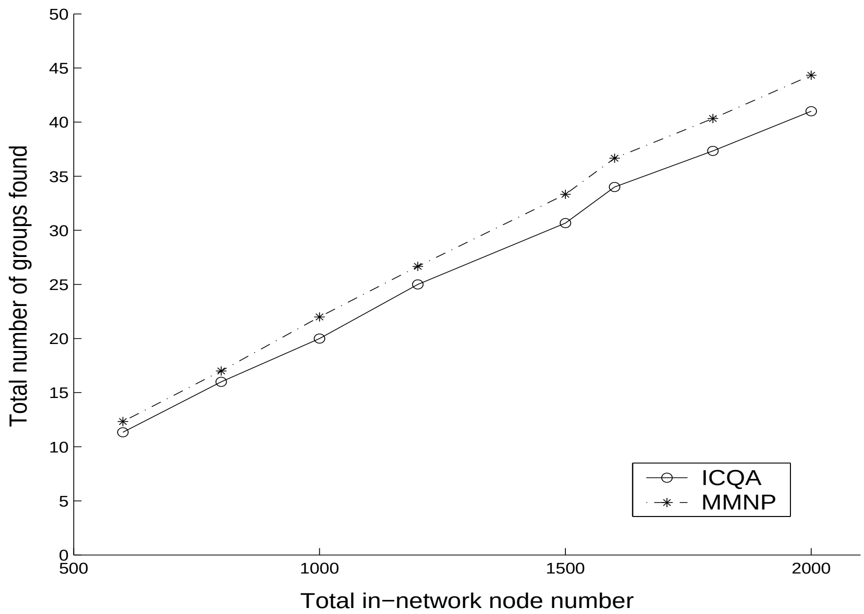

Figure 4.2: The number of groups found by MIND and ICQA

4.5.1 Number of Groups Formed by MIND and ICQA

We set the total in-network node number to different values and let the networks perform MIND and ICQA. For each n, simulations run with several random seeds to generate different networks. Results are averaged. Figure 4.2 shows the group numbers found in networks with different n's.

We can see that MIND always outperforms ICQA in terms of the number of groups formed. Obviously, the larger the number of groups can be formed, the more the redundancy of each group is exploited. This output shows that an approach like MIND that aim to maximize ι of the resulting topology can exploits redundancy well.

Table 4.2: The grouping results of five networks with $n = 1500$

Net	MIND Group Number	ICQA Group Number	MIND Average ι	ICQA Average ι
1	34	31	0.145514	0.031702
2	33	30	0.145036	0.036649
3	33	31	0.156483	0.033578
4	32	31	0.152671	0.029030
5	33	32	0.146560	0.033109

As an example, in case that $n = 1500$, the results of five networks are listed in Table 4.2.

The difference between the average ι of the groups in each network shows that groups formed by MIND result in topologies with larger ι's. It demonstrates that ι is good indicator of redundancy in different networks.

4.5.2 The Performance of the Resulting Groups

Although MIND forms more groups than ICQA does, which implies longer lifetime of the networks, another importance consideration is how these groups formed by MIND and ICQA perform. We let 10000 events randomly occur in the network area except the margin. We compare how many events happen at the locations where the quality is less than the requirement $s = 0.6$ when each resulting group is conducting surveillance work (We call the number of such events the *failure number* of group). Figure 4.3 shows the average failure numbers of the resulting groups when different node numbers are set.

We can see that the groups formed by MIND outperform those formed by ICQA because the groups formed by MIND result in lower failure numbers. This further demonstrates that MIND is a good approach for sensor-grouping problem.

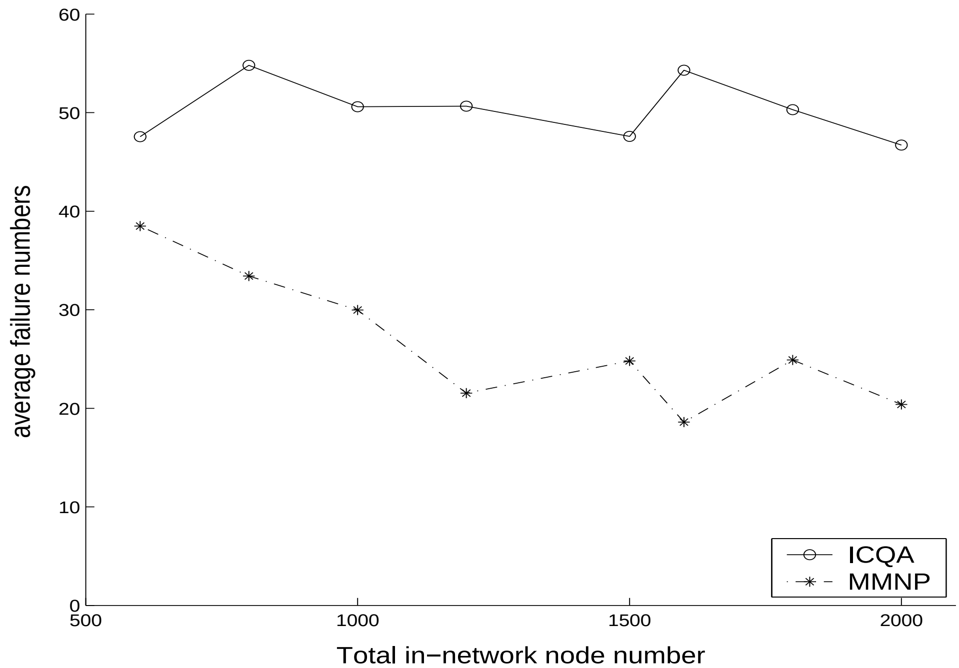

Figure 4.3: The failure numbers of MIND and ICQA

4.6 Conclusion

This chapter proposes ι, a novel index for evaluation of point-distribution. ι is the minimum distance between each pair of points normalized by the average distance between each pair of points. We find that a set of points that achieve a maximum value of ι result in a honeycomb structure. We propose that ι can serve as a good index to evaluate the distribution of the points, which can be employed in coverage-related problems in wireless sensor networks (WSNs). We set out to validate this idea by employing ι to a sensor-grouping problem. We formulate a general sensor-grouping problem for WSNs and provide a general sensing model. With an algorithm called Maximizing-ι Node-Deduction (MIND), we show that maximizing ι at sensor nodes is a good approach to solve this problem. Simulation results verify that MIND outperforms a greedy algorithm that exploits sensor-redundancy we design in terms of the number and the performance of the groups formed. This demonstrates a good application of employing ι in coverage-related problems.

□ **End of chapter.**

Chapter 5

Conclusion

In the recent decade, advances in Micro Electro-Mechanical Systems (MEMS) have made in-situ sensing with wireless sensor networks (WSNs) a promising technique. As wireless integrated network sensors are powered with a small battery and they usually work in an unattended manner, the main constraint of a sensor node is that its energy resource is limited.

To enable in-situ sensing, sensor nodes and WSNs should function in an energy-efficient manner. Energy optimization techniques must be performed in every level of the design of a sensor network system. The work described in this thesis investigates various aspects of power saving approaches to achieve energy-efficient and reliable WSNs. Energy optimization techniques on networking issues in the literatures are classified into the following five categories.

- Route data packets via energy-efficient path, *i.e.*, route data packets so that the energy required for the data transport is minimized.
- Exploit the redundancy of data packets through some techniques such as in-network data aggregation or data fusion, and source reporting rate control.
- Adjust the transmitter power level of a sensor node to let

it communicate with its intended receivers in an energy-efficient way.

- Avoid useless packets, *i.e.*, minimize protocol overheads.
- Schedule the in-network sensor nodes so that they can work in sleep mode to save energy when they are not required to perform sensing tasks or communication tasks.

We first study routing issues and data transportation issues. A WSN is usually employed to conduct one or a few specific tasks; *i.e.*, only one or a few specific applications are running at sensor nodes. Strict layering is not necessary in WSN because applications of a network are always deterministic before the network is set up. Based on the features of WSNs, we discard the common layering network-protocol principle by coupling data transport protocol and applications: let the applications solve an optimization problem and feed back required reporting rates of sources. Based on this consideration, we propose PORT, a Price-Oriented Reliable Transport protocol for wireless sensor networks to reliably and energy-efficiently convey sensor information to the sink. Under the constraint that the sink must obtain reliable information on the phenomenon of interest, PORT minimizes the energy consumptions using two schemes. One is based on the sink's application-based optimization approach that feeds back the optimal reporting rate of each source according to the contribution of the sources and the energy consumption of the sensor-to-sink communication from each source to the sink. The other is a locally optimal routing scheme for in-network nodes according to feedback of downstream communication conditions. The communication-condition estimation is based on an estimation of link-loss rate along the sensor-to-sink traffic path. PORT can obtain the sensor-to-sink communication condition such as congestion and weak links which cause packet loss, and thus it adapts well to network dynamics caused by these factors. We code PORT on the NS-2 network simulation tool. Simulation results in an application case study demonstrate that PORT is an effective transport protocol for reducing energy consumption comparing to existing schemes. Thus, it can prolong the life time and reliability of wireless sensor networks. This work falls into class 1, 2, 4 in the above classification.

In our second work, we examine the problem of transmitter power control for energy-efficient sensor-to-sink communications. We model this problem based on the network and application features of WSNs. An intuitive implementation to solve this problem, namely BOU, is presented. We identify the broadcast explosion problem in BOU, and then improve BOU by allowing a waiting period before each broadcasting. We show that the waiting time should be proportional to the probability that a node would find a more energy-efficient path to the sink, and present an efficient approximation algorithm to calculate the probability. Simulations are designed to evaluate BOU and BOU-WA. The results validate the effectiveness of BOU-WA; specifically, it can set up energy-efficient paths for sensor-to-sink traffic with low overhead in a reasonable converging time. This work falls into class 2, 3 and 4 of the above classification.

In the last work presented in this thesis, we propose ι, a novel index for evaluation of point-distribution. ι is the minimum distance between each pair of points normalized by the average distance between each pair of points. We find that a set of points that achieve a maximum value of ι result in a honeycomb structure. We propose that ι can serve as a good index to evaluate the distribution of the points, which can be employed in coverage-related problems in wireless sensor networks (WSNs). We set out to validate this idea by employing ι to a sensor-grouping problem. We formulate a general sensor-grouping problem for WSNs and provide a general sensing model. With an algorithm called Maximizing-ι Node-Deduction (MIND), we show that maximizing ι at sensor nodes is a good approach to solve this problem. Simulation results verify that MIND outperforms a greedy algorithm that exploits sensor-redundancy we design in terms of the number and the performance of the groups formed. This demonstrates a good application of employing ι in coverage-related problems. This work falls into class 5 of the above classification.

In conclusion, this thesis studies various energy-efficient approaches to achieve reliable and energy-efficient wireless sensor networks. Simulation studies demonstrate the effectiveness of these approaches. In future work, we are particularly interested in approaches which fall into the first of the above classification.

□ **End of chapter.**

Appendix A

List of Research Conducted

- E. C.-H. Ngai, *Y.Zhou*, M. R. Lyu, and J. Liu. Reliable Reporting of Delay-Sensitive Events in Wireless Sensor-Actuator Networks, In Proc. of *The 3rd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS'06)*, Vancouver, Canada, October 9-12, 2006.
- *Y.Zhou*, H. Yang, M. R. Lyu, and E. C.-H. Ngai. A Point-Distribution Index And Its Application to Sensor Grouping in Wireless Sensor Networks, In *Proc. of the International Wireless Communications and Mobile Computing Conference (IWCMC'06)*, Vancouver, Canada, July 3-6, 2006.
- E. C.-H. Ngai, *Y.Zhou*, M. R. Lyu, and J. Liu. DFTT: A Delay-Aware Fault Tolerant Transport Protocol for Wireless Sensor-Actuator Networks, *Technical Report*, April, 2006.
- *Y.Zhou*, M. R. Lyu, and J. Liu. On Setting up Energy-Efficient Paths with Transmitter Power Control in Wireless Sensor Networks, In *Proc. of The 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS'05)*, pp. 440-448, Washington, DC, November 7-10, 2005.
- *Y.Zhou*, M. R. Lyu, J. Liu, and H. Wang. PORT: A Price-Oriented Reliable Transport Protocol for Wireless Sensor Networks, In Proc. of *The 16th IEEE International Sympo-*

sium on Software Reliability Engineering (ISSRE'05), pp. 117-126, Chicago, IL, USA, November 8-11, 2005.

- *Y.Zhou*. References on wireless sensor networks http://www.cse.cuhk.edu.hk/~yfzhou/sensor.html

Appendix B

Algorithms in Chapter 3 and Chapter 4

Algorithm 1 BOU: The basic algorithm

input X
/*X is this node's physical location.*/
$\eta \Leftarrow +\infty$
/*η is the node cost of this node.*/
$downstream\_neighbor \Leftarrow NULL$
/*$downstream\_neighbor$ is the neighboring node to which this node sends sensor-to-sink data packets.*/
$Pr \Leftarrow 0$
/*Pr is the power level setting with which the node sends sensor-to-sink data packets.*/
loop
    Wait until receiving a configuration packet
    $Y \Leftarrow$ The location information obtained from the configuration packet
    $\eta_{neighbor} \Leftarrow$ The node cost information obtained from the configuration packet
    $Pr' \Leftarrow$ The power level calculated with Equation (3.3)
    $\omega \Leftarrow \eta_{neighbor} + cost(e)$
    /*$cost(e)$ is the cost of the edge from this node to the neighbor calculated with Equation (3.6).*/
    if $\omega < \eta$ **then**
        $\eta \Leftarrow \omega$
        $Pr \Leftarrow Pr'$
        $downstream\_neighbor \Leftarrow$ the neighbor that has sent the configuration packet
        Create a configuration packet with the node cost η, the location X and the identity of this node
        Broadcast this configuration packet
    end if
end loop

Algorithm 2 The broadcast scheme in BOU-W

If there is another configuration packet which is scheduled to be broadcasted, cancel it.
Calculate the probability ρ that there exists another path to the sink of which the path cost is smaller than ω
$T = \alpha \cdot \rho$
Schedule that the configuration packet will be broadcasted in T seconds.

Algorithm 3 The broadcast scheme in BOU-WA

1: If there is another configuration packet which is scheduled to be broadcasted, cancel it.
2: Calculate the probability ρ' that there exists a node u such that the path cost of the path from this node to the sink, immediately via the node u and then immediately via the one that sends this node the configuration packet, is smaller than ω
3: $T = \alpha \cdot \rho'$
4: Schedule that the configuration packet will be broadcasted in T seconds.

Algorithm 4 An algorithm to find out optimal point-distribution so that χ is maximized

input n, ϵ and τ
/* ϵ is a small value which indicates the threshold of the improvement of each step of this greedy algorithm. If the improvement is less than ϵ, this algorithm stops. τ is a small value which is the step size by which the points move towards a better χ. ε is a small 2-D random vector. */
for $i = 1$ to n **do**
    $x_i \Leftarrow$ a random variable (2-D vector)
end for
$\chi = 0$
repeat
    $\chi_0 = \chi$
    Calculate the distance between each node-pair
    $\mu \Leftarrow$ the average of the distances
    Find m and n, the distance between them is the shortest one among all the distances
    $\nu \Leftarrow$ this shortest distance
    $\chi = \frac{\nu}{\mu}$
    $x_{tmp} = x_m$
    $x_m = x_m + \frac{(x_m - x_n + \varepsilon)}{||x_m - x_n||} \cdot \tau$
    Update μ
    Update ν
    $\chi_1 = \frac{\nu}{\mu}$
    $x_m = x_{tmp}$
    $x_{tmp} = x_n$
    $x_n = x_n + \frac{(x_n - x_m)}{||x_n - x_m||} \cdot \tau$
    Update μ
    Update ν
    $\chi_2 = \frac{\nu}{\mu}$
    if $\chi_2 < \chi_1$ **then**
        $x_n = x_{tmp}$
        $x_m = x_m + \frac{(x_m - x_n)}{||x_m - x_n||} \cdot \tau$
    end if
until $\chi - \chi_0 < \epsilon$

Bibliography

[1] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Elsevier Ad Hoc Network Journal*, 2(3):325–349, 2005.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on wireless sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, April 2002.

[4] ALERT. Alert systems organization homepage. http://www.alertsystems.org/.

[5] G. Asada, T. Dong, F. Lin, G. Pottie, W. Kaiser, and H. Marcy. Wireless integrated network sensors: Low power systems on a chip. In *Proc. of the European Solid State Circuits Conference (ESSCIRC)*, The Hague, Netherlands, October 1998.

[6] F. Aurenhammer. Vononoi diagram - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(2):345–405, September 1991.

[7] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, Upper Saddle River, 1992.

[8] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. In *Proc. of the First ACM International Workshop on Sensor Networks and Applications (WSNA)*, Atlanta, GA, October 2002.

[9] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. *IEEE Personal Communication*, October 2000.

[10] M. Cardei and D. Du. Improving wireless sensor network lifetime through power aware organization. *ACM Journal of Wireless Networks*, 11(3):333–340, May 2005.

[11] M. Cardei, D. MacCallum, X. Cheng, M. Min, X. Jia, D. Li, and D.-Z. Du. Wireless sensor networks with energy efficient organization. *Journal of Interconnection Networks*, 3(3-4):213–229, December 2002.

[12] M. Cardei, M. Thai, Y. Li, and W. Wu. Energy-efficient target coverage in wireless sensor networks. In *Proc. of the 2005 IEEE Infocom*, Miami, Florida, March 2005.

[13] M. Cardei and J. Wu. Coverage in wireless sensor networks. In M. Ilyas and I. Magboub, editors, *Handbook of Sensor Networks*. CRC Press, 2004.

[14] A. P. Chandrakasan and R. W. Brodersen. *Low Power CMOS Digital Design*. Kluwer, Norwell, MA, 1996.

[15] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Kluwer Wireless Networks*, 8(5):481–494, September 2002.

[16] X. Chen and M. R. Lyu. A sensibility-based sleeping configuration protocol for dependable wireless sensor networks.

CSE Technical Report, The Chinese University of Hong Kong, 2005.

[17] M. Chu, H. Haussecker, and F. Zhao. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications*, 16(3), August 2002.

[18] Crossbow Technology, Inc. *Mote In-Network Programming User Reference Version 20030315*, 2003.

[19] L. Doherty, B. Warneke, B. Boser, and K. Pister. Energy and performance considerations for smart dust. *International Journal of Parallel and Distributed Systems and Networks*, 4(3):121–133, 2001.

[20] D. Estrin, D. Culler, and K. Pister. Connecting the physical world with pervasive networks. *IEEE Pervasive Computing*, 1(1), January-March 2002.

[21] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2001.

[22] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proc. of the Fifth Annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, Seattle, Washington, August 1999.

[23] K. Fall and K. Varadhan. *The ns manual*, Dec. 2003. http://www.isi.edu/nsnam/ns.

[24] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wire-

less sensor networks. *ACM Mobile Computing and Communications Review (MC2R)*, 1(2):295–298, January 2002.

[25] A. Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, 1985.

[26] M. Goemans and D. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comp.*, 24:296–317, 1995.

[27] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proc. of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Seattle, WA, August 1999.

[28] W. Heinzelman, A. Sinha, A. Wang, and A. Chandrakasann. Energy-scalable algorithms and protocols for wireless microsensor networks. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey, June 2000.

[29] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In *Proc. of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, Cambridge, MA, November 2000.

[30] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. of the 6th MobiCom*, Boston, Massachusetts, August 2000.

[31] R. Jain, W. Hawe, and D. Chiu. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *Technical Report DEC-TR-301*, September 1984.

[32] J. Kahn, R. Katz, and K. Pister. Next century challenges: Mobile networking for "smart dust". In *Proc. of the fifth Annual ACM International Conference on Mobile Computing and Networking*, Seattle, Washington, 1999.

[33] H. Karl and A. Willig. A short survey of wireless sensor networks. *TKN Technical Report TKN-03-018*, Technical University Berlin, October 2003.

[34] R. Kershner. The number of circles covering a set. *American Journal of Mathematics*, 61:665–671, 1939.

[35] B. Krishnamachari, D. Estrin, and S. Wicker. Modelling data-centric routing in wireless sensor networks. In *Proc. of the 21st IEEE Infocom*, New York, 2002.

[36] S. S. Kulkarni and L. Wang. MNP: Multihop network reprogramming service for sensor networks. In *Proc. of the 25th International Conference on Distributed Computing Systems (ICDCS)*, June 2005.

[37] F. L. Lewis. Wireless sensor networks. In D. J. Cook and S. K. Das, editors, *Smart Environments: Technologies, Protocols, and Applications*, New York, 2004. John Wiley.

[38] J. Li and P. Mohapatra. An analytical model for the energy hole problem in many-to-one sensor networks. In *Proc. of the IEEE Vehicular Technology Conference*, Dallas, Texas, Fall 2005.

[39] B. Liu and D. Towsley. A study of the coverage of large-scale sensor networks. In *Proc. of the 1st IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS'04)*, Fort Lauderdale, FL, October 2004.

[40] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless sensor networks for habitat moni-

toring. In *Proc. of the ACM International Workshop on Wireless Sensor Networks and Applications*, 2002.

[41] W. Manges, G. Allgood, and S. Smith. It's time for sensors to go wireless. part 1: Technological underpinnings. *Sensors Magazine*, April 1999.

[42] W. Manges, G. Allgood, and S. Smith. It's time for sensors to go wireless. part 2: Take a good technology and make it an economic success. *Sensors Magazine*, May 1999.

[43] S. Megerian, F. Koushanfar, G. Qu, G. Veltri, and M. Potkonjak. Exposure in wireless sensor networks: Theory and pratical solutions. *ACM Journal of Wireless Networks*, 8(5):443–454, 2002.

[44] R. Min, M. Bhardwaj, S. Cho, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan. Low-power wireless sensor networks. In *Proc. of the 14th International Conference on VLSI Design*, Bangalore, India, January 2001.

[45] R. Min, M. Bhardwaj, S. Cho, A. Sinha, E. Shih, A. Wang, and A. Chandrakasan. An architecture for a power aware distributed microsensor node. In *Proc. of the IEEE Workshop on signal processing systems (SIPS)*, October 2000.

[46] S. S. P. Levis, N. Patel and D. Culler. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proc. of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*, 2004.

[47] J. Paek, K. Chintalapudi, J. Cafferey, R. Govindan, and S. Masri. A wireless sensor network for structural health monitoring: Performance and experience. In *Proc. of the 2nd IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*, Syndney, Australia, May 2005.

[48] G. Pottie and W. Kaiser. Wireless integrated network sensors. *Communications of ACM*, 43(5), May 2000.

[49] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40C50, 2002.

[50] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *Proc. of the 19th IEEE Infocom*, Tel Aviv, Israel, March 2002.

[51] T. Rappaport. *Wireless Communications: Principles and Practices (2nd Edition)*. Prentice Hall, Upper Saddle River, 2002.

[52] P. Rentala, R. Musunuri, S. Gandham, and U. Saxena. Survey on sensor networks. *Technical Report, UTDCS-33-02*, University of Texas at Dallas, 2002.

[53] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, 17(8):1333–1344, 1999.

[54] Y. Sankarasubramaniam, O. Akan, and I. Akyildiz. ESRT: event to sink reliable transport in wireless sensor networks. In *Proc. of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, June 2003.

[55] P. Santi. A bibliography on topology control in ad hoc networks. A collection of papers before 2003 on Topology Control.

[56] P. Santi. The critical transmitting range for connectivity in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 4(3), May-June 2005.

[57] P. Santi, D. Blough, and F. Vainstein. A probabilistic analysis for the range assignment problem in ad hoc networks. In *Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 212–220, Long Beach, October 2001.

[58] C. Schurgers and M. Srivastava. Energy efficient routing in wireless sensor networks. In *Proc. of the 2001 MILCOM*, volume 1, pages 357–361, 2001.

[59] L. Schwiebert, S. K. S. Gupta, and J. Weinmann. Research challenges in wireless networks of biomedical sensors. In *Proc. of the ACM International Conference Mobile Computing and Networking*, pages 151–165, 2001.

[60] R. Shah and J. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proc. of the IEEE Wireless Communications and Networking Conference*, Orlando, FL, March 2002.

[61] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *Proc. of the IEEE International Conference on Communications (ICC'01)*, volume 2, Helsinki, Finland, June 2001.

[62] M. B. Srivastava, R. R. Muntz, and M. Potkonjak. Smart kindergarten: sensorbased wireless networks for smart developmental problem-solving enviroments. In *Proc. of the ACM International Conference Mobile Computing and Networking*, 2001.

[63] F. Stann and J. Heidemann. RMST: reliable data transport in sensor networks. In *Proc. of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[64] T. Stathopoulos, J. Heidemann, and D. Estrin. A remote code update mechanism for wireless sensor networks. *Technical Report*, UCLA, 2003.

[65] D. Tian and N. D. Georganas. A node scheduling scheme for energy conservation in large wireless sensor networks. *Wireless Communications and Mobile Computing*, 3(2):272–290, May 2003.

[66] Y. Tseng, S. Ni, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8(2/3):153–167, March 2002.

[67] M. Tubaishat and S. Madria. Sensor networks: an overview. *IEEE Potentials*, 22(2):20–23, April 2003.

[68] C. Wan, A. Campbell, and L. Krishnamurthy. Psfq: A reliable transport protocol for wireless sensor networks. In *Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, September 2002.

[69] C. Wan, S. Eisenman, and A. Campbell. Coda: Congestion detection and avoidance in sensor networks. In *Proc. of the 1st ACM International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, November 2003.

[70] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proc. of the First ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, CA, November 2003.

[71] R. Wattenhofer, P. Bahl, L. Li, and Y. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *Proc. of the 2001 IEEE Infocom*, April 2001.

[72] G. Xing, C. Lu, R. Pless, and J. A. O'Sullivan. Co-Grid: an efficient converage maintenance protocol for distributed sensor networks. In *Proc. of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN)*, Berkeley, CA, April 2004.

[73] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. D. Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Sensor Networks*, 1(1), 2005.

[74] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A wireless sensor network for structural monitoring. In *Proc. of the ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Baltimore, MD, November 2004.

[75] T. Yan, T. He, and J. A. Stankovic. Differentiated surveillance for sensor networks. In *Proc. of the First ACM International Conference on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, CA, November 2003.

[76] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. PEAS: A robust energy conserving protocol for long-lived sensor networks. In *Proc. of the 23rd International Conference on Distributed Computing Systems (ICDCS)*, Providence, Rhode Island, May 2003.

[77] Y. Zhou. References on wireless sensor networks. http://www.cse.cuhk.edu.hk/~yfzhou/sensor.html.

[78] Y. Zhou, M. R. Lyu, and J. Liu. On setting up energy-efficient paths with transmitter power control in wireless sensor networks. In *Proc. of the 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS)*, Washington, DC, November 2005.

[79] Y. Zhou, M. R. Lyu, J. Liu, and H. Wang. PORT: A price-oriented reliable transport protocol for wireless sensor networks. In *Proc. of the 16th IEEE International Symposium on Software Reliability Engineering (ISSRE)*, Chicago, IL, November 2005.

[80] Y. Zhou, H. Yang, M. R. Lyu, and E. C.-H. Ngai. A point-distribution index and its application to sensor-grouping in wireless sensor networks. In *Proc. of the 2006 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Vancovour, Canada, July 2006.