Data aggregation for wireless sensor networks: an energy-efficient approach

Huang, Jiajia

2018

Huang, J. (2018). Data aggregation for wireless sensor networks: an energy-efficient approach. Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/73212>

<https://doi.org/10.32657/10356/73212>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**Data Aggregation for Wireless Sensor Networks:
An Energy-Efficient Approach**

HUANG JIAJIA

**School of Electrical and Electronic
Engineering**

2017

Data Aggregation for Wireless Sensor Networks: An Energy-Efficient Approach

HUANG JIAJIA

HUANG JIAJIA

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

2017

Acknowledgments

First and foremost, I would like to thank my supervisor, Boon-Hee Soong, for his patience and guidance through my study. I appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating.

Besides, my gratefulness should also go to all of the excellent lab-mates and staffs from the Centre for Infocomm Technology (INFINITUS), Edwin Chan, Yee Loon Sum, Hongxin Shao, Yunxiang Liu and Heng Zhang.

Thanks to my friends, my time at Nanyang Technological University was enjoyable. I am grateful to my supporting roommates Dan Wu, Xinyan Li and Mengting Yu for their encouragement and support during my ups and downs. I am also grateful to my friend Ming Zhao for the constructive discussions and advice.

Lastly, I would like to thank my family for all their love and accompany. For my parents who give me unconditional love and encouragement for all my decisions. For my younger brother Wei Huang who grows up together with me. And most of all for my encouraging, supportive, and patient fiancé Bo Hu whose understanding and support during the whole journey of this Ph.D. is so appreciated.

Contents

Acknowledgments	i
Contents	ii
Summary	vii
List of abbreviations	x
List of symbols	xi
List of tables	xii
List of figures	xiii
1 Introduction	1
1.1 Background	1
1.2 Motivation	5
1.3 Objectives	6
1.4 Thesis Main Contributions	7
1.5 Organization of the Thesis	10
2 Literature Review	11
2.1 Overview of Data Aggregation	12
2.2 Function-Based Data Aggregation	13

Contents

2.2.1	Classification of Aggregation Function	14
2.2.2	Structured and Unstructured Function-Based Data Aggregation	17
2.3	Correlation-Based Data Aggregation	20
2.3.1	Distributed Source Coding	20
2.3.2	CS Based Data Aggregation	21
2.3.3	Applying CS into WSNs: the State-of-the-Art	26
2.4	Energy Harvesting Wireless Sensor Networks	30
2.5	Concluding Remarks	32
3	Function-Based Data Aggregation	33
3.1	Introduction	34
3.2	Network Model	37
3.2.1	Interference Model	39
3.3	Hybrid Scheduling Assignment	39
3.3.1	Priority-Aware Time Slot Allocation	40
3.3.2	CDMA Code Sequence Allocation	41
3.3.3	Transmission Power Allocation	42
3.3.4	Decentralized Operation of Proposed Algorithm	42
3.4	Computation Failure Probability	44
3.4.1	Using Iterative Method to Solve for P_e and α_N	46
3.4.2	Trade-off Between Computation Failure Probability and Computation Latency	47
3.5	Average Computation Time	48
3.5.1	Using Interior Barrier Function Method to Solve for Optimal α_N and τ	49
3.6	Performance Evaluation	50

Contents

3.6.1	Average Computation Time	51
3.6.2	Energy Consumption	53
3.7	Concluding Remarks	54
4	Energy-Aware Hybrid Compressive Sensing for Data Gathering Sensor Networks	56
4.1	Introduction	57
4.2	System Model	61
4.2.1	Network Model	61
4.2.2	Hybrid CS Based Data Gathering	62
4.2.3	Problem Description	65
4.3	Problem Formulation	66
4.3.1	Hybrid CS Based Tree Construction Constraints	67
4.3.2	Scheduling Constraints	68
4.4	Complexity Analysis	70
4.5	Approximation Algorithms	71
4.5.1	Hybrid CS Based Data Gathering Tree Construction	72
4.5.2	SINR-Based Priority-Aware Link Scheduling	73
4.6	Performance Analysis	77
4.6.1	Approximation ratio of the tree construction algorithm	77
4.6.2	Bounds of the scheduling algorithm	77
4.7	Performance Evaluation	78
4.8	Concluding Remarks	83
5	Cost-Aware Stochastic Compressive Data Gathering for Wireless Sensor Networks	85
5.1	Introduction	86

Contents

5.2	System Model	89
5.2.1	Network model	89
5.2.2	Recovery of Compressible Data	90
5.2.3	Stochastic Compressive Data Gathering	91
5.2.4	Equivalent Measurement Matrix and Compression Probability	93
5.2.5	Expected Energy Cost	95
5.2.6	The Randomness of the Data Gathering Process	96
5.3	Problem Formulation	96
5.4	Optimal Compression Probability	98
5.5	Data Recovery Using Sketching Decoder	100
5.6	Performance Evaluation	102
5.7	Concluding Remarks	108
6	Harvesting-Aware Compressed Data Gathering for Optimal Sustainable Information Throughput in EH-WSNs	109
6.1	Introduction	110
6.2	Harvesting-Aware Compressed Data Gathering Framework	113
6.2.1	Network Model	113
6.2.2	Energy Neutrality and Cost-Harvest Ratio (CHR)	114
6.2.3	Sustainable Information Throughput	115
6.2.4	Harvesting-Aware Compressed Data Gathering and Recovery	116
6.2.5	CS Recovery	116
6.3	MINLP Formulation	118
6.4	Hardness of the Problem	122
6.5	Approximation Algorithms	123
6.5.1	CHR Based Compressed Data Gathering Tree	123
6.5.2	Graph-Based Priority-Aware Link Scheduling (GPLS)	125

Contents

6.6	Performance Analysis	127
6.6.1	Approximation Ratio of the Tree Construction Algorithm . .	127
6.6.2	Bound of Scheduling Algorithm	127
6.7	Performance Evaluation	128
6.8	Concluding Remarks	133
7	Conclusions and Future Work	136
7.1	Conclusions	136
7.2	Future Work	139
	List of publications	142
	Bibliography	143

Summary

As a major component of Internet of Things (IoT), pervasive wireless sensor networks (WSNs) are becoming ubiquitous in all aspects of life such as environmental monitoring, military, intelligent transportation. Data gathering is a crucial yet energy-consuming task for WSNs. More and more sensing devices are deployed, which results in a huge amount of data to be transmitted. Usually, the sensor node is equipped with tiny, non-rechargeable battery and limited computation capability. Consequently, it's imperative to improve energy efficiency and prolong network lifetime for WSNs powered by non-rechargeable batteries. In addition to non-rechargeable battery, energy harvesting (EH) has emerged to be a promising power supply to WSNs, which may unlock the possibility of perpetual operation for WSNs. In practical environment, data redundancy and data correlation commonly exist, which motivates the development of data aggregation techniques for both battery powered and energy harvesting WSNs (EH-WSNs). Data aggregation can be categorized as function-based and correlation-based. This thesis aims to develop both function-based and correlation-based data aggregation algorithms for conventional WSNs powered by non-rechargeable battery to achieve energy efficiency and prolong network lifetime. Further, this thesis applies data aggregation into EH-WSNs to achieve perpetual operation and satisfying application requirements.

First of all, this thesis proposes a function-based data aggregation algorithm for a single-hop sensor network. In certain applications such as alarm systems, only

the maximum value is required. Therefore, function-based data aggregation can be used to reduce energy consumption as well as transmission delay. A priority-aware hybrid scheduling scheme is proposed to achieve fast decentralized energy-efficient MAX function computation in a single-hop network. In our proposed scheme, priority-aware TDMA scheme is used to schedule the transmission. CDMA is used to recover information whenever a collision occurs. The proposed scheme helps to reduce the number of transmissions and delay for MAX computation with good scalability.

Secondly, this thesis focuses on correlation-based data aggregation algorithms based on compressive sensing (CS). Leveraging the load balancing features of CS, a novel hybrid CS based data gathering algorithm with a dense measurement matrix for tree-based multi-hop data gathering network is developed. CS technique can guarantee the full recovery of sensor data while imposing simple computation load on the sensor. An optimization problem is formulated and solved by approximation algorithms. The proposed scheme significantly reduces total energy consumption and data gathering latency and dramatically improves network lifetime compared with benchmark algorithms.

The sensor network may experience changing environment where deterministic routing path is vulnerable to attacks and is less energy-efficient. In this case, it would be beneficial to allow a certain amount of exploration in the routing path. A cost-aware stochastic compressive data gathering algorithm is developed for multi-hop WSNs. The mathematical Markov chain based model is used to analyze the stochastic compressive data gathering process. A sparse random measurement matrix with optimized compression probability is constructed to minimize expected total cost while guaranteeing the recovery accuracy. Extensive simulations on both synthetic and real data show that the proposed algorithm requires less total expected cost to achieve a certain level of accuracy compared with benchmark algorithm. Finally, the proposed algorithm prolongs network lifetime for a given

recovery accuracy.

Finally, this thesis proposes a harvesting-aware compressed data gathering algorithm, in which the heterogeneous EH rates of the nodes are taken into consideration in order to maximize information throughput while maintaining sustainable operation. An optimization problem is formulated and solved by approximation algorithms. Our proposed algorithm can significantly improve information throughput while maintaining sustainable operation.

List of Abbreviations

BP	Basis pursuit
CDF	Cumulative distribution function
CDMA	Code division multiple access
CS	Compressive sensing
CHR	Cost-harvest ratio
DAG	Directed acyclic graph
DCT	Discrete cosine transform
DSC	Distributed source coding
EH	Energy harvesting
EH-WSNs	Energy harvesting wireless sensor networks
GPLS	Graph-based priority-aware link scheduling
HCS-DGT	Hybrid CS based data gathering tree
IoT	Internet of Things
i.i.d	Independent and identically distributed
KKT	Karush-Kuhn-Tucker
MBST	Minimum bottleneck spanning tree
MP	Matching pursuit
MINLP	Mixed integer nonlinear programming
MAC	Medium access control
RIP	Restricted isometry property
SINR	Signal to interference and noise ratio
SPLS	SINR-based priority-aware link scheduling
TDMA	Time division multiple access
WSNs	Wireless sensor networks

List of Symbols

$\mathbb{R}$	Real numbers
$\mathbb{R}_+$	Positive real numbers
$\mathbb{R}^n$	Real $n \times 1$ matrices
$\mathbb{R}^{m \times n}$	Real $m \times n$ matrices
$\mathbb{C}$	Complex numbers
$\mathbb{Z}$	Integers
$\mathbb{Z}_+$	Non-negative integers
$\mathbf{u}^{n \times 1}$	$n \times 1$ data vector
Ψ	Transform matrix
Φ	Measurement matrix
$\ x\ _1$	ℓ_1 -norm of vector x
$\ x\ _2$	Euclidean (or ℓ_2 -) norm of vector x
$\lceil x \rceil$	The ceiling function of x

List of Tables

2.1	Summary of aggregation function types.	15
2.2	The classification of function-based data aggregation.	17
2.3	Classification of CS based data aggregation	26
4.1	Notations in problem formulation	66
6.1	Notations in MINLP formulation	119

List of Figures

1.1	The number of connected devices over 2 decades [1].	2
2.1	Taxonomy of data aggregation algorithms	13
2.2	CS measurement operation.	24
3.1	A single-hop network. The dashed circle with a node in its center is the communication area of the node. The radius of the dashed circle denotes the communication range. d_1, d_2 are the distance from sensor node 1 and 2 to the sink node, respectively. $d_1, d_2 \leq r$	37
3.2	Time slot allocation illustration.	41
3.3	Operation of hybrid scheduling-based maximum function computation.	43
3.4	Failure probability as a function of maximum waiting time Γ for different spreading gain.	47
3.5	Average waiting time τ_1 as a function of spreading gain G	52
3.6	Average waiting time τ_1 as a function of network size K	53
3.7	Average computation time τ as a function of network size K	54
3.8	Average transmission energy E_{tx} as a function of network size K	55
4.1	Comparison of non-CS, plain CS and hybrid CS based data gathering schemes.	63
4.2	Recovery error as a function of CS factor. $n = 400$	78

LIST OF FIGURES

4.3	Performance of non-CS, plain-CS, and hybrid CS based data gathering under different network sizes.	80
4.4	Performance of non-CS, plain-CS, and hybrid CS based data gathering under different CS factors.	81
5.1	Stochastic compressive data gathering.	91
5.2	The solution framework of cost-aware stochastic compressive data gathering.	98
5.3	Synthetic data vector and the corresponding transform coefficients using Fourier basis.	103
5.4	Performance of proposed algorithm on synthetic data.	104
5.5	Performance of proposed algorithm on real data.	105
5.6	Network lifetime vs the recovery error. Synthetic data, network size $n = 200$	107
5.7	Network lifetime vs the recovery error. Real data, DCT transform, network size $n = 54$	108
6.1	Two states of an EH node.	114
6.2	Comparasion of CS-based and Non-CS data gathering scheme. . . .	117
6.3	Bottleneck traffic for different network sizes and node densities . . .	129
6.4	Information throughput for different network sizes and node densities.	131
6.5	Information throughput under different CS factor and node densities.	132
6.6	Information throughput under different EH material size and node densities.	134

Chapter 1

Introduction

1.1 Background

The era of internet of things (IoT) has come. IoT enables objects to sense, communicate, and make decisions by exploiting underlying technologies [2,3]. A large set of applications such as predictive maintenance, smart city, industrial automation, and intelligent transportation are gradually becoming reality thanks to technology advances. The functionality of IoT is realized with the help of pervasive wireless sensor networks (WSNs). In order to make the objects smart, more and more sensing devices have been put into use. The number of sensing devices is growing fast. Specifically, the number of sensing devices is predicted to hit 50 billion by the year of 2020 [4]. The growing trend of the connected devices from 1992 to 2020 is shown in Fig. 1.1. Nevertheless, the huge amount of connected devices bring both convenience as well as challenges. The huge amount of devices create an ever growing deluge of data to be transmitted, processed, and stored. As reported in [5], the data generated worldwide is 1250 billion GB in 2010 and is growing by 58% per year.

Wireless sensor network (WSN) is one of the crucial components of IoT. A

Figure 1.1: The number of connected devices over 2 decades [1].

WSN is a network which consists of sensor nodes that are equipped with versatile sensing abilities and limited power supply. The sensor nodes are usually deployed in a certain region to collect data about the environment. The collected data are transmitted wirelessly to the sink node for further decision making. As for deployment method, sensor nodes can be either randomly deployed or in a pre-determined way to achieve certain optimization. WSNs have several distinctive features that bring both flexibility and design challenges. The sensor nodes are relatively small in size. The tiny size of the sensor node have facilitated deployment in some applications such as body area network. Nonetheless, the compact size comes at the cost of stringent resource constraints. In particular, the sensor nodes have limited computation capability and storage. Hence, sensor nodes cannot afford complex computation. Moreover, sensor nodes are extremely energy-constrained since they are powered by limited-sized batteries. In addition, the sensor nodes are usually deployed in environment where human intervention is impossible, which makes it

hard to replace or recharge the battery. Energy depletion of sensor nodes significantly degrades the performance of the application. For example, in a bridge structural monitoring system, if the relay nodes die out, the message cannot be delivered to the sink node, and further decision cannot be made. Consequently, conserving energy and prolonging network lifetime are primary goals of WSNs. Generally, network lifetime is defined as the time span from sensor deployment to the instant when the network is considered nonfunctional [6]. The definition of nonfunctional is application-specific.

WSNs have a large set of applications spanning from the field of agriculture, smart building, and structural health monitoring to military [7–9]. Essentially, these applications can be categorized into two types: monitoring and tracking. Monitoring applications focus on collecting the status data of a relatively slow-changing phenomena and sending the data back to a sink node. Examples include environmental monitoring, bridge or building structural health monitoring and medical health monitoring. Tracking applications typically involve detecting and locating a particular moving object. Namely, the tracking of animals, vehicles, and individuals. Various applications may have specific characteristics and requirements such as latency, data accuracy and throughput.

Data gathering is the most crucial process of the WSNs for all the applications. Data gathering refers to the process where sensor nodes sense the environment and transmit readings to the sink node. Whereas, data gathering is also an energy-consuming task. It's challenging to achieve a trade-off between the energy efficiency and application requirements. Extensive research works on energy-efficient data gathering have been done to satisfy both energy constraint and application requirements, such as power control and medium access control (MAC) algorithms [10]. Conventionally, energy-efficient data gathering algorithms focus on raw data gathering, where the data is collected and transmitted without any processing at the source node and intermediate nodes. It is shown that tra-

ditional raw data gathering algorithms are not efficient enough to cope with the tremendous amount of data generated by new applications.

Data aggregation emerges as a promising paradigm for energy-efficient data gathering. Data aggregation is defined as the process of combining the data from multiple sensors to eliminate redundant transmission and provide aggregated information at the sink node [11]. Based on the information recovery at the sink node, data aggregation can be categorized into two types. The first is the function-based data aggregation which delivers only a function value to the sink node. The other is correlation-based data aggregation, where the goal is to aggregate the data for transmission and recover all the sensor readings at the sink node.

Energy-conserving algorithms can prolong network lifetime for battery-powered WSNs. Nevertheless, the sensor node will eventually die out. In addition to limited-sized, non-rechargeable battery, energy harvesting (EH) has emerged to be an alternative way to power the sensor nodes. EH is to harvest energy from ambient sources to power the sensor nodes. Potential energy sources include solar, wind, vibration, and radio waves. The advances of EH techniques may gradually unlock the possibility of perpetual operation of the WSNs. Whereas, the sporadic nature of the energy source has made it difficult to continuously power the sensor node. Hence, adaptive transmission schemes are expected to achieve perpetual operation while satisfying application requirements.

In brief, for battery-powered WSNs, technical challenges arise from achieving the trade-off between energy efficiency and application requirements when a large amount of data need to be transmitted. WSNs powered by EH techniques have the potential to achieve perpetual operation. However, challenges still exist on managing the harvested energy and data transmission. This thesis presents a comprehensive review about the state-of-the-art development of data aggregation algorithms and EH algorithms. Moreover, this thesis proposes a suite of energy-efficient algorithms base on data aggregation for battery-powered WSNs in various

network scenarios. A novel harvesting-aware data aggregation algorithm is proposed for energy harvesting wireless sensor networks (EH-WSNs).

1.2 Motivation

Data gathering is an important yet resource-consuming task for sensor nodes. Achieving energy efficiency and prolonging network lifetime are two primary objectives in designing protocols for WSNs. Further, more application requirements such as latency and data accuracy need to be satisfied. Given the dilemma between the tremendous demand of data collection tasks and the limited resource of sensor nodes, technical challenges arise to strike a balance between energy efficiency and other system requirements. Even the sensor nodes can harvest energy from the environment, the harvested energy should be managed wisely so that sustainable operation can be maintained.

Conventionally, sensor nodes transmit and forward the raw data without intermediate processing through a certain route. Raw data gathering would be less efficient when an overwhelming amount of data are waiting for transmission. Energy-efficient algorithms have been developed for raw data gathering focusing on power management such as duty-cycle management [12, 13], optimal sensor node placement [14], and power control [15]. In duty-cycle management algorithms, nodes are put into sleep mode from time to time in order to conserve energy and prolong the network lifetime. Node placement is to deploy the nodes such that the target area is covered and system requirements are met. Power control algorithms conserve energy by adjusting the transmitting power to maintain successful data delivery rather than transmitting at full power capability [15]. Nonetheless, all these aforementioned methods pay no attention to the data and may not be able to deal with large amount of data.

In practical environment, data redundancy and data correlation commonly

exist. In addition, in some applications, the users are only interested in a small portion of the data. If the sensors only transmit the data that are of concern, significant amount of energy can be saved. Leveraging the fact that computation energy is less than transmission energy, researchers have been investigating methods to trade off data transmission for computation complexity. The aforementioned reasons have motivated the development of data aggregation as a promising method to combat data deluge. Data aggregation exploits data processing at the intermediate nodes to combine information from multiple nodes, which eliminates data transmissions and conserves energy. Both function-based and correlation-based data aggregation achieve energy efficiency by reducing the number of transmissions.

Although energy-efficient algorithms can prolong network lifetime for battery-driven WSNs, the sensor nodes will eventually deplete their energy. In practical environment, the sensor nodes are required to work for months or even years without maintenance. Moreover, several problems are associated with the batteries itself. First, current leakage happens even when the node is not activated, which wastes a portion of the valuable energy [16]. Further, the energy density of the battery is limited, which may prevent the sensor node from working for a long period of time [17]. The above facts motivate the introduction of EH into WSNs. EH provides an alternative way to power the sensor node, which may boost the perpetual and maintenance-free operation of the WSNs.

1.3 Objectives

The overall objective of this thesis is twofold. First, for WSNs powered by limited-sized, non-rechargeable batteries, the objective is to design data aggregation based algorithms to achieve a satisfying trade-off between energy-efficiency and application requirements. Further, for WSNs that can harvest energy from the environment, the focus is no longer on conserving energy. Instead, the objective becomes

more information centric. For instance, given limited harvested energy, it's important that the transmission protocols are designed in a way that can transmit more information while achieving sustainable operation. More specifically, the following research objectives should be fulfilled.

- To thoroughly review the state-of-the-art data aggregation algorithms and EH algorithms for WSNs, identify the research challenges and categorize the research directions.
- To propose a function-based data aggregation algorithm to efficiently deliver a function value to the sink node while achieving the trade-off between energy efficiency, latency and data accuracy.
- To develop a correlation-based data aggregation algorithm for a multi-hop network to reduce the number of transmissions while guaranteeing the recovery of all the information with certain fidelity.
- To design a robust, energy-efficient correlation-based data gathering algorithm for sensors working in dynamic environment to achieve both energy efficiency and data recovery quality.
- To propose a correlation-based algorithm for EH-WSNs to maximize the information throughput while achieving perpetual operation.

1.4 Thesis Main Contributions

In order to address the challenges of the data deluge era, data aggregation based solutions are proposed to tackle the problems from different perspectives. For WSNs powered by non-rechargeable batteries, this thesis aims to develop energy efficient data aggregation algorithms (both function-based and correlation-based) to achieve good trade-off between energy efficiency and data fidelity and latency.

For EH-WSNs, this thesis seeks to develop correlation-based algorithm to maximize information throughput while maintaining sustainable operation. More specifically, this thesis has made the following contributions.

First of all, this thesis proposes a function-based data aggregation algorithm for fast and energy-efficient function value computation in a single-hop network. This algorithm is suitable for applications that require only the maximum value of the whole data set. Individual values are not the concern. Priority-aware time division multiple access (TDMA) is used to schedule the transmission of the sensory data. Code division multiple access (CDMA) technique is used to improve success probability when collision occurs. The technical contribution is twofold. First, the expression for computation failure probability is derived and the optimal value is found using iterative method. Further, an optimization problem is formulated to minimize average computation time subjected to the constraints on failure probability. The synergy of CDMA and priority-aware TDMA contributes to reduction of computation failure probability as well as average computation delay. Extensive simulation results show that the proposed algorithm significantly improves success probability and reduces gathering latency.

Second, this thesis focuses on correlation-based data aggregation algorithms based on compressive sensing (CS). Taking heterogeneous transmission cost into consideration, an energy-efficient, low latency, and general data gathering scheme based on hybrid CS with a dense measurement matrix is proposed. A multi-objective mixed integer nonlinear programming (MINLP) problem is formulated with the objective to prolong network lifetime and reduce data gathering latency. Since the MINLP problem is NP-hard, it is decomposed into an energy-efficient hybrid tree construction problem and an SINR-based scheduling problem. Then, two approximation algorithms with provable bounds are proposed to solve the two decomposed problems respectively. Finally, extensive simulation results demonstrate the performance of the proposed algorithm in terms of recovery quality,

total energy consumption, data gathering latency, and network lifetime. The simulation results show that, compared with benchmark algorithms, the proposed scheme significantly reduces total energy consumption and data gathering latency and dramatically improves network lifetime with satisfying recovery quality.

Next, in practical environment, the sensor network may experience changing environment where deterministic routing path is vulnerable to attacks and is less energy-efficient. In this case, it would be beneficial to allow a certain amount of exploration in the routing path. Taking the changing environment into account, a cost-aware stochastic compressive data gathering algorithm for wireless sensor network working in changing environment is developed. The mathematical Markov chain based model is used to analyze the stochastic compressive data gathering process. A sparse random measurement matrix with optimized compression probability is constructed to minimize expected total cost while guaranteeing the recovery accuracy. Extensive simulations on both synthetic and real data show that the proposed algorithm requires less total expected cost to achieve a certain level of accuracy compared with the benchmark algorithm. Further, the proposed algorithm demonstrates prolonged network lifetime for a given recovery accuracy.

Finally, the possibility of exploiting CS techniques in an EH-WSNs is explored. The advances in EH techniques provide a possible alternative to power the sensor nodes, which may unlock the possibility of perpetual operation of the WSNs. In sensor network powered by ambient renewable energy sources, network lifetime is no longer a critical concern. Instead, designing communication protocol to improve sustainable information throughput becomes crucial. A harvesting-aware compressed data gathering scheme is proposed to maximize the information throughput while guaranteeing the sustainable operation of the network. A MINLP problem is formulated to maximize the throughput subjected to constraints on energy neutrality. Numerical results are provided to show that the proposed CS based algorithm significantly improves the information throughput for an EH-WSN. Fur-

thermore, our proposed algorithm requires smaller material size to achieve the same information throughput.

1.5 Organization of the Thesis

In this chapter, we introduced the background and definition of the problems, as well as the motivation and contributions of this thesis. Sequentially, the organization of the thesis is described as follows. Chapter 2 provides a survey on related literature of data aggregation algorithms and EH for WSNs. An overview of data aggregation is given. Followed by the definitions and literature for function-based data aggregation and correlation-based aggregation. Finally, related research works of EH-WSNs are reviewed. Chapter 3 presents a function-based data aggregation algorithm. More specifically, this chapter presents a priority-aware hybrid scheduling for fast and energy-efficient MAX function computation in a single-hop network. Chapter 4 presents our first work on correlation-based data aggregation. An energy-efficient, low latency, and general data gathering scheme based on hybrid CS is proposed. The data gathering problem is characterized as a multi-objective MINLP problem, which can be solved by two approximation algorithms. Chapter 5 describes the second work on correlation-based data aggregation. This chapter presents a cost-aware stochastic compressive data gathering algorithm for wireless sensor network working in changing environments. The mathematical Markov chain based model is used to analyze the stochastic compressive data gathering process. Chapter 6 presents a method to apply CS into EH-WSNs to improve the information throughput while achieving energy neutrality. Chapter 7 concludes this thesis while the possible future works are also discussed.

Chapter 2

Literature Review

Pervasive WSNs have enabled the burgeoning development of IoT. Sensor nodes are small in size, and are equipped with only limited batteries and computation capabilities. Data gathering is a crucial functionality of WSNs. At the same time, the pervasive applications of sensors generate a deluge of data that challenge the traditional way of data transmission and processing. Further, with the limited resources, different applications may have various requirements on delay and data fidelity. Data transmission costs significantly more energy than computation [18]. For example, it has been shown that the energy to transmit one byte data equals to the energy of thousands or even millions of computation operations for different hardware [19]. Intuitively, researchers start to investigate algorithms to exploit computation complexity to reduce communication energy consumption. Data aggregation has emerged to be such a method to reduce data transmission volume by exploiting data processing at intermediate nodes. The decrease in data traffic eventually results in energy conservation and prolonged network lifetime. Apart from network lifetime and energy efficiency, there are several other key considerations for data gathering in WSNs, namely, the latency and data accuracy. Latency measures the delay involved in data transmission and indicates the freshness of the data. In some applications, data freshness is a critical concern [20]. Data accuracy

measures the quality of the data received at the sink node. It's a great challenge to achieve a trade-off between these performance metrics for data aggregation algorithms. Moreover, with the advances of EH techniques, there is a possibility to achieve sustainable operation for the sensor networks. Still, the problem of managing the harvested energy while achieving system requirements is a great challenge.

This chapter surveys the literature of different categories of data aggregation algorithms for energy-constrained WSNs. Moreover, this chapter also reviews the research progress of EH-WSNs. The remainder of this chapter is organized as follows. Section 2.1 provides an overview of the data aggregation algorithms. The data aggregation algorithms are categorized as function-based and correlation-based. Section 2.2 introduces of function-based data aggregation. Section 2.3 introduces correlation-based data aggregation which includes distributed source coding (DSC) based algorithm and CS based algorithm. Section 2.4 introduces the development of EH-WSNs. Finally, Section 2.5 concludes this chapter.

2.1 Overview of Data Aggregation

Data aggregation, also referred to as data fusion, is defined as the process of aggregating data from multiple sensors to eliminate redundant transmissions while providing quality aggregated information at the sink node [11]. Only the most critical information is transmitted to the sink node. Exploiting data redundancy and data correlation, data aggregation algorithms can achieve the trade-off between energy efficiency and other system requirements such as latency and data accuracy. The underlying reasons to use data aggregation are twofold. First, data redundancy commonly exists in sensory data, which can be exploited to reduce data transmission. Second, in some applications, the users are only interested in a small portion of the data, if we only transmit the data that are of concern, significant amount of energy can be saved. Accordingly, data aggregation algorithms can be catego-

Figure 2.1: Taxonomy of data aggregation algorithms

rized into two types. The first is function-based data aggregation, where the sink node only receives statistical information (MAX/MIN/MEDIAN) about the sensed data set, the rest of the information is lost. The other is correlation-based data aggregation, where the data correlation is exploited to reduce data transmission, the sink node can recover the original sensory data vector with a certain level of fidelity. Function-based data aggregation can be further categorized as structured and unstructured algorithms based on the network topology. Correlation-based data aggregation can be further categorized as DSC and CS based on whether the algorithm is dependent on a static correlation model. Fig. 2.1 summarizes the taxonomy.

2.2 Function-Based Data Aggregation

In some applications, the sink node is interested in statistical information rather than all the original data, which motivates the development of function-based data aggregation. Function-based data aggregation extracts and delivers statistical in-

formation of the original data vector via aggregation functions instead of delivering all the raw data to the sink node. Function-based data aggregation is essential in reducing the data volume for sensor network, and promoting energy efficiency of the system. Moreover, function-based aggregation is also one of the bases for scalability in large-scale services [21]. Further, distributed function-based data aggregation is the fundamental for the decentralized determination of meaningful global properties. In this section, related definitions of aggregation functions will be given. Then, based on network topology, the related papers are categorized as structured algorithms and unstructured algorithms.

First, a precise definition of aggregation function is given by Definition 2.1.

Definition 2.1. (*Aggregation Function*) [22]: An aggregation function f takes a multiset of elements from a set I and produces an output from a set O :

$$f : \mathbb{N}^I \rightarrow O.$$

A multiset is a generalization of set, in which elements are allowed to appear more than once. Therefore, aggregation function takes multiset as input means that: 1) a particular value may repeat several times; 2) the order in which the elements are aggregated is not important [22]. For aggregation functions such as MAX, MIN and SUM, both I and O are the same set. For aggregation functions such as COUNT, the output is an integer, regardless of the input set domain.

2.2.1 Classification of Aggregation Function

There are several perspective to categorize aggregation function. Table 2.1 gives some examples of aggregation function classifications. Specific definitions of each function type will be introduced.

Symmetric function	type-sensitive	AVERAGE, SUM
	type-threshold	MAX, MIN
Duplicate sensitiveness	duplicate-sensitive	AVERAGE, SUM
	duplicate-insensitive	MAX, MIN
Divisible		MAX/MIN, AVERAGE

Table 2.1: Summary of aggregation function types.

Symmetric Function

Symmetric function refers to the type of functions whose value remain unchanged when the values of arguments are permuted. Symmetric function covers most of the functions in applications. The formal definition of symmetric functions is given as follows.

Definition 2.2. (*Symmetric Function*) [23]: a class of functions which are invariant with respect to permutations of their arguments, for example, $f(x_1, \dots, x_n) = f(\pi(x_1, \dots, x_n))$, for all permutation π .

Symmetric functions only depend on sensor data, not its identity [24]. Distributed symmetric function computation is also called a counting problem [25]. Symmetric functions can be further classified into type-sensitive functions and type-threshold function. The definition of type-sensitive and type-threshold functions can be find as follows.

Definition 2.3. (*Type-Sensitive Function*) [23]: A symmetric function $f(\underline{x})$ is said to be type sensitive if there exists some γ with $0 < \gamma < 1$, and an integer $\bar{n}$, such that for $n > \bar{n}$, and any $j \leq n - \lceil \gamma n \rceil$, given any subset $\{x_1, x_2, \dots, x_j\}$, there are two subsets of values $\{y_{j+1}, y_{j+2}, \dots, y_n\}$ and $\{z_{j+1}, z_{j+2}, \dots, z_n\}$, such that

$$f(x_1, \dots, x_j, z_{j+1}, \dots, z_n) \neq f(x_1, \dots, x_j, y_{j+1}, \dots, y_n).$$

To determine the value of type-sensitive function, a certain fraction of arguments should be known. Some well-known examples of type-sensitive function are

AVERAGE, MODE and SUM. The MODE of a set of data values is the value that appears most often.

Definition 2.4. (*Type-Threshold Function*) [23]: A symmetric function $f(\underline{x})$ is said to be type-threshold function if there exists a non-negative threshold vector $\underline{\theta}$ of length $|\chi|$, such that $f(\underline{x}) = f'(\tau(\underline{x})) = f'((\min(\tau(\underline{x}), \underline{\theta})))$, for $\underline{x} \in \chi^n$, with $\min$ signifying element-wise minimum.

Some examples of type-threshold function include MAX/MIN, k th largest. Loosely speaking, the value of a type-threshold function is entirely determined by certain outstanding measurements. For this class of functions, an exponential speedup is possible [24].

In plain English, the aforementioned definitions state that, if we want to estimate the value of a type-sensitive function with certain fidelity, a minimum fraction of argument should be known. Whereas, the value of a type-threshold function can be determined by a fixed number of known arguments [23]. Generally, computing type-sensitive function is more complex than computing type-threshold function.

Duplicate Sensitiveness

Duplicate sensitiveness indicates whether the result of a specific function will be affected by duplicate inputs. The result of duplicate-insensitive aggregation functions will not be affected by duplicate readings from a single device. In contrast, the result of duplicate-sensitive aggregation function will change when a duplicate reading is reported. Duplicate sensitiveness is important since it implies restrictions on network properties and on certain optimization [26]. As for example, MAX and MIN are duplicate-insensitive while SUM, AVERAGE, MEDIAN, and MODE are duplicate-sensitive.

Divisible Function

Definition 2.5. (*Divisible Function [23]*): A function f is divisible if

- $r_n = |R(f, n)|$ is nondecreasing in n , where $|R(f, n)|$ is the function range;
- given any partition $\Pi(S) = S_1, S_2, \dots, S_p$ of $S \subset [n]$, there exists a function $g^{\Pi(S)}$, such that for any $\underline{x} \in \chi^n$

$$f(\underline{x}_S) = g^{\Pi(S)} \left(f(\underline{x}_{S_1}), f(\underline{x}_{S_2}), \dots, f(\underline{x}_{S_p}) \right),$$

where $\underline{x}_S = \{x_i\}$, $i \in S$.

If a function is divisible, then it can be computed in a divide-and-conquer fashion [23], which allows distributed computation.

2.2.2 Structured and Unstructured Function-Based Data Aggregation

According to network topology, function-based data aggregation algorithms can be categorized into two types: structured and unstructured. In structured algorithm, a hierarchical structure such as a tree topology or cluster based topology will be established, on which the data will be transmitted. In contrast, unstructured algorithms can operate independently without a specific topology [22]. The classification of function-based data aggregation can be summarized in Table 2.2.

Classification criteria	Features	References
structured function-based data aggregation	requires the establishment of hierarchical topology	[23, 25–28]
unstructured function-based data aggregation	works independently without establishing a topology	[29–31]

Table 2.2: The classification of function-based data aggregation.

In structured function-based data aggregation algorithm, a hierarchical topology will be established, then the data will be aggregated layer by layer, from the source to the sink node. Madden *et al.* [26] proposed a classical hierarchical function-based data aggregation algorithm, the Tiny AGgregation (TAG) service for ad-hoc sensor networks. TAG relies on the tree topology for computation of MAX, MIN, AVERAGE and SUM. Giridhar *et al.* [23] provided the information theoretic fundamentals on computing and communicating functions over sensor networks. They investigated the maximum rate at which functions of sensor measurements can be computed and communicated to the sink node. The authors tessellated the network into squared cells and designated the cell with the sink node as the root. A spanning tree was built on the cell graph. Based on this structure, they proved that the maximal rate to compute type-sensitive function in multi-hop networks is $O(\frac{1}{\log n})$ and the maximal rate to compute type-threshold function in multi-hop networks is $O(\frac{1}{\log \log n})$. Ying *et al.* considered distributed symmetric function computation in noisy multi-hop WSNs [25]. They aimed to minimize the total energy consumption used by the network while satisfying computation accuracy with high probability. In-network processing was exploited to reduce information transmission in the network with controlled redundancy to combat the noisy channel. A cluster based approach was proposed to compute divisible function in [27]. They designed a general protocol for divisible function which consists of intra-cluster protocol and inter-cluster protocol. In intra-cluster, all the nodes send data to cluster head, then cluster head broadcasts this received data to the whole cluster, each node in the cluster will compute a function value with the received broadcast value and then send it back to cluster head. Then, the following inter-cluster algorithm will aggregate the value through a tree topology to the sink node. Cao *et al.* [28] considered the computation of symmetric function and identity function over a sensor network where the density of the sensors varies with respect to their distance to the sink node. The problem of function

computation in large scale heterogeneous WSNs was studied. They proposed an energy-efficient algorithm based on in-homogeneous tessellation of the network. Corresponding optimal energy consumption scaling laws was derived.

Structured function-based algorithms perform exact function based aggregation with simple and energy-efficient approaches. Whereas, structured algorithms are vulnerable to node failure. A single point of node failure can significantly affect the function result. Furthermore, it's expensive to maintain a structured topology in dynamic environment.

Unstructured function-based algorithms work independently without establishing a topology. Generally, unstructured algorithms rely on more robust random access or gossip mechanism to communicate the aggregation function. Consequently, they are more suitable for dynamic environment. Sudeep *et al.* [29] considered in-network computation of MAX in structure-free random multi-hop WSNs. Structure-free network refers to a network in which nodes do not have an identity and hence do not know the topology. A schedule for transmission cannot be defined. Random access protocol was exploited. They showed that when pipeline MAX protocol is used in a minimum structure setting, the rate of computing the MAX is $\Omega\left(\frac{1}{(\log n)^2}\right)$. A decentralized algorithm based on gossip mechanism to compute statistical information of the global attribute was proposed in [30]. Gossip mechanism provides scalable and robust information dissemination with a low overhead [31].

In a nutshell, function-based data aggregation is suitable for applications where only a certain function value is important. It significantly eliminates redundant transmissions by only delivering a critical function value to the sink node. Several function types are defined to characterize complexity of different functions. Type-sensitive functions are harder to compute than type-threshold functions. Divisible functions can be computed in the distributed manner, which results in better scalability. We have reviewed literature regarding structured and un-

structured function-based data aggregation algorithms. Although highly energy-efficient, function-based data aggregation sacrifices much information of the whole measured data vector. Consequently, its application is limited. In the following section, correlation-based data aggregation which aims to recover the whole data vector will be introduced.

2.3 Correlation-Based Data Aggregation

Function-based data aggregation can significantly reduce energy consumption. However, it only provides limited information, while the rest of the information is lost. In the following, we will review literature regarding correlation-based data aggregation which exploits the data correlation to compress and reduce the number of data transmissions while guaranteeing the recovery of original data with certain fidelity at the sink node. There are two types of algorithms. The first relies on priori knowledge of the data correlation model, and cannot adapt to changing correlation models, such as DSC. The second, on the other hand, can perform well regardless of the correlation model, such as CS based algorithms. In the following, we will review both DSC and CS.

2.3.1 Distributed Source Coding

DSC is a coding method based on Slepian-Wolf theorem [32]. DSC compresses multiple sensor outputs in a distributed manner using only statistical information of the data at the sensor without inter-sensor communication. Generally, DSC algorithms consists of two parts: the tracking of data correlation structure and code construction [33]. The sink node or cluster head will continuously track the amount of correlation between the sensor nodes. Then, based on the correlation structure, a codebook will be constructed for the sensor nodes to encode its own data such

that the decoder at the sink node can decode all the information. Several encoding methods have been proposed in the literature [34, 35]. DSC brings significant energy conservation for typical sensor data whose correlation structure is static or changes slowly. However, the decoding of a sensor's information may depend on the decoding of other sensor's data. Hence, the loss of a single sensor reading may cause negative effect on the decoding of other sensors' readings. In other words, DSC is not robust to packet loss. Moreover, DSC requires prior knowledge of the data correlation structure, which is hard to obtain in practical scenario.

In brief, DSC exploits data correlation to performs non-collaborative data compression at the source nodes. DSC technique shifts the computation burden from the sensor nodes to the sink node and avoids data exchanges between sensor nodes. Nonetheless, DSC technique relies heavily on the prior information about the data correlation structure. Hence, it's not robust to changing correlation structure. Besides, it's sensitive to anomaly sensor data. For example, the data in DSC is predefined as main or side information. Therefore, the loss of main information significantly degrades the decoder performance.

In the following section, another correlation-based data aggregation, CS technique, will be introduced. CS technique has similar advantages as DSC such as simple encoding process and little inter-node data exchange. Moreover, CS has further merits. For instance, CS data reconstruction is insensitive to packet loss, and CS is robust to noise.

2.3.2 CS Based Data Aggregation

CS technique has attracted increasing research attention in many application fields, including data gathering WSNs, image processing, cognitive radio communications, etc. This section focuses on the application of CS into WSNs. In real life, various types of sensory data exhibit sparsity in different domains such as time, frequency,

and space, which inspires the application of CS into WSNs to reduce energy consumption and extend network lifetime. This section provides a detailed review of the state-of-the-art research works on the application of CS into WSNs. To begin with, the basic principles of CS and the features of CS techniques are outlined. Followed by an introduction of the classification metrics. Subsequently, we review the existing work in each category, highlighting the benefits and shortcomings.

CS Basics

Compressive sensing, also referred to as compressed sensing, is a promising technique that has attracted considerable research attention in the field of signal processing, computer science, and communication engineering. The field of CS was developed from a series of landmark papers by Candes, Romberg [36, 37] and Donoho [38]. They have shown that a signal with a sparse representation in a transform domain can be recovered exactly from a small set of linear, non-adaptive measurements. It's not necessary for the data vector to be sparse itself but can be compressible with sparse representations in some known transform domain [39]. For example, smooth data vectors are sparse in the Fourier basis while piecewise smooth data vectors are sparse in the wavelet basis [37].

More specifically, consider a data vector $\mathbf{u} \in \mathbb{R}^n$. $\mathbf{u}$ is sparse under orthonormal transform basis Ψ , which means that $\mathbf{u}$ can be represented by a k -sparse vector $\boldsymbol{\theta}$, such that

$$\mathbf{u} = \Psi\boldsymbol{\theta}, \tag{2.1}$$

where $\boldsymbol{\theta}$ is a k -sparse coefficient vector, only k out of n elements of $\boldsymbol{\theta}$ are non-zero, and $k \ll n$. Fourier transform basis, wavelet transform basis and discrete cosine transform (DCT) basis can be used as Ψ to obtain the sparse representation for a specific data vector.

Then, m random linear projections of the data vector are collected through measurement matrix Φ , obtaining

$$\mathbf{y} = \Phi \mathbf{u} = \Phi \Psi \boldsymbol{\theta} = \mathbf{A} \boldsymbol{\theta}, \quad (2.2)$$

where Φ is an $m \times n$ measurement matrix, $\mathbf{A} = \Phi \Psi$ is an $m \times n$ equivalent sensing matrix.

As for data reconstruction, the sink node performs the ℓ_1 -norm minimization as shown below in (6.5) to compute the sparsest $\boldsymbol{\theta}$.

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_{\ell_1} \quad \text{subject to} \quad \mathbf{y} = \mathbf{A} \boldsymbol{\theta}, \quad (2.3)$$

where $\|\boldsymbol{\theta}\|_{\ell_1} = \sum_i |\theta_i|$. Let $\hat{\boldsymbol{\theta}}$ be the optimal solution for the above optimization problem. The reconstructed data vector is given by $\hat{\mathbf{u}} = \Psi \hat{\boldsymbol{\theta}}$. There are plenty of recovery algorithms available in the literature. For example, linear programming [40] can be used to solve (6.5). In addition, basis pursuit (BP) techniques [36, 41] can also be used. Moreover, greedy algorithms such as matching pursuit (MP) can also be used to solve (6.5) iteratively [42, 43]. Fig. 2.2 demonstrates the process of CS measurement.

Recovery Conditions

The equivalent sensing matrix $\mathbf{A}$ plays a key role in data compression and recovery. The equivalent sensing matrices should satisfy some conditions in order to guarantee the recovery of the data vector. For example, the restricted isometry property (RIP) condition and coherence are two common criteria to guarantee the data recovery.

Figure 2.2: CS measurement operation.

The RIP Condition

The first criterion to evaluate the quality of sensing matrices is the RIP condition introduced in [41]. The RIP is a sufficient condition on $\mathbf{A}$ for exact recovery of a s -sparse $\boldsymbol{\theta}$.

Definition 2.6. *An $m \times n$ matrix $\mathbf{A}$ is said to satisfy RIP of order k , $k < n$, if there is an isometry constant $0 < \delta_k < 1$ such that*

$$(1 - \delta_k) \|\boldsymbol{\theta}\|_{\ell_2}^2 \leq \|\mathbf{A}\boldsymbol{\theta}\|_{\ell_2}^2 \leq (1 + \delta_k) \|\boldsymbol{\theta}\|_{\ell_2}^2, \quad (2.4)$$

holds for all k -sparse vectors, where δ_k is the smallest number satisfying (2.4).

If $\mathbf{A}$ satisfies the RIP of order $2k$, then $\mathbf{A}$ approximately preserves the distance between any pair of k -sparse vectors [37, 44]. This encouraging fact guarantees the existence of robust and efficient algorithms for CS recovery [37].

It has been shown that, with a fixed basis $\boldsymbol{\Psi}$ and a dense random measurement matrix $\boldsymbol{\Phi}$, matrix $\mathbf{A}$ satisfies RIP with high probability [37]. Nonetheless, checking RIP condition is not computationally tractable [45, 46].

Coherence

The coherence of the sensing matrix $\mathbf{A}$ is an alternative property to guarantee CS recovery, which is computationally tractable. The coherence of matrix $\mathbf{A}$ can be defined as follows,

Definition 2.7. *The coherence of a matrix $\mathbf{A}$, $\mu(\mathbf{A})$, is the largest absolute inner product between any two columns $\mathbf{a}_i$, $\mathbf{a}_j$ of $\mathbf{A}$:*

$$\mu(\mathbf{A}) = \max_{1 \leq i, j \leq n, i \neq j} \frac{|\mathbf{a}_i^T \mathbf{a}_j|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2}. \quad (2.5)$$

The coherence of $\mathbf{A}$ is easy to compute and provides a convenient way to evaluate the quality of the sensing matrix [46]. For instance, Donoho *et al.* [47] have shown that a k -sparse data vector can be exactly recovered as long as $\mu(\mathbf{A}) < \frac{1}{2k-1}$ in the noiseless case.

2.3.3 Applying CS into WSNs: the State-of-the-Art

There has been extensive research effort to apply CS technique into data gathering WSNs [48–53]. And the papers are quite diverse in their application scenarios. In the following, a taxonomy of the existing research papers on CS based data aggregation are proposed. The papers are classified based on two criteria: the measurement matrices and the network topology. As we introduced above, the choice of measurement matrix affects the recovery performance at the receiver. Hence, different algorithms are designed to guarantee data recovery with the various measurement matrices. The measurement matrix can be categorized as random matrix and deterministic matrix. The random measurement matrix can be further divided into dense random matrix and sparse random matrix. Network topology depends on the way the sensor nodes communicate to the sink node. If the sensor nodes communicate directly to the sink node, then it's a single-hop network, otherwise, it's a multi-hop network.

Table 2.3 gives an overview of the classification of CS based data aggregation algorithms.

Classification criteria		Reference
measurement matrix	random matrix	[48, 49, 51, 53–57]
	deterministic matrix	[58, 59]
network topology	single-hop network	[46, 60, 61]
	multi-hop network	[62, 63]

Table 2.3: Classification of CS based data aggregation

For a dense random measurement matrix, all the elements are generated ran-

domly from certain probability distributions and are nonzero. Plenty of research works exploit dense random matrices as the measurement matrix. For example, Luo *et al.* [48] presented the first complete design to apply CS theory to sensor data gathering for large scale network. The authors proposed a framework for compressive data gathering with the objective of reducing global data transmissions and extending network lifetime. Instead of transmitting original data, each sensor node multiplies its reading with corresponding column vector in the measurement matrix and sends the product to next hop. Each node contributes to weighted sum by adding its own product. Finally, the sink receives several weighted sums of all the readings and recovers all the original readings via ℓ_1 -minimization. This work demonstrated the load balancing feature of CS technique and analyzed the network capacity of CS based data gathering network. This way of directly applying CS is also referred to as plain CS. Xiang *et al.* [49, 54] proved in their works that plain CS may not be beneficial, since it forces every link to carry the same traffic load, resulting in unnecessarily high traffic for early stage transmission. To address this issue, they proposed hybrid CS aggregation to minimize total energy consumption for data gathering. CS coding starts only when the outgoing traffic exceeds a predefined threshold. Hybrid CS reduces the overall energy consumption.

For sparse random matrix, a certain portion of entries are zero, while other entries are generated randomly. Sparse random matrices have the potential to further reduce data transmissions in the network. Whereas, the trade-off between sparsity of the measurement matrix and the number of linear random projections should be carefully considered in order to guarantee the recovery requirements [55]. In the paper [55], Wang *et al.* proposed a distributed algorithm based on sparse random projections to enable energy-efficient data gathering with refinable recovery error. Each sensor node stores a random projection, and the sink node determines the number of sensors to query based on the desired approximation error. The authors showed that $O(\text{poly}(k, \log n))$ sparse random projections are sufficient to recover

a data approximation which is comparable to the optimal k -term approximation, with high probability. The average non-zeros in each random projection vector is $O(\log n)$. Here, n is the number of nodes in the network, k is the signal sparsity. In [55], the entries of sparse random measurement matrix are generated from i.i.d probability distribution and low complexity sketching decoder was used to recover the data. Several extension works have been done to adapt sparse random projection method to different application scenarios. Ebrahimi *et al.* [51] considered compressive data gathering by exploiting sparse random projection. The nonzero coefficients are distributed evenly in the measurement matrix such that none of the columns has all-zero entries. Multiple projection trees were constructed and SINR-based scheduling method was proposed to reduce data gathering latency and total number of transmissions. The authors implicitly assumed that all links have the same unit energy cost. Another set of papers extend the uniform sparse random measurement matrix to nonuniform random matrix [53, 56, 57]. Shen *et al.* [56] proposed a nonuniform compressive sensing (NCS) method for a heterogeneous data gathering WSN. Due to the heterogeneous resource supply of the sensor nodes, the sparse random measurement matrix becomes nonuniform. That is, the entries of the sparse random measurement matrix follow nonuniform probability distribution. The authors also designed a distributed implementation of NCS and showed that the proposed NCS has about the same reconstruction accuracy as traditional CS while requiring less samples, thus is more energy-efficient. Liu *et al.* [53] proposed a novel compressive data collection scheme for WSNs. The data collection process was modeled as a nonuniform sparse random projection (NSRP). An NSRP estimator that guarantees optimal error bound was proposed.

Although random matrix serves as a good candidate as CS measurement matrix, there is no efficient algorithm to test whether a random matrix satisfy RIP. Even though statistically speaking, it satisfies RIP with overwhelming probability. In contrast, deterministic matrices can guarantee stable recovery without proba-

bilistic arguments [58, 59]. Moreover, deterministic matrices require less storage space compared to random matrices. Amini *et al.* [58] constructed deterministic matrices with small coherence using p -ary block codes. These matrices provide more flexibility in the choice of number of rows. Ravelomanantsoa *et al.* [64] proposed a simple deterministic measurement matrix to facilitate the hardware implementation. A thresholding approach was proposed in the DCT domain. The research on deterministic CS measurement design is still at its early stage.

CS technique can be exploited in various network scenarios to achieve performance improvements. Subsequently, the research works can be classified based on network topology in which the CS technique is applied. First, the following papers reported the use of CS technique for single-hop WSNs [46, 60, 61]. Yang *et al.* [60] addressed compressive data transmission problem for single-hop EH-WSNs. The sensor nodes were powered by intermittently available energy harvested from the environment. The authors provided theoretical guarantees on the number of measurements for reliable and computationally efficient recovery, by showing that the sensing matrix satisfies the RIP, under reasonable conditions. Chen *et al.* [46] proposed an optimized active node selection method for single-hop compressive sleeping WSNs. Both spatial correlation of the sensor nodes and temporal correlation were exploited. This method showed improvements on reconstruction accuracy, spectrum reuse and network lifetime. Similarly, a cost-aware activity scheduling for single-hop CS in WSNs was proposed in [61]. In this work, the authors considered compressive sleeping node where only a fraction of nodes will be activated to perform sensing tasks and send their data to the fusion center. Cost diversity are exploited and the trade-off between the cost and recovery error was investigated. CS technique is also exploited in multi-hop scenario. Xu *et al.* [62] proposed a hierarchical data aggregation scheme using CS technique. Different compression thresholds were set based on cluster size at different level of data aggregation tree to optimize the number of total transmissions. In [63], the authors considered a

two-hop network. They proposed a neighbor-aided compressive sensing (NACS) scheme for data gathering in spatial-temporal correlated WSNs. In this framework, the sensor node sends the raw readings within several consecutive time slots to a randomly selected neighbor. Then, the neighbor generates projections of the received data and its own readings, and directly sends them to the sink.

In short, CS is a promising technique to be introduced into sensor networks. It suggests a way to compress the data for energy-efficient communication without the need to adapt to the correlation structure. However, the complex interplay of CS and routing design, the nonuniform energy cost of sensor networks remain to be addressed to fully exploit the benefits of CS techniques.

2.4 Energy Harvesting Wireless Sensor Networks

Traditional WSNs are powered by portable, non-rechargeable batteries, which limits their network lifetime. Hence, energy conservation is the crucial concern in order to prolong network lifetime. Whereas, there are applications where the sensor networks are required to operate for decades while battery recharge or replace is impractical. Recently, the advances in EH techniques provide a possible alternative to power the sensor nodes, which may unlock the possibility of perpetual operation of the WSNs.

EH is to acquire renewable energy from ambient environment and convert to electricity to power the devices [65]. There are plenty of natural sources available for harvesting: solar energy, indoor lighting, RF signal, and vibration, each associates with different technologies [66, 67]. Besides, wireless power transfer in which energy is transferred from one node to another can also be another way of renewable energy input [68]. There are two types of EH architectures: *Harvest-Use* and *Harvest-Store-Use* [66]. A *Harvest-Use* architecture is suitable for applications that require only sporadic supply of energy. For instance, piezo-electric material

embedded in shoes can generate energy to transmit information for the track of the shoe-wearer [69]. The *Harvest-Store-Use* architecture consists of a storage component, which can store the extra harvested energy for later use. For example, solar energy is more abundant during daytime, hence, it can be harvested and stored to power the system over the night.

The idea of EH is not new [70]. Nevertheless, new challenges arise to harvest energy for small size devices such as the sensor nodes. For example, the energy storage system for renewable energy in sensor nodes is comparatively small, which leads to sporadic energy supply. The harvested energy should be wisely managed such that information can be successfully delivered while perpetual operation of WSNs can be achieved.

With renewable energy sources, network lifetime is no longer a critical concern since the ultimate goal of EH-WSN is self-sustainability and virtually perpetual operation. Consequently, the design focus has been shifted from energy conservation to intelligent management of the harvested energy to achieve certain performance requirements with sustainable operation. Generally, two performance requirements are investigated in the literature: the throughput and the latency. In [71], Orhan *et al.* investigated the problem of throughput maximization in a two-hop half-duplex relay network with EH capability. The authors derived optimal transmission policies under the assumption of known energy arrivals at the source and the relays. They considered two scenarios: one relay and two parallel relays. The problem was formulated as an convex optimization problem and solved efficiently. Other papers focusing on efficient circuit design [72], energy storage using super-capacitors [73], and optimal MAC protocol to improve throughput in terms of bits per second [74] can also be found in the literature.

To sum up, the advances of EH techniques have provided an alternative way to power the sensor nodes, which may boost the perpetual operation of the sensor nodes. Whereas, given the limited size of sensor nodes, and unstable characteristics

of ambient power source, it's challenging to design proper transmission algorithms to achieve system requirements while maintaining sustainable operation.

2.5 Concluding Remarks

The data deluge brought by the pervasive sensors have posed a great challenge to traditional energy-constrained WSNs. In this chapter, possible solutions from various perspectives are reviewed and discussed. First, data aggregation algorithms which aim to reduce data traffic by combining information from multiple sensors are discussed. Next, EH can be used to power the sensor nodes for sustainable operation while satisfying system requirements is also reviewed. Given the extensive works in all these fields, some research problems remain relatively unexplored. For instance, how to achieve the trade-off between energy consumption and application requirements such as latency and accuracy for data aggregation, how to adapt CS into sensor networks, and how to design proper transmission algorithms to achieve system requirements while maintaining sustainable operation. The following chapters will focus on solving these problems.

Chapter 3

Function-Based Data Aggregation

As discussed in Chapter 2, transmitting all the raw data sensed by the source nodes to the sink node is not an efficient way. Since in some applications, only a function value is needed for further decision making, it's not necessary to transmit all the data to the sink node. Consequently, function-based data aggregation can be applied to significantly reduce data transmissions and energy consumption. In this chapter, a function-based data aggregation algorithm is proposed for a single-hop network to reduce energy consumption as well as data gathering latency. More specifically, a priority-aware hybrid scheduling scheme is proposed to achieve fast, decentralized, and energy-efficient MAX function computation in a single-hop network. Here, fast means that the proposed scheme has low average computation delay. In our proposed scheme, priority-aware TDMA scheme is used to first schedule the transmission. The node with the maximum value is given the priority to transmit. Whereas, multiple concurrent transmissions will result in collisions, which may degrade the transmission success probability for pure TDMA-based scheme. Therefore, in order to improve the performance with the influence of collision, CDMA is introduced to recover information whenever collision occurs. Our contribution consists of two parts. First, the expression for computation failure probability is derived and the optimal value is found using iterative method. Further, an optimization problem is formulated to minimize average computation

time subjected to the constraints on failure probability. The synergy of CDMA and priority-aware TDMA contributes to reduction of computation failure probability as well as average computation delay. Moreover, the proposed scheme also significantly reduces energy consumption compared to the traditional round-robin TDMA transmission, since the scheduling rule in the proposed scheme helps to reduce number of transmissions for a single computation. Numerical results show that our algorithm reduces average computation time and energy consumption for one-shot MAX function computation, with the advantage of better scalability.

3.1 Introduction

In monitoring applications, energy efficiency is the primary consideration for WSNs. The reason is that the sensor nodes are often battery-driven and are deployed densely, which makes it difficult to recharge or replace the battery. In addition, timely and reliable data transmission is of great importance. Moreover, the massive deployment of sensor nodes requires that the algorithms designed for WSNs should have good scalability.

Traditionally, in a monitoring system, source nodes will sense the environment and send all the raw data without any processing in the intermediate nodes towards the sink node for further decision and control. However, downloading all the raw data appears to be an inefficient way, since the sink node is more interested in certain aggregate functions (MAX, AVERAGE, MEDIAN) of the distributed sensor data to make further control decision. Data transmission volume can be reduced by transmitting only the key information rather than all the raw data. It has been shown that communication cost is much higher than processing cost [75]. Hence, it is worthwhile to develop an algorithm that is relatively complex in processing but reduces the number of transmissions. One of the ways to reduce the number of data transmission is to develop novel function-based data aggregation

algorithms [76–79]. Sensor nodes can perform pre-processing algorithms to reduce the number of transmissions while guaranteeing the delivery of function value.

As introduced in Chapter 2, there are various aggregation functions such as MAX/MIN, AVERAGE, MEDIAN, each corresponds to different properties in applications. MAX function is one of the most popular aggregation function. It can be categorized as type-threshold function [77, 78, 80]. The value of a type-threshold function is entirely determined by certain outstanding measurements. Moreover, MAX function is also a divisible function, which allows decentralized computation. In many applications, timely, decentralized, and reliable MAX function computation is important. For example, in a fire alarm system, an alarm will be triggered when the maximum value of the temperature exceeds certain threshold [81]. Hence, the focus of this chapter is to provide a fast, decentralized and energy-efficient algorithm to compute the MAX function.

Previous works on function-based data aggregation focused on deriving information-theoretic results based on the assumption of centralized control [28, 29, 82]. For instance, Khude *et al.* [82] aimed to establish scaling laws for the time and energy complexity of the distributed function computation over random wireless networks, under the assumption of centralized contention-free scheduling of packet transmissions. Cao *et al.* [28] considered the computation of symmetric function over a sensor network where the density of the sensors varies with respect to their distance to the sink node. They proposed an energy-efficient algorithms based on in-homogeneous tessellation of the network. Kamath *et al.* [29] investigated the rate of computing MAX function in a structure-free network using random access MAC. Centralized control is relatively simple in terms of analysis, and it provides the upper bound performance of an algorithm. Nonetheless, centralized control may introduce significant control overhead, which makes it hard to implement in practice. On the other hand, decentralized computation algorithms rely on only local information to make decisions. Decentralized algorithms scale well and are

easier to implement although optimal performance is not always guaranteed. With proper design, decentralized algorithms can also achieve satisfactory performance. For example, Goldenbaum *et al.* [79] proposed a decentralized analog computation scheme to compute the AVERAGE function in a single-hop network. The analog computation scheme exploited the match between the function structure and the mathematical structure of the channel to estimate the function value with provable error bound. However, this analog scheme cannot be used to compute the MAX function.

This chapter presents a decentralized, fast, and energy-efficient MAX function computation algorithm in a single-hop network through hybrid scheduling scheme. A priority-aware TDMA scheduling is adopted, in which the node with the maximum value has the priority to transmit. Then CDMA scheme is introduced on top of the priority-aware TDMA scheduling scheme to further reduce computation time and computation failure probability. The more practical physical interference model is assumed. More specifically, the main contributions of this chapter can be given as follows.

- A hybrid scheduling algorithm is proposed to achieve scalable MAX function computation in a single-hop network.
- The expressions of computation failure probability is derived. An optimization problem to minimize computation failure probability is formulated and solved by iterative method.
- Sequentially, the expression of average computation time is derived. An optimization problem to minimize average computation time subjected to constraints on computation failure probability is formulated and solved.
- Numerical results show that our proposed scheme significantly reduces energy consumption and computation latency, and has better scalability compared with the benchmark algorithm.

Figure 3.1: A single-hop network. The dashed circle with a node in its center is the communication area of the node. The radius of the dashed circle denotes the communication range. d_1, d_2 are the distance from sensor node 1 and 2 to the sink node, respectively. $d_1, d_2 \leq r$.

The rest of this chapter is organized as follows. The network model is described in Section 3.2. Section 3.3 gives the hybrid scheduling assignment. In Section 3.4, the expression for computation failure probability is derived, and iterative method is used to solve for optimal scheduling assignment to minimize failure probability. In Section 3.5, the expression of average computation time is derived and the optimization problem to minimize average computation time subjected to the constraints on computation error probability is formulated. The solution is provided via an algorithm to the optimization problem. Numerical results are shown in Section 3.6. Finally, Section 3.7 concludes this chapter.

3.2 Network Model

Our system consists of one sink node and K sensor nodes. Assume that the K channels from sensor nodes to the sink node are independent and identically distributed (i.i.d.). Any node that is interested in the MAX value can be the sink

node. Assume a single-hop network where the transmission of one node can be received by all the other nodes [23]. Single-hop network models the situation where the sink node lies in the communication range of the transmitter, as shown in Fig.3.1. For example, the sensor's communication range is r , and if the sensor is deployed in a room, the distance between the sensor and the sink node is d , which satisfies $d \leq r$. In this case, the sensor network can be modeled as a single-hop network. This chapter discusses the problem with the assumption of a single-hop network. However, our algorithm can be used in multi-hop network with clustering techniques. For instance, when the number of sensor nodes grows, the sensor nodes can form clusters with some of the famous clustering techniques [18, 83, 84] and elect powerful cluster heads to communicate with the sink node. Then within each cluster, our algorithm can still apply. In order to keep our discussion more compact, clustering technique is not discussed in this chapter.

At a specific time t , each node k ($k = 1, \dots, K$) takes a measurement u_k of the environment (e.g. temperature, humidity), which is only known to itself. The measurements are assumed to be i.i.d. across all the nodes in the network. Let us assume that u_k is uniformly distributed over $[0, 1]$. It is worth noting that our proposed scheme works for all real-valued measurements with arbitrary probability distribution function. This assumption is justified by the fact that any random variables with a continuous cumulative distribution function (CDF) can be transformed to another random variable that is uniformly distributed over $[0, 1]$, which is also assumed in [85]. Thus, to simplify our analysis, in the following sections, we follow this assumption. The sensor readings at time t for the whole network can be represented as a vector $\mathbf{u} = [u_1, \dots, u_K]$. All the nodes are assumed to be synchronized to facilitate TDMA schedule. Sensor nodes are equipped with CDMA encoder and decoder. Transmission time is slotted and the length of one time slot is denoted as Δ . Assume that the sensor readings will be stable for a period of time denoted as $T_0 = \Gamma + D$, where $\Gamma = N\Delta$, $N \in \mathbb{Z}_+$ controls the total amount

of time slots that can be used to compute the maximum function, $\mathbb{Z}_+$ denotes the set of non-negative integers ; D is the transmission time. The purpose of our algorithm is to make the maximum value $u_m = \max_k \{u_k\}$ among the sensor readings available at the sink node within time T_0 . In the following sections, the node with the maximum value is denoted as the best node.

3.2.1 Interference Model

There are mainly two types of interference model: the protocol model and the physical model. Some research works considered the problem of computation under protocol model [28,76], which assumed that information cannot be recovered if more than one node transmit concurrently within the transmission range. Protocol model can simplify the analysis. However, physical model can better characterize the features of wireless medium, and therefore is considered as a more realistic model [86,87]. This chapter adopts the physical interference model in the algorithm design. Following physical interference model, the sink node can successfully receive the transmission from node k only when the signal-to-interference-plus-noise ratio (SINR) at the sink node is above a threshold, which can be expressed as

$$\text{SINR}(k) = \frac{P_{\text{rx}}(k)}{\sum_{l \neq k} P_{\text{rx}}(l) + \sigma^2} \geq \gamma^*, \quad (3.1)$$

where $P_{\text{tx}}(k)$ is the receive power of node k , $\sum_{l \neq k} P_{\text{rx}}(j)$ is the interference power from concurrent transmissions, σ^2 is the background noise, γ^* is the SINR threshold.

3.3 Hybrid Scheduling Assignment

In a single-hop network, the objective here is to find a decentralized transmission schedule to facilitate fast and energy-efficient MAX function computation at the sink node. More specifically, the problem can be formulated as to come up with

a 3-tuple assignment $A = (T, C, P)$ for the single-hop network so that the MAX function can be efficiently computed at the sink node with less delay overhead. Here, T is the time slot assignment vector; C is a $K \times L$ matrix that describes the CDMA code sequence for the network, L is the length of CDMA sequence, P is the power assignment vector. That is, for a node k , a schedule assignment $A_k = (T_k, C_k, P_k)$ will determine that node k transmits at time slot T_k , using CDMA code sequence C_k and with transmit power P_k . In the following, the time, code, and power allocation rules will be introduced respectively.

3.3.1 Priority-Aware Time Slot Allocation

Traditional TDMA assigns each node a non-overlapping predefined slot to transmit. Compared to other contention based MAC schemes, TDMA can eliminate the hidden terminal problem and avoid collision without extra overhead. Nevertheless, the non-overlapping time slot assignment for each node is inefficient when not all the nodes have to transmit in one computation round. For instance, round-robin transmission scheme is widely used for collision-free scheduling, in which nodes take turns to transmit. However, it incurs large delay in the case of MAX function computation, since the sink node needs to wait for the transmissions of all nodes before computing the maximum.

In order to be fast and energy-efficient, the proposed TDMA slot assignment $T = [T_1, \dots, T_K]$ is designed to be priority-aware. Since the aim is to compute the MAX function, priority is given based on sensor reading. Sensors with larger values have higher priority level in transmission [85]. Formally, the priority-aware TDMA can be characterized by a mapping function $f : u_k \rightarrow T_k$. The domain of u_k is divided into $N + 1$ sections with various lengths. The length of section i can be represented as $\alpha_N[i]$. For notation simplicity, define $I(i) = [1 - \sum_{j=0}^i \alpha_N[j], 1 - \sum_{j=0}^{i-1} \alpha_N[j])$ to denote the i th section. For example, $I(0) = [1 - \alpha_N[0], 1]$ represents the first interval. With the above notation, the

Figure 3.2: Time slot allocation illustration.

time slot allocation can be expressed as the following,

$$T_k = f(u_k) = i\Delta \quad u_k \in I(i), i = 1, \dots, N. \quad (3.2)$$

Fig. 3.2 shows the principles of time slot allocation. It can be observed that if $u_k \in [1 - \alpha_N[0], 1]$, node k will set its timer as 0, it will transmit immediately after it obtain its measurement. If $u_k \in [1 - \sum_{j=0}^i \alpha_N[j], 1 - \sum_{j=0}^{i-1} \alpha_N[j])$, node k will set its timer as $i\Delta$, sensor reading will be transmitted after time duration $i\Delta$. Intuitively, this measurement to timer mapping scheme indicates that the larger the sensor reading, the smaller the timer value, which guarantees that the node with the maximum value will transmit first.

The duration length of the sections can be adjusted to minimize the failure probability and computation delay.

3.3.2 CDMA Code Sequence Allocation

CDMA assigns a unique orthogonal code sequence to each node so that different nodes can transmit simultaneously without interfering with each other. To use CDMA, the sensor node should be equipped with an encoder and decoder.

In order to mitigate the influence of collision in time domain, each node k is also allocated with a CDMA code sequence C_k of length L . A code with length L has a spreading gain $G = 2^L$. It can be observed in the following SINR analysis

that the use of CDMA can reduce the interference power by a factor $\frac{1}{G}$.

3.3.3 Transmission Power Allocation

For our assumption, power allocation is done so that each node can achieve uniform receive power P_{rx} . Assume that the transmitter knows the channel state information. Therefore, transmission power for node k will be $P_k = \frac{P_{rx}}{h_k}$, in which h_k is the channel power gain from node k to the sink node.

3.3.4 Decentralized Operation of Proposed Algorithm

In our proposed algorithm, a node k is given an assignment $A_k = (T_k, C_k, P_k)$. This assignment aims to give a priority-aware schedule that can make the maximum value available at the sink node with minimum average computation time. The operation of the network can be summarized as follows.

Step 1. Each sensor node will first sense the environment and produce a sensor reading u , as shown in Fig. 3.3a.

Step 2. After obtaining the sensor reading, an assignment (T_k, C_k, P_k) is generated locally based on the sensor reading, which determines the time, code sequence and power for node k to transmit, as shown in Fig. 3.3b.

Step 3. According to the assignment in Step 2, at time slot T_k , node k will transmit at time slot T_k with CDMA code C_k and transmission power P_k , as shown in Fig. 3.3c.

To save the overall energy consumption, once the sink node successfully decodes the maximum value, it will broadcast a message to the rest of the network so that other nodes will not transmit.

(a) Sense the environment, produce sensor reading.

(b) Hybrid scheduling assignment.

(c) Data transmission.

Figure 3.3: Operation of hybrid scheduling-based maximum function computation.

3.4 Computation Failure Probability

Computation failure refers to the situation when the maximum value cannot be successfully recovered at the sink node. Sensor nodes whose timers expire at the same time as the best node will interfere with the best node. To guarantee the computation of the MAX function, a certain target SINR γ^* should be maintained. Computation failure probability is related to the ability of an algorithm to deal with interference. In the following, the computation failure probability will be derived with the schedule assignment $A = (T, C, P)$.

Let us assume that the best node transmits in slot i . Then, the computation failure occurs when the received SINR is lower than a threshold γ^* , i.e.,

$$P_e = P(\text{SINR}_{(i)} < \gamma^*). \quad (3.3)$$

The SINR for the nodes in i th timer interval can be expressed as

$$\text{SINR}_{(i)} = \frac{P_{\text{rx}}}{\frac{P_{\text{rx}}}{G}(M_i - 1) + \sigma^2}, \forall i = 0, 1, \dots, N, \quad (3.4)$$

where M_i is the number of nodes in interval i , P_{rx} is the receive power, σ^2 is white Gaussian noise power.

According to physical model, if the best node lies in interval i , the target SINR γ^* should be met in order to decode the maximum value from M_i concurrent transmissions. Simultaneous transmitters M_i in interval i should be controlled in order to meet target SINR. From $\text{SINR}_{(i)} \geq \gamma^*$, the following result regarding the number of concurrent transmissions in one interval can be derived,

$$M_i \leq 1 + \frac{G}{\gamma^*} - \frac{G\sigma^2}{P_{\text{rx}}}. \quad (3.5)$$

We use $M^* = 1 + \lceil \frac{G}{\gamma^*} - \frac{G\sigma^2}{P_{\text{rx}}} \rceil$ to denote the upper limit of simultaneous transmitters in one interval, where $\lceil x \rceil$ is the ceiling function that gives the smallest integer that is greater or equal to x .

CDMA code is used to distinguish the maximum value when there are more than one node transmitting at the same time. The success probability for CDMA to decode the maximum value from M_i transmitters is denoted as $P_{\text{CDMA}(i)}$, whose value is evaluated as follows,

$$P_{\text{CDMA}(i)} = \begin{cases} 1 & M_i \leq M^* \\ 0 & M_i > M^* \end{cases}. \quad (3.6)$$

Equation (3.6) shows that when the number of concurrent transmitters in one interval exceed M^* , the maximum value cannot be decoded. Therefore, the failure probability in interval i can be obtained as,

$$\begin{aligned} P_{ei} &= P(M_i > M^*) \\ &= \sum_{M_i=M^*}^K \binom{K}{M_i} (\alpha_N[i])^{M_i} (1 - \sum_{j=0}^i \alpha_N[j])^{K-M_i}, \end{aligned} \quad (3.7)$$

where $\binom{K}{M_i}$ denotes binomial coefficient.

There are $N+1$ intervals in total, hence, the failure probability of our proposed scheme is

$$\begin{aligned} P_e &= \sum_{i=0}^N P_{ei} \\ &= \sum_{i=0}^N \sum_{M_i=M^*}^K \binom{K}{M_i} (\alpha_N[i])^{M_i} (1 - \sum_{j=0}^i \alpha_N[j])^{K-M_i}. \end{aligned} \quad (3.8)$$

The first objective is to minimize the failure probability, which can be described by the following optimization formulation.

$$\begin{aligned} \min_{0 \leq \alpha_N[j] \leq 1} &: P_e \\ \text{s.t.} & \sum_{j=1}^N \alpha_N[j] \leq 1, \\ & 0 \leq \alpha_N[j] \leq 1, \quad j = 0, \dots, N. \end{aligned} \quad (3.9)$$

3.4.1 Using Iterative Method to Solve for P_e and α_N

It's not easy to prove the convexity of the problem. Hence, as to solve the problem, an iterative method is proposed to get a feasible solution. Rewrite P_e as follows,

$$P_e = P_{e0} + P_{e1} + \cdots P_{eN}, \quad (3.10)$$

where $P_{ei} = \sum_{M_i=M^*}^K \binom{K}{M_i} (\alpha_N[i])^{M_i} (1 - \sum_{j=0}^i \alpha_N[j])^{K-M_i}$.

$P_{e0} = \sum_{M_0=M^*}^K \binom{K}{M_0} (\alpha_N[0])^{M_0} (1 - \alpha_N[0])^{K-M_0}$ is a function of $\alpha_N[0]$, $P_{e1} = \sum_{M_1=M^*}^K \binom{K}{M_1} (\alpha_N[1])^{M_1} (1 - \alpha_N[0] - \alpha_N[1])^{K-M_1}$ is a function of $\alpha_N[0]$ and $\alpha_N[1]$. P_{ei} is a function of $\alpha_N[0], \alpha_N[1], \dots, \alpha_N[i]$, and so forth. For each P_{ei} , solve for their minimum value with regard to $\alpha_N[i]$, after obtaining the local optimal $\alpha_N[i]$, the value can be used to calculate $\alpha_N[i+1]$.

Starting from P_{e0} , solve $\frac{\partial P_{e0}}{\partial \alpha_N[0]} = 0$ for $\alpha_N[0]$ that minimizes P_{e0} . Then pick the roots that satisfy the constraints. The solution $\alpha_N[0]$ will be carried forward to the calculation of $\alpha_N[1]$. For P_{e1} , the value of $\alpha_N[0]$ that is obtained above is used. Solve $\frac{\partial P_{e1}}{\partial \alpha_N[1]} = 0$ for $\alpha_N[1]$, pick the roots that satisfy the constraints. $\alpha_N[1]$ can be obtained and used in the calculation of $\alpha_N[2]$. Similarly, for $2 \leq i \leq N$, by evaluating $\frac{\partial P_{e2}}{\partial \alpha_N[2]} = 0, \dots, \frac{\partial P_{ei}}{\partial \alpha_N[i]} = 0$, a set of feasible solutions $\alpha_N[0], \alpha_N[1], \dots, \alpha_N[N]$ can be obtained. This set of $\alpha_N[0], \alpha_N[1], \dots, \alpha_N[N]$ is a feasible solution to the problem, but not necessarily the optimal one.

The procedure of iterative method can be summarized in Algorithm 1.

Algorithm 1: Iterative method to solve for α_N and P_e

Require: $N \geq 0, K \geq 1$

Ensure: $\alpha_N[0], \alpha_N[1], \dots, \alpha_N[N], P_e$

1: **for** $0 \leq i \leq N$ **do**

2: $P_{ei} = \sum_{M_i=M^*}^K \binom{K}{M_i} (\alpha_N[i])^{M_i} (1 - \sum_{j=0}^i \alpha_N[j])^{K-M_i}$;

3: Solve $\frac{\partial P_{ei}}{\partial \alpha_N[i]} \leftarrow 0$ for $\alpha_N[i]$ that minimizes P_{ei} .

4: **end for**

5: $P_e \leftarrow P_{e0} + P_{e1} + \dots + P_{ei} + \dots + P_{eN}$;

3.4.2 Trade-off Between Computation Failure Probability and Computation Latency

Figure 3.4: Failure probability as a function of maximum waiting time Γ for different spreading gain.

Intuitively, if more time is allowed for computation, the probability of collision and failure can be reduced. There is a trade-off between the computation failure probability and the computation latency for the proposed algorithm. Fig. 3.4 plots the failure probability as a function of maximum waiting time Γ for different spreading gain when the network size is 10. As expected, it can be observed that when the maximum waiting time is increased, the failure probability decreases. Moreover, for a certain maximum waiting time, larger spreading gain results in smaller failure probability, which indicates that longer CDMA code can effectively improve the success probability when collision happens.

In the following section, an optimization problem is formulated to formally investigate the trade-off between failure probability and average computation time.

3.5 Average Computation Time

The second objective of applying function-based data aggregation is to reduce latency. In this section, the expression of average computation time is formally derived for our proposed scheme. The average computation time τ consists of two parts given by the following,

$$\tau = \tau_1 + D, \quad (3.11)$$

where τ_1 represents the average time that the node with the maximum value needs to wait before transmission, and D denotes transmission time, which is proportional to the spreading gain..

According to the mapping function in Section 3.3.1, the best node can be assigned to transmit in any time slot with certain probability based on the interval where its sensor value falls in. More precisely, denote the event that the best node transmit at time slot i as A_i . The probability of event A_i , denoted as $P(A_i)$, can be computed as the following,

$$P(A_i) = \sum_{m=1}^K \binom{K}{m} (\alpha_N[i])^m (1 - \sum_{j=0}^i \alpha_N[j])^{K-m}. \quad (3.12)$$

Therefore, we have

$$\tau_1 = \sum_{i=0}^N i \Delta P(A_i) \quad (3.13)$$

Then, the average computation time to finish one computation is denoted as,

$$\tau = \sum_{i=0}^N i \Delta P(A_i) + D \quad (3.14)$$

$$= \sum_{i=0}^N \sum_{m=1}^K i \Delta \binom{K}{m} (\alpha_N[i])^m (1 - \sum_{j=0}^i \alpha_N[j])^{K-m} + D. \quad (3.15)$$

The goal is to find a vector of α_N to minimize the average computation time.

Intuitively, if all the nodes to transmit at time 0, it would be the most time-saving scheme. However, using this naive scheme, computation failure probability increases as network size grows. Therefore, the scheme should be designed in a way that average computation time is minimized while maintaining constraint on computation failure probability. In order to guarantee the feasibility of the solution, for given timer level N and spreading gain G , a proper failure probability constraint η is needed, which should be less or equal to the optimal value.

The optimization problem can be formulated as,

$$\begin{aligned} \min_{\{\alpha_N\}} \quad & \tau = \sum_{i=0}^N \sum_{m=1}^K i \Delta \binom{K}{m} (\alpha_N[i])^m (1 - \sum_{j=0}^i \alpha_N[j])^{K-m} + D \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_N[i] \leq 1 \end{aligned} \quad (3.16)$$

$$0 \leq \alpha_N[i] \leq 1, \quad i = 0, 1, \dots, N \quad (3.17)$$

$$P_e \leq \eta. \quad (3.18)$$

3.5.1 Using Interior Barrier Function Method to Solve for Optimal α_N and τ

Given different N and K , the convexity of the objective function and the nonlinear constraints can vary. Therefore, this problem is solved for each fixed N and K using interior barrier function method [88] for general constrained nonlinear optimization problem. This method uses barrier functions to prevent the current solution from stepping out of the feasible region. A barrier function is continuous and nonnegative over the interior of the constraints, and approaches ∞ as the boundary is approached from the interior. For given N and K , the optimization problem (3.16) is transformed into an unconstrained one. First, an auxiliary function $f_u(\alpha_N, \mu) = \tau + \mu B(\alpha_N)$ is defined, where μ is a small positive number, and $B(\alpha_N)$ is the barrier function. The constrained optimization problem can be

transformed into an unconstrained problem as the following,

$$\min_{\{\alpha_N, \mu\}} f_u \quad (3.19)$$

Then, the necessary condition for optimal solution, Karush-Kuhn-Tucker (KKT) condition, is used,

$$\nabla f_u = 0. \quad (3.20)$$

This will gives us the solution as a function of μ , $\alpha_N = s(\mu)$. Then, choose a tolerance ϵ , a decrease factor β , an interior start point α_N^0 , and an initial μ^0 . For iteration $i > 0$, $\alpha_N^{i+1} = s(\mu^i)$. Update μ using the equation $\mu^{i+1} = \beta\mu^i$. Terminate the iteration when $\mu^i B(\alpha_N^{i+1}) < \epsilon$. The solution procedure can be explained using the pseudo code in Algorithm 2 shown below.

Algorithm 2: Solve for optimal α_N and τ by interior barrier function method

Require: $N \geq 0, K \geq 1, \eta$

Ensure: α_N, τ^*

- 1: Set an interior point α_N^0 as a starting point.
 - 2: Choose a tolerance ϵ , decrease factor β , initial μ^0 , and barrier function $B(\alpha_N)$
 - 3: **while** $i \geq 0, a_i \geq \epsilon$ **do**
 - 4: $\tau^i = f(\alpha_N^i) + \beta\mu^i$;
 - 5: Using KKT conditions to solve the problem that minimize
 $f(\alpha_N^i) + B(\alpha_N)\mu^i$;
 - 6: Obtain the optimal solution α_N^{i+1} ;
 - 7: Update $\mu^{i+1} = \beta\mu^i$
 - 8: $i = i + 1$;
 - 9: **end while**
-

3.6 Performance Evaluation

In this section, numerical results are presented to evaluate the performance of our proposed algorithm. The round-robin TDMA is taken as benchmark in our simulation. Two metrics are used to evaluate the performance of our proposed scheme: (1) the average computation time which measures the delay performance

of our proposed algorithm; (2) the energy consumption which indicates the energy efficiency of our proposed algorithm.

The white Gaussian noise power is normalized to $\sigma^2 = 1$. The SINR threshold γ^* is set to be 5 dB [89]. The maximum waiting time is set as $\Gamma = 10\Delta$.

For the first part, the performance of our algorithm is investigated from the aspect of improvement on average computation time. Firstly, given certain error probability constraints, the influence of CDMA scheme to the average computation time is investigated under different network size. The effect of CDMA scheme is manifested by its spreading gain. Larger spreading gain indicates longer CDMA code. Therefore, the influence of the spreading gain G to average computation time is evaluated. Secondly, scalability measures the performance of an algorithm when the number of nodes in the network grows. The scalability is examined by observing the performance of average computation time as network size grows from 1 to 100. For the second part, the energy consumption of our proposed scheme is investigated.

3.6.1 Average Computation Time

Average Waiting Time for Round-Robin TDMA Transmission

As a benchmark, the average waiting time of round-robin TDMA transmission is given as the following. In round-robin TDMA, all the nodes take turns to transmit their reading. In this case, for a network of K nodes, the average waiting time for the best node can be computed as $\frac{(K-1)\Delta}{2}$, which grows linearly as network size K grows.

Hybrid Scheduling: The Effect of Spreading Gain G

First of all, CDMA of our hybrid scheme provides the ability to decode the maximum value even when other nodes transmit simultaneously with the best node.

Figure 3.5: Average waiting time τ_1 as a function of spreading gain G .

CDMA code with larger spreading gain can tolerate more concurrent transmissions. Therefore, given a certain computation failure probability, our proposed algorithm reduces the average waiting time. Fig. 3.5 plots the average waiting time τ_1 as a function of spreading gain G . As shown in the figure, for given network size K , when the length of the CDMA sequence increases, the reduction on average waiting time compared to the benchmark becomes more significant. For example, when network size $K = 10$, when G increases from 2 to 64, average waiting time drops from 0.5Δ to $10^{-5}\Delta$.

Hybrid Scheduling: Scalability

For our proposed hybrid scheme, the decision of transmission is made by each node in a decentralized way. Therefore, when the number of nodes grows, little additional control overhead will occur. Furthermore, when network scales, the delay performance of our proposed scheme is maintained at a satisfying level. Fig. 3.6 investigates the scalability performance for our algorithm. It can be observed that by relaxing the constraints on failure probability from strict $P_e = 0$ of benchmark TDMA to $P_e = 10^{-3}$, our proposed scheme has almost 10 times reduction on average waiting time. Moreover, Fig. 3.7 shows the average computation time as

Figure 3.6: Average waiting time τ_1 as a function of network size K .

a function of network size. It shows that the proposed scheme has significantly less average computation time when the network size K is more than 15. In small size network, the proposed scheme outperforms benchmark when $P_e = 10^{-2}$ and $P_e = 10^{-3}$. Therefore, it can be concluded that our hybrid scheduling algorithm has good scalability.

3.6.2 Energy Consumption

For the following results, the transmission energy of our proposed scheme is considered. The energy model is given as follows. We assume AWGN channels. Assume that all the nodes are randomly deployed within the transmission range r_0 of the sink node. The transmission power is proportional to the power of distance from sensor node to sink node, $P_{tx} \propto d_i^2$, Assume that the node with the maximum value lies in interval i , the average energy consumption can be given as

$$E_{tx} = M_i P_{tx} D. \quad (3.21)$$

The simulation results are averaged over 1000 runs.

Fig. 3.8 plots the energy consumption of our proposed scheme as a function

 Figure 3.7: Average computation time τ as a function of network size K .

of the network size, compare with the round-robin TDMA benchmark. When the network size is smaller than the collision tolerance of CDMA, $K < M^*$, in order to maintain small failure probability and small average computation time, our proposed hybrid scheme tends to schedule all the nodes to transmit at the same time. Hence, the average energy consumption increases as the network size grows. When $K \geq M^*$, the collisions with the best node are controlled by scheduling other nodes to transmit in different time slot. In our algorithm, once the maximum value is transmitted, the rest of the nodes will not transmit, only a portion of nodes whose timer slot allocations are the same with the best node will transmit. Therefore, the energy consumption of our proposed scheme decreases and converges.

3.7 Concluding Remarks

In this chapter, a fast, decentralized, and energy-efficient MAX function computation method through hybrid scheduling has been proposed for a single-hop network. In our proposed algorithm, a schedule assignment is generated for each node to determine its time, transmission sequence and transmission power. The hybrid

Figure 3.8: Average transmission energy E_{tx} as a function of network size K .

schedule assignment is designed so that the node with the maximum value has the priority to transmit. The synergy effect of priority-aware TDMA and CDMA scheme contributes to a novel technique for the reduction of average computation time and computation failure probability. By increasing the length of CDMA sequence, the performance of our algorithm can be improved. Numerical results show that our decentralized scheme is fast, scalable, and energy-efficient.

Chapter 4

Energy-Aware Hybrid Compressive Sensing for Data Gathering Sensor Networks

Function-based data aggregation significantly reduces data traffic, however, only the function value is delivered to the sink node while the rest of the information is lost. In this chapter, we will introduce data aggregation algorithm based on CS. CS technique is a promising method in which the data are compressed for transmission and recovered at the sink node with certain fidelity. There are several advantages of CS technique. First, only simple operation is needed at resource-limited sensor nodes, the complex recovery is shifted to the resource-abundant sink node. Second, different from traditional correlation-based compression algorithms that heavily rely on the prior information about the correlation model, the recovery of CS performs well without the information about the correlation model. CS is applied into multi-hop data gathering WSNs to reduce the number of transmissions and energy consumption. However, traditional CS assumes uniform energy cost for each transmitter, which is not suitable for WSNs with heterogeneous transmission links. In this chapter, we apply CS in situations where the transmission costs for wireless links in sensor network are different, and we seek to find an energy-efficient way to collect a more general class of sensory data with low latency. We propose an energy-efficient, low latency, and general data gathering scheme based

on hybrid CS with dense measurement matrix. We formulate a MINLP problem with the objective to prolong network lifetime and reduce data gathering latency. Since the MINLP problem is NP-hard, it is decomposed into an energy-efficient hybrid tree construction problem and a SINR-based scheduling problem. Then, two approximation algorithms with provable bounds are proposed to solve the two decomposed problems respectively. Finally, extensive simulation results demonstrate the performance of the proposed scheme in terms of recovery quality, total energy consumption, data gathering latency and network lifetime. The simulation results show that, compared with benchmark algorithms, the proposed scheme significantly reduces total energy consumption and data gathering latency and dramatically improves network lifetime with satisfying recovery quality.

4.1 Introduction

As we discussed in Chapter 2, data gathering by WSNs is one of the crucial components of IoT. In the following, we recapitulate the challenges and motivations to apply CS technique into multi-hop sensor network.

Sensors are deployed in a field to gather environmental information. And the data will be sent periodically to the sink node. There are mainly two challenges to address in such a network. The first challenge lies in improving the overall energy efficiency and extending network lifetime with energy-constrained sensor nodes. The network lifetime is the duration of time when the network can perform its task correctly. Specifically, the network lifetime can be defined as the operation time of the sensor network until the first node drains out its energy [90]. However, the energy cost to transmit sensor data to the sink node is different for each node due to diverse wireless link conditions. Moreover, in tree-based multi-hop data gathering, the nodes near the sink node suffer from heavier traffic load than other nodes since they have to relay data for downstream nodes, which is called the hot

spot problem [53]. Both energy diversity and the hot spot problem contribute to the unbalanced energy consumption among the network, which impairs energy efficiency and shortens the network lifetime. The second challenge is to provide a guarantee on the data gathering latency, since in various applications, outdated information has no value or even can cause negative effects [20].

Previous research efforts have shown that applying CS technique into multi-hop data gathering network can relieve hot spot problem by spreading the traffic load evenly in the network [48,49]. In addition, CS suggests a way of reducing data traffic without adapting to data correlation structure [91]. Nonetheless, directly applying CS, referred to as plain CS, may not be the best choice [49], since it may cause unnecessary higher traffic for downstream nodes than traditional non-CS schemes, leading to more energy consumption for downstream nodes. Besides, in CS-based data gathering, a node can transmit only when it receives all the information from downstream nodes, which inherently leads to longer latency than non-CS scheme, whose transmission schedule is not constrained by its transmission order.

Existing research works try to solve the problem of plain CS in different ways. Xie *et al.* [92] proposed a hybrid CS based data gathering in clustered network in order to further reduce the number of transmissions. Ebrahimi *et al.* [51] exploited sparse random projections to reduce the number of transmissions and reduce data gathering latency. In sparse random projections, there are only a few of non-zero elements in each row of the measurement matrix, which corresponds to less transmissions to improve energy efficiency. However, when using sparse random projections, the sparsity of the measurement matrix and the sparsity of the signal will affect the number of projections needed to recover the signal [93]. In contrast, dense measurement matrices guarantee the recovery of signal with a stable number of random projections regardless of the sparsity of the signal. Moreover, in [51,92], the authors assumed uniform energy cost for all the wireless links. However, the

cost of transmitting a packet on different wireless links may vary due to distance, channel fading, etc. Therefore, the diversity of energy cost should be considered in the CS based data gathering framework. Xiang *et al.* [49] applied hybrid CS with dense measurement matrix into multi-hop data gathering WSN to minimize total energy consumption. Whereas, the greedy algorithm in [49] constructs a tree to minimize total energy without considering the data gathering latency.

This chapter presents an energy-aware and low latency data gathering scheme based on hybrid CS with a dense measurement matrix. The energy cost diversity of the WSNs is considered when designing the hybrid CS based data gathering scheme. CS with a dense measurement matrix is used to guarantee the recovery of a more general class of data. Furthermore, in order to reduce overall energy consumption while maintaining the quality of recovery, hybrid CS is used together with dense measurement matrix. Additionally, a scheduling method to efficiently schedule the compressed data under a physical interference model is also developed. The hybrid CS based data gathering problem is formulated as a multi-objective MINLP problem to minimize data gathering latency and to extend network lifetime. Due to the NP-hardness of the problem, two approximation algorithms are proposed to solve the decomposed problems. On one hand, an energy and latency aware hybrid CS based tree is constructed to minimize link energy cost as well as to reduce data gathering latency. Sensor nodes are categorized as aggregators and forwarders according to the tree topology and a predefined CS factor, which introduces hybrid traffic into the network. Aggregators transmit coded sensor readings while forwarders just transmit the original sensor readings. On the other hand, SINR-based priority-aware scheduling utilizes spatial reuse to reduce data gathering latency while guaranteeing successful concurrent transmissions. To achieve this, for a particular link to transmit successfully, the accumulated interference from other simultaneous transmissions should be controlled below a tolerable threshold. More specifically, the main contributions in this chapter can be sum-

marized as follows:

- An energy-aware and low latency hybrid CS based scheme with a dense measurement matrix is proposed for general data gathering in WSNs.
- The hybrid CS based data gathering problem is formulated as a MINLP problem in order to seek the optimal routing topology, aggregator set and scheduling assignment to maximize network lifetime and minimize data gathering latency.
- Due to the NP-hardness of the problem, a practical solution is given by dividing the problem into two sub-problems: (1) to construct a hybrid CS based, energy-efficient, and latency aware data gathering tree, obtain the set of aggregators and forwarders (2) to develop a SINR-based scheduling scheme for the hybrid data traffic to minimize the total time slots. Two approximation algorithms are proposed to solve the two sub-problems.
- The performance bounds of our proposed tree construction and SINR-based link scheduling algorithms are analyzed, respectively.
- The performance of the proposed algorithm is evaluated through extensive simulations, the results of which show that the proposed algorithm outperforms the benchmark algorithms.

The remainder of the chapter is organized as follows. Section 4.2 sets up the system model and formally describes the hybrid CS based data gathering problem. In Section 4.3, the optimization problem is formulated as a multi-objective MINLP problem. Section 4.4 presents the proof of the NP-hardness of the investigated problem. In Section 4.5, the NP-hard problem is decomposed into two sub-problems, and approximation algorithms are developed to solve them. In Section 4.6, the performance bounds on the tree construction algorithm and scheduling

algorithm are given respectively. Section 4.7 evaluates the performance of the proposed algorithm through simulation. Section 4.8 gives concluding remarks of this chapter.

4.2 System Model

4.2.1 Network Model

The WSN is modeled as a weighted connected graph $G = (V, E)$, where V is the set of n nodes that are randomly deployed in certain region, E denotes the set of wireless links between any two sensor nodes that reside within each other's communication radius. Fixed and uniform transmit power P is assumed for all sensor nodes. Further, we define a cost matrix $\mathbf{C} \in \mathbb{R}_+^{n \times n}$ on edges of the network with entries $c_{ij} : (i, j) \in E$ denoting the link energy cost to transmit one data packet on link (i, j) . Assume that all packets are of equal size. All the sensor nodes access the wireless link through TDMA scheme where time is divided into fixed-length slots. Physical interference model which is derived from simplified propagation model is used. With the physical interference model, receiver j can successfully receive the data from transmitter i if the SINR of link (i, j) , denoted as γ_{ij} , is above a threshold β , whose expression is given as follows:

$$\gamma_{ij} = \frac{P d_{ij}^{-\alpha}}{\sum_{(k,l) \in E: k \neq i} P d_{kj}^{-\alpha} + \eta} \geq \beta, \quad \forall (i, j) \in E, \quad (4.1)$$

where d_{ij} is the Euclidean distance between node i and node j . P is the transmit power, $\alpha \geq 2$ is the path loss exponent, η is the ambient noise. The total time slots to schedule all the transmissions is defined as the data gathering latency. Assume that each node has the same initial energy E_0 , then the network lifetime is defined as the number of operation rounds until the first node depletes its energy.

Notations

$\mathbf{u}$ denotes the data vector, Ψ denotes transform basis, Φ denotes the measurement matrix, θ denotes the transform coefficient vector, $\|\theta\|_{\ell_1}$ denotes the ℓ_1 -norm of θ , m denotes the CS factor.

4.2.2 Hybrid CS Based Data Gathering

Hybrid CS technique is exploited to extend network lifetime and to reduce data gathering latency. This section introduces the principles of hybrid CS and how it can be integrated in to data gathering framework of WSNs.

Previous research works have shown that various types of real-life signals are compressible under proper transform basis [48, 94, 95]. According to CS theory, if an n -dimension data vector is sparse/compressible in some domain, then the data vector can be accurately reconstructed from far less observations than the original data vector dimension.

More specifically, consider a data vector $\mathbf{u} \in \mathbb{R}^n$. $\mathbf{u}$ is sparse under orthonormal transform basis Ψ of size $n \times n$, which means that $\mathbf{u}$ can be represented by a k -sparse vector θ , such that $\mathbf{u} = \Psi\theta$, where θ is a k -sparse coefficient vector, only k out of n elements of θ are non-zero, and $k \ll n$. Fourier transform basis, wavelet transform basis and DCT basis can be used as Ψ to obtain the sparse representation for a specific data vector.

Then, we collect m random linear projections of the data vector with measurement matrix Φ , obtaining $\mathbf{y} = \Phi\mathbf{u} = \Phi\Psi\theta$, where $\mathbf{y}$ is an $m \times 1$ vector. Φ is an $m \times n$ dense measurement matrix whose entries can be generated according to i.i.d Gaussian random distribution or Bernoulli random distribution [37]. CS can accurately reconstruct $\mathbf{u}$ from $\mathbf{y}$ as long as the matrix $\mathbf{A} = \Phi\Psi$ satisfies the RIP condition. Fortunately, with a fixed basis Ψ and a dense random measurement matrix Φ , matrix $\mathbf{A}$ satisfies RIP with high probability [37]. For simplicity, in this

(a) Non-CS data gathering. Node 1 is the sink node. The total traffic load for the network is 19, the maximum traffic load is 5.

(b) Plain CS based data gathering. CS factor $m = 2$. Node 1 is the sink node. The total traffic load for the network is 16, the maximum traffic load is 2.

(c) Hybrid CS based data gathering. CS factor $m = 2$. Node 1 is the sink node. Node 2, 3, 6, 7, 9 are selected as aggregators. The total traffic load for the network is 13, the maximum traffic load is 2.

Figure 4.1: Comparison of non-CS, plain CS and hybrid CS based data gathering schemes.

chapter, the number of random linear projections is defined as CS factor.

As for data reconstruction, the sink node performs the ℓ_1 -norm minimization as shown below in (4.2) to compute the sparsest θ .

$$\min_{\theta} \|\theta\|_{\ell_1} \quad \text{subject to} \quad \mathbf{y} = \Phi \Psi \theta, \quad (4.2)$$

where $\|\theta\|_{\ell_1} = \sum_i |\theta_i|$. Let $\hat{\theta}$ be the optimal solution for the above optimization problem. The reconstructed data vector is given by $\hat{\mathbf{u}} = \Psi \hat{\theta}$. ℓ_1 -minimization can be solved in polynomial time by linear programming.

Remark: In contrast to the dense measurement matrix used in our proposed scheme, Φ can also be sparse random matrix in which only a few nonzero entries exist in each row [55]. However, if Φ is sparse, larger number of random linear projections are needed in order to guarantee the accurate recovery of the data vector. Moreover, sparse random projections may not work well when the data is also very sparse [55]. Hence, in our proposed scheme, in order to reduce the number of random projections, and to guarantee the recovery of more general types of data vector, dense measurement matrices are adopted.

Next, we elaborate on how to apply hybrid CS in a multi-hop WSN. The key idea of hybrid CS is to start CS coding only when the outgoing traffic load is no less than m , otherwise raw data are transmitted [49]. CS factor m is assumed to be a global knowledge. Accordingly, two types of nodes are defined in the network. If a node has more than $m - 1$ downstream nodes, it's an aggregator, otherwise it's a forwarder. Suppose there are n nodes deployed in a sensing field. Each node acquires a reading u_i . The goal is to collect all the sensory data $\mathbf{u} = [u_1, \dots, u_n]^T$ at the sink node through multi-hop data transmission. A forwarder transmits the raw data u_i . An aggregator transmits coded readings that are produced by multiplying the original sensory data u_i with corresponding column vector $\phi_i = [\phi_{1i}, \dots, \phi_{mi}]^T$ in the measurement matrix Φ . Then, the coded readings are summed along the route to the sink node. Eventually, the sink node receives m linear combinations

of the original data vector, $\mathbf{y} = \sum_{i=1}^n u_i \phi_i$, rather than a length n raw data. Then, the sink node runs the recovery algorithm to obtain the n raw data. Here, the measurement matrix Φ is assumed to be global *a priori* knowledge for every node in the network [48], or the column vector ϕ_i can be generate by node i locally with specific pseudo-random number generator with a publicly known algorithm and its seed [49, 54]. Fig. 4.1 gives a comparison between non-CS, plain CS and hybrid CS based data gathering, assuming CS factor $m = 2$. As shown, in non-CS data gathering, the nodes near the sink carry heavy traffic load because they need to relay accumulated traffic from their downstream nodes. While for the plain CS based data gathering, the data packets are compressed en-route so that the traffic load for each communication link is the same. For hybrid CS based data gathering, CS coding is applied only when the traffic load exceeds $m - 1$ at node 2, 3, 6, 7, 9. Thus, hybrid CS based data gathering not only relieves the hot spot problem but also reduces the total traffic load, which leads to reduced data gathering latency and less total energy consumption.

4.2.3 Problem Description

Given the system model, the hybrid CS based data gathering problem can be elaborated as follows. Given a connected weighted graph G , find the subsets of links to form a latency-aware, hybrid CS based data gathering tree rooted at the sink node, categorize the sensor nodes as aggregators and forwarders. Given the hybrid CS based data gathering tree, allocate transmission time slot for each link so that the data gathering latency is minimized, network lifetime is prolonged and the SINR constraint for each link is satisfied.

Table 4.1: Notations in problem formulation

Parameters	
E_0	The initial energy for the sensor nodes.
S	A set of time slots that is large enough to deliver packets from all sensor nodes to the sink.
V	The set of nodes in the network.
E	The set of edges in the network.
T	The hybrid CS based data gathering tree.
r	The sink node.
n	Total number of nodes in the network.
m	CS factor.
P	Node transmission power.
α	Path loss exponent.
β	SINR threshold.
η	Background noise.
c_{ij}	The link energy cost to transmit one data packet over link (i, j) .
d_{ij}	The Euclidean distance between node i and node j .
Variables	
$a_{ij}^s \in \{0, 1\}$	Indicate whether link (i, j) is activated in time slot s .
$x_{ij} \in \{0, 1\}$	Indicate whether link (i, j) is selected in the data gathering tree.
$f_{ij} \geq 0$	The data traffic load (number of packets transmitted) on link (i, j) .
$\bar{f}_{ij} \geq 0$	The virtual data traffic load on link (i, j) , used to specify connectivity.
$y_i \in \{0, 1\}$	Indicate whether node i is an aggregator.

4.3 Problem Formulation

In this section, the hybrid CS based data gathering problem is formulated as a multi-objective MINLP problem. The notations used in this section is given in Table 4.1. The constraints are derived as the following.

4.3.1 Hybrid CS Based Tree Construction Constraints

First of all, hybrid CS technique changes the traditional flow conservation constraints. If node i is an aggregator, the outgoing traffic load always equals to m , if node i is a forwarder, the outgoing traffic load equals the incoming traffic load plus one. A set of extended flow conservation constraints are given to characterize this phenomena [49] as follows.

$$\sum_{j:(i,j) \in E} f_{ij} - \sum_{j:(j,i) \in E} f_{ji} + (n-m)y_i \geq 1, \quad \forall i \in V \setminus \{r\}. \quad (4.3)$$

$$\sum_{j:(i,j) \in E} f_{ij} - (m-1)y_i \geq 1, \quad \forall i \in V \setminus \{r\}. \quad (4.4)$$

Moreover, the sink node needs to collect data from all the nodes in the network. Therefore, the data gathering tree is a spanning tree. Constraint (4.5) follows directly from the property that the number of edges for an n -node spanning tree is $n-1$.

$$\sum_{(i,j) \in E} x_{ij} = n-1. \quad (4.5)$$

In the following constraints, virtual link flow $\bar{f}_{ij}$ is used to ensure connectivity from every node to the sink node.

$$\sum_{j:(i,j) \in E} \bar{f}_{ij} - \sum_{j:(j,i) \in E} \bar{f}_{ji} \geq \frac{1}{n-1}, \quad \forall i \in V \setminus \{r\}. \quad (4.6)$$

$$x_{ij} - \bar{f}_{ij} \geq 0, \quad \forall (i,j) \in E. \quad (4.7)$$

The following constraints state that only links that carry positive traffic flow

are selected to form the data gathering tree.

$$f_{ij} - x_{ij} \geq 0, \quad \forall (i, j) \in E. \quad (4.8)$$

$$x_{ij} - \frac{f_{ij}}{n} \geq 0, \quad \forall (i, j) \in E. \quad (4.9)$$

4.3.2 Scheduling Constraints

Given the constraints to build a hybrid CS based tree, more constraints should be specified to achieve successful transmission as well as to facilitate compressive operation.

First of all, a link should be scheduled only when it's on the hybrid data gathering tree, this constraint can be expressed as

$$a_{ij}^s \leq x_{ij}, \quad \forall s \in S, \quad \forall (i, j) \in E. \quad (4.10)$$

In addition, due to the half duplex feature of the sensor nodes, a node cannot transmit and receive in the same time slot. This constraint can be given as

$$\sum_{i:(i,j) \in E} a_{ij}^s + \sum_{k:(j,k) \in E} a_{jk}^s \leq 1, \quad \forall s \in S. \quad (4.11)$$

Furthermore, in order to guarantee successful transmission on link (i, j) , the SINR constraint for link (i, j) should be satisfied. For a particular link (i, j) , concurrent transmissions of other links will be regarded as interference. The activated subset of links are different for each slot. Therefore, the SINR for link (i, j) is different in each slot. The following constraints guarantee that γ_{ij} is above the threshold β for each time slot s .

$$\frac{P d_{ij}^{-\alpha} a_{ij}^s}{\eta + \sum_{(k,l) \in E; k \neq i} P d_{kj}^{-\alpha} a_{kl}^s} \geq \beta, \quad \forall s \in S, \quad (i, j) \in E. \quad (4.12)$$

On the hybrid CS based data gathering tree, there are two types of nodes. The aggregator needs to transmit m coded packets while the forwarder needs to

4.3. PROBLEM FORMULATION

transmit its own original packet and to relay all incoming packets for its children. The sensor node can transmit only one packet in one time slot. The number of activated time slots for outgoing links is different for aggregators and forwarders. But for both aggregators and forwarders, the activated time slots for outgoing links equal to the outgoing traffic load. Therefore, the activated time slots can be expressed as

$$\sum_{s \in S} a_{ij}^s = f_{ij}, \quad \forall (i, j) \in E. \quad (4.13)$$

In order to facilitate CS operation, a node needs to collect all the data from its downstream nodes before transmitting the coded packets. Constraints to guarantee transmission order are specified as in [51]. For example, in time slot s , if node i wants to transmit its data via link (i, k) , it needs to verify that all its children have already transmitted in previous time slots.

$$\sum_{\bar{s}=1}^{s-1} \sum_{j:(j,i) \in E} a_{ji}^{\bar{s}} + M(1 - a_{ik}^s) \geq \sum_{j:(j,i) \in E} x_{ji}, \quad (4.14)$$

$$\forall s \in S, \quad \forall (i, k) \in E,$$

in which M is a sufficiently large constant that is larger than the total number of links in the spanning tree T .

Given the above constraints, the multi-objective optimization problem can be formulated as the following,

$$\begin{aligned} & \text{maximize} \quad J_1 = \frac{E_0}{\max_{(i,j) \in E} \{c_{ij} f_{ij} x_{ij}\}} \quad \text{and} \\ & \text{minimize} \quad J_2 = \sum_{(i,j) \in T} \sum_{s \in S} a_{ij}^s, \\ & \text{subject to} \quad \text{constraints (4.3)-(4.14)}. \end{aligned} \quad (4.15)$$

Here, the the first objective, J_1 , is the network lifetime counted in operation rounds. The second objective, J_2 , is the data gathering latency counted in time slots.

4.4 Complexity Analysis

With the boolean variables and the multiple objectives, the formulated problem is actually NP-hard. The authors of [96] have proved that the SINR-based data gathering problem is NP-hard by reducing from the *max-connections* problem [97]. However, our proposed scheduling problem is more complex due to the different traffic load at the aggregator and forwarder and the consideration of transmission order to allow hybrid CS coding. The following lemma states the NP-hardness of the above optimization problem.

Lemma 4.1. *The problem of developing a data gathering scheme to maximize network lifetime as well as to minimize data gathering latency based on hybrid CS through joint tree construction and SINR-based scheduling is NP-hard.*

Proof. The above optimization problem involves two objectives: (1) to maximize network lifetime (2) to minimize the total data gathering latency with physical interference model. Our problem can be then expressed as the following: among all the hybrid CS based data gathering trees that maximize the network lifetime, find the tree and corresponding scheduling allocation that lead to the least data gathering latency while the SINR constraints for all links are satisfied.

Firstly, the complexity of constructing a hybrid CS based data gathering tree that maximize network lifetime is evaluated. The formula of J_1 indicates that, given a fixed initial energy E_0 , maximizing network lifetime is to minimize the maximum transmission energy cost $f_{ij}c_{ij}$ in the network. Since for hybrid CS based data gathering scheme, the traffic load f_{ij} is upper bounded by the CS factor m , the problem is then transformed into one to minimize the maximum link energy cost c_{ij} . That is, to find a minimum bottleneck spanning tree (MBST), which asks for a spanning tree to minimize edge cost of the bottleneck edge. The bottleneck edge of a spanning tree is the edge with the highest edge weight. A graph may have many MBSTs. The problem of finding MBST can be efficiently solved by Camerinis algorithm [98].

Next, assuming that the hybrid CS based data gathering tree is given, we prove that the SINR-based data gathering problem for a hybrid CS tree is NP-hard by reducing from the One-Shot Scheduling problem, which has been proved to be NP-hard [99]. The One-Shot Scheduling problem is to pick a subset of weighted links such that the total weight is maximized and the SINR of every scheduled receiver is above the threshold. The problem of finding the minimum latency schedule for a given tree can be decomposed into a series of One-Shot Scheduling problem. In each One-Shot Scheduling problem, an auxiliary graph is constructed from the given tree, an edge is added to the auxiliary graph if it satisfies the following criteria: (1) the corresponding link in the data gathering tree has remaining traffic load; (2) the link contains leaf node or its children links have completed transmission. After resolving the One-Shot Scheduling problem on the auxiliary graph, the traffic load at each link is updated. This step is repeated until the traffic load of every link becomes zero. The SINR-based minimum latency schedule problem is a combination of several One-Shot Scheduling problems. Therefore, the SINR-based data gathering problem for a hybrid CS tree is NP-hard. In conclusion, the problem of developing a hybrid CS based data gathering scheme to maximize network lifetime as well as to minimize data gathering latency through joint tree construction and SINR-based scheduling is NP-hard.

4.5 Approximation Algorithms

In order to give a practical solution to the NP-hard problem, we decompose the hybrid CS based data gathering problem into two sub-problems: to construct an energy-efficient and latency aware hybrid CS based tree, and to schedule link transmission such that the data gathering latency is minimized. Two approximation algorithms, the hybrid CS based data gathering tree construction (HCS-DGT) algorithm and the SINR-based priority-aware link scheduling (SPLS) algorithm, are proposed.

4.5.1 Hybrid CS Based Data Gathering Tree Construction

First of all, a HCS-DGT algorithm is proposed. The goal is to minimize the maximum energy cost as well as the data gathering latency. When building data gathering tree, two criteria are considered. Firstly, links with less energy cost are preferred. Secondly, the constructed tree should introduce less data gathering latency, which is achieved by constraining the node degree in the tree. Since the data may be compressed en-route, a node can transmit only when all its children have finished transmission. Hence, if a node has many children, the transmission latency increases. Additionally, if the data gathering tree is tall, then it takes a long time for all the data to reach the sink node. Combining the above observations, the aim is to develop a degree-constrained balanced data gathering tree. In HCS-DGT algorithm, each node finds a shortest path to the sink node based on the edge cost C . Then, a hop count h_v is assigned to each node. This procedure guarantees that the data packets are transmitted through the route with least link energy cost. Node v belongs to layer h if it's h hops away from the sink node. Nodes at layer 1 are connected to the sink node and added to T . Then, layer by layer, nodes in layer h select a parent from nodes in layer $1, \dots, h-1$. Node p with the minimum weight is selected as the parent of v . The weight of node p is defined as the sum of node degree and the size of sub-tree that rooted at node p . Then, edge (p, v) is added to tree T . This operation is repeated until all layers are examined and all the nodes find a parent. After the tree is constructed, the number of descendants for each node is computed. A node is selected as the aggregator and added to aggregator set A if the its number of descendants is no less than the CS factor m , otherwise, it is added to forwarder set F . The procedure of tree construction is summarized in Algorithm 3.

The time complexity of our proposed tree construction problem is analyzed as the following. It takes $O(|V|)$ to find a shortest path for every node. Assume

the maximum hop count is H , and the number of nodes in level h is n_h . We have $n_1 + n_2 + \dots + n_H = |V| - 1$. In the worst case, it takes $O(n_h n_{h-1})$ to select parents for nodes in layer h . The time complexity for constructing the tree is $O(n_1 n_2 + n_2 n_3 + \dots + n_{H-1} n_H) = O((|V| - 1)^2)$. It takes $O(|V|)$ to get the number of descendants for every node. Hence, the overall time complexity of HCS-DGT algorithm is $O(|V| + (|V| - 1)^2 + |V|) = O(|V|^2)$.

- *Practical Implementation:* During the setup stage, each node acquires information about its neighbors distributively. Then all the information is collected at the sink node. The sink node then runs Algorithm 1 to compute the hybrid CS based data gathering tree T and disseminates the routing information to the whole network.

4.5.2 SINR-Based Priority-Aware Link Scheduling

Given the hybrid CS based data gathering tree, we aim to find a scheduling method that enables en-route hybrid data compression, guarantees successful transmission with physical interference model, and minimizes the data gathering latency for the network. The goals are challenging due to the fact that the scheduling coordination to satisfy the SINR constraint for each node is notoriously hard. What's more, the CS coding operation inherently brings longer delay compared to pipeline scheduling algorithm where the transmission order is not the concern. This section presents a SPLS algorithm. To begin with, a priority value, traffic load and the *Txbuffer* value are assigned to each transmitter. The priority value of a node is determined by its parent's node degree, and its hop count to the sink node. The traffic load of a node is the total amount of traffic that will flow through a node. The *Txbuffer* records the current number of packets for transmission. For a forwarder, its traffic load equals to the number of its downstream nodes and the initial *Txbuffer* is one. For an aggregator, the traffic load and the initial *Txbuffer* are both m . Next, the

Algorithm 3: HCS-DGT algorithm.

Input : Graph $G = (V, E)$, cost matrix $\mathbf{C}$, sink node r , CS factor m .
Output: Hybrid CS based data gathering tree $T = (V, E')$, aggregator set A , forwarder set F .

```

1 foreach node  $v \in V$  do
2   find the shortest path from node  $v$  to the sink node based on  $\mathbf{C}$ , record
   hop count  $h_v$  to the sink node ;
3   node  $v$  learns a set of neighbors  $N(v)$ ;
4 end
5 forall the  $v \in \{v | h_v = 1\}$  do
6   parent( $v$ ) =  $r$  ;
7    $T = T \cup \{v\}$  ;
8 end
9  $Bestweight = \infty$  ;
10 for  $layerNo = 2$  to  $maxhop$  do
11   foreach  $v \in \{v | h_v = layerNo\}$  do
12     forall the  $p \in N(v)$  &  $h_p < h_v$  do
13        $T_{test} = T \cup \{p, v\}$  ;
14        $weight = d(p) + height(p)$  ;
15       if  $weight < Bestweight$  then
16          $Bestweight = weight$ ;
17          $T_{cand} = T_{test}$ ;
18       end
19     end
20   end
21    $T = T_{cand}$ 
22 end
23 Get the number of descendants  $numDes$  for every node ;
24 When  $numDes(v) \geq m$ ., add  $v$  into  $A$ , otherwise, add  $v$  into  $F$  ;

```

interference neighborhood around receiver j of link (i, j) is determined with the interference localization method in [100]. If node i wants to successfully transmit to node j , simultaneous active transmitters within the interference neighborhood of receiver node j should be controlled such that the total interference coming from inside the interference neighborhood is less than $(1 - \epsilon)I_{ij}^{max}$, where $\epsilon \in [0, 1]$ is a constant, $I_{ij}^{max} = \frac{Pd_{ij}^{-\alpha}}{\beta}$ is the maximal tolerable interference for successful transmission on link (i, j) .

In each time slot t , transmitters that meet the following requirements are added to the *ReadySet*: (a) the transmitter is a leaf node or all of its children have finished transmission; (b) the transmitter has remaining traffic load. Then, the *ReadySet* is sorted based on the priority in a descending order to get *SortedReadySet*. At the beginning, the schedule for time slot t , S_t , is empty. Then, the transmitter with the highest priority is added to S_t , its traffic load is reduced by 1. If the parent of the considered transmitter is a forwarder, its *Txbuffer* is increased by 1. The remaining transmitter u of link (u, v) in *SortedReadySet* is added to S_t if the total interference from the transmitters in current schedule to receiver v is less than its tolerable level, and the interference from u to the receiver of transmitters in the current schedule is less than their tolerable level. The active transmitter set for time slot t is finalized after all transmitters in the *SortedReadySet* are examined once. Then, a new time slot schedule starts and the procedure is repeated until traffic load of all transmitters become zero. The procedure of SPLS can be summarized as in Algorithm 4.

Now, the time complexity of our link scheduling algorithms is analyzed as the following. It takes $O(|V|)$ to assign priority to each node. It takes $O(|V|)$ to determine interference neighborhood. Then, it takes $O(|V|)$ to traverse every node and judge whether it can be added to *ReadySet*. Assume that on average, there are a transmitters in the *ReadySet*, then the sort step takes $O(a)$. Assume that for each transmitter, it takes on average b attempts to add it into the schedule, then

the scheduling step takes $O(ab)$. Assume that the average traffic load for each link is λ , the outer loop may run $\lambda(|V| - 1)$ times. The overall time complexity for our algorithm is $O((3|V| + ab)\lambda(|V| - 1))$. a, b , and λ are bounded by $|V|$. Therefore, the time complexity of our algorithm is $O(|V|^4)$.

- *Practical Implementation:* Given the hybrid CS based data gathering tree T from Algorithm 1, the sink node runs Algorithm 2 to obtain the scheduling matrix S and disseminates S to all other nodes. Then, sensor nodes transmit their sensor readings through the given topology in determined time slots.

Algorithm 4: SPLS algorithm

Input : Hybrid CS based data gathering tree T , aggregator set A ,
forwarder set F .
Output: A scheduling matrix S .

- 1 Assign traffic load to each node, for a aggregator, $load(v) = m$,
 $Txbuffer(v) = m$; for a forwarder, $load(v) = numDes(v)$,
 $Txbuffer(v) = 1$;
- 2 Calculate priority for each node ;
- 3 Compute the interference neighborhood for each node ;
- 4 $t = 1$;
- 5 $sumLoad = sum(load)$;
- 6 **while** $sumLoad > 0$ **do**
- 7 Get *ReadySet* and sort *ReadySet* based on priority, obtain
 SortedReadySet;
- 8 **forall** the node $v \in SortedReadySet$ **do**
- 9 **if** activating v won't conflict with transmissions in set S_t &
 transmission of node v won't be affected by S_t **then**
- 10 $S_t = S_t \cup \{v\}$;
- 11 $load(v) --$;
- 12 $Txbuffer(v) --$;
- 13 **if** $v.parent \in F$ **then**
- 14 $Txbuffer(v.parent) ++$;
- 15 **end**
- 16 **end**
- 17 **end**
- 18 $t = t + 1$;
- 19 Update priority;
- 20 Update $sumLoad$;
- 21 **end**

4.6 Performance Analysis

This section presents the performance analysis of the proposed algorithm by evaluating the approximation ratio of the proposed tree construction algorithm and giving the bound for the scheduling algorithm.

4.6.1 Approximation ratio of the tree construction algorithm

Lemma 4.2. *Given a connected weighted graph $G = (V, E)$ with cost matrix $\mathbf{C}$, if the maximum edge cost is c_{max} , the minimum edge cost is c_{min} . Define the ratio of c_{max} and c_{min} as $\rho = \frac{c_{max}}{c_{min}}$. Then, the HCS-DGT algorithm is better than ρ -approximation to the optimum.*

Proof. The objective is to construct a tree that maximizes network lifetime. According to the expression of J_1 , given a fixed initial energy, the optimal solution minimizes the maximum energy cost. Write the optimum value of the maximum energy cost as $OPT = \min\{\max_{(i,j) \in T^*} \{f_{ij}c_{ij}\}\}$, where T^* is the optimal data gathering tree. For hybrid CS data gathering, the maximum traffic load of the network is m . The lower bound for OPT can be obtained as $OPT \geq mc_{min}$. The proposed HCS-DGT algorithm gives the performance of mc_{ij} , where $(i, j) \in T$ and T is the proposed tree. The approximation ratio is $\frac{c_{ij}}{c_{min}} \leq \frac{c_{max}}{c_{min}} = \rho$. That is, the maximum energy cost of the proposed tree is at most ρ times the optimum value. Therefore, the proposed HCS-DGT algorithm is better than ρ -approximation to the optimum solution.

4.6.2 Bounds of the scheduling algorithm

Lemma 4.3. *Given a hybrid CS based data gathering tree, the lower bound on the time slots to schedule all the transmissions is*

$$\max_{i \in V} \left\{ \sum_{j \in D_i} f_{ji} + f_{ik}h_i \right\}, \quad (4.16)$$

where k is the parent of node i , D_i is the set of children of node i , h_i is the hop count of node i to the sink node.

Proof. In the hybrid CS based data gathering tree, every node needs to wait for their children nodes to finish transmission so that data can be compressed and transmitted towards the sink node. Denote the set of children of node i as D_i . For each node j in the set D_i , the traffic from j to i is f_{ji} . Then, the minimum number of time slots to complete gathering data from children nodes is $\sum_{j \in D_i} f_{ji}$. Similarly, after collecting all the data from its children nodes, node i needs to send the aggregated or non-aggregated data through h_i hops to the sink node, which requires at least $f_{ik}h_i$ slots. If all transmissions can be done in parallel without collision, then the lower bounds on the schedule time slot is (4.16).

In the worst case, if the SINR constraints cannot be fulfilled, only one transmitter can transmit in one time slot, then the total number of data gathering time slots will be upper bounded by the total number of traffic load of the network.

4.7 Performance Evaluation

Figure 4.2: Recovery error as a function of CS factor. $n = 400$.

This section evaluates the performance of the proposed algorithm in terms of: recovery quality, the total energy consumption, the total number of time slots for

data gathering and the network lifetime. Network lifetime is measured in terms of operation rounds. For comparison, plain CS with SPLS scheduling, MECDA [49] with SPLS, and non-CS with MWF [96] scheduling method are also implemented.

In this simulation, the sensor nodes are randomly deployed in a square field of 500×500 unit distance such that the network is connected. Assume that each node has an initial energy storage of $E_0 = 100$ J. Assume that packet length L is 2000 bits/packet. The transmit power for all nodes are normalized to $P = 1$ and the ambient noise is negligible. Moreover, assume that the path loss exponent $\alpha = 3$, SINR threshold $\beta = 2$, $\epsilon = 0.25$ [51]. We vary the number of sensor nodes from 50 to 450, with communication radius ranges from 33 to 100. The transmission energy cost for different links is calculated based on first-order radio model [18] as in the following. Assume that the packet length is L bits, the link energy cost to transmit one packet on link (i, j) can be computed as, $c_{ij} = E_{tx}(L, d_{ij}) + E_{rx}(L) = 2E_{elec}L + \epsilon_{amp}Ld_{ij}^2$, where d_{ij} is the Euclidean distance between node i and node j , E_{elec} is the energy used to run transmitter or receiver circuitry, ϵ_{amp} is the energy used to achieve acceptable $\frac{E_b}{N_0}$, the energy per bit to noise power spectral density ratio. Without loss of generality, other energy consumption models such as [84] can also be used, which will not affect the performance superiority of our proposed algorithm.

First, Fig. 4.2 gives an example of recovery quality of our proposed algorithm. The recovery quality is measured by recovery error. The definition of recovery error can be given as,

$$e = \frac{\|\mathbf{u} - \hat{\mathbf{u}}\|_2^2}{\|\mathbf{u}\|_2^2}, \quad (4.17)$$

where $\mathbf{u}$ is the original data vector, and $\hat{\mathbf{u}}$ is the recovered data vector. Synthetic data vectors are generated for $n = 400$. The compressed data are recovered using ℓ_1 -minimization algorithm. The recovery error is evaluated for different CS factor and data vector sparsity k . As shown in Fig. 4.2, the recovery quality of the

(a) Total energy consumption as a function of the network size. The number of nodes in the network grows from 100 to 450. CS factor $m = 10\%n$.

(b) Total data gathering time slots as a function of the network size. The number of nodes in the network grows from 100 to 450. CS factor $m = 10\%n$.

(c) Network lifetime as a function of the network size. The number of nodes in the network grows from 100 to 450. CS factor $m = 10\%n$.

Figure 4.3: Performance of non-CS, plain-CS, and hybrid CS based data gathering under different network sizes.

(a) Total energy consumption vs CS factor, CS factor ranges from $5\%n$ to $40\%n$, network size $n = 200$.

(b) Total data gathering time slots vs CS factor, CS factor ranges from $5\%n$ to $40\%n$, network size $n = 200$.

(c) Network lifetime vs CS factor, CS factor ranges from $5\%n$ to $40\%n$, network size $n = 200$.

Figure 4.4: Performance of non-CS, plain-CS, and hybrid CS based data gathering under different CS factors.

proposed algorithms are guaranteed. For instance, the recovery error is reduced to 10^{-5} when the CS factor increases beyond 50.

Next, Fig. 4.3a and Fig. 4.4a plot the total energy consumption for the proposed HCS-DGT-SPLS, plain CS based algorithm, non-CS based algorithm, and the MECDA based algorithm, respectively. Fig. 4.3a shows that, when the number of nodes in the network increases, the proposed HCS-DGT-SPLS scheme consumes significantly less energy compared with plain CS, non-CS based algorithm, and MECDA based algorithm. Specifically, when $n = 300, m = 30$, the proposed algorithm consumes only $\frac{1}{4}$, $\frac{2}{5}$, and $\frac{4}{9}$ energy of plain CS, non-CS, and MECDA, respectively. Fig. 4.4a shows that, when n is fixed at $n = 200$, the CS factor increases from 10 to 80, the total energy consumption for the proposed HCS-DGT-SPLS, plain CS, and MECDA increases while the total energy consumption for non-CS remains the same. The total energy consumption for the proposed algorithm is always lower than that of plain CS, non-CS, and MECDA when CS factor is less than $0.4n$. Plain CS based algorithm codes all the sensor readings into a length m vector, which creates more traffic load for the nodes near the leaf node, resulting in more total energy consumption than other algorithms. The proposed HCS-DGT-SPLS scheme triggers CS coding for a node only when its traffic load exceeds a threshold, leading to reduction of global transmissions and total energy consumption.

Then, Fig. 4.3b and Fig. 4.4b plot the total data gathering time slots for the proposed HCS-DGT-SPLS, plain CS-SPLS, non-CS-MWF, and the MECDA-SPLS, respectively. Results show that for small CS factor (eg. $m = 0.1n$), the proposed scheme achieves comparable time slots as non-CS-MWF algorithm and significantly less time slots compared with plain CS-SPLS and MECDA-SPLS algorithm. However, when CS factor increases beyond $0.25n$, data gathering time slots for the proposed scheme become more than the non-CS-MWF algorithm. The reason is that, in CS based algorithm, a node needs to wait for all its down-

stream nodes' measurements before compressing and transmitting its coded data. Whereas, nodes in non-CS algorithms can be scheduled to transmit regardless of the transmission order.

Finally, Fig. 4.3c and Fig. 4.4c plot the network lifetime for the proposed HCS-DGT-SPLS, plain CS, non-CS based algorithms, and MECDA algorithm, respectively. It can be observed that given the same initial energy, the proposed algorithm significantly prolongs network lifetime. The proposed HCS-DGT-SPLS shares the same network lifetime as plain CS because they have the same maximum traffic load and the same tree topology. And the network lifetime of the WSN is significantly longer than the non-CS based algorithm and MECDA. For instance, when network size scales from 50 nodes to 450 nodes, $m = 0.1n$, the proposed HCS-DGT-SPLS scheme prolongs network lifetime by $2\times \sim 8\times$ compared with non-CS based algorithm. When CS factor increases, the gap between the proposed algorithm and the non-CS based algorithm narrows but the proposed algorithm still possesses obvious advantage. Namely, when $n = 200$, $m = 80$, the network lifetime for CS-based algorithm is 317 rounds longer than non-CS based algorithm. This is due to the load balancing feature of the CS. The proposed algorithm dramatically reduces the maximum traffic load, resulting in extended network lifetime.

4.8 Concluding Remarks

This chapter presented an energy-efficient, low latency, and general data gathering algorithm based on hybrid CS with a dense measurement matrix. The problem was mathematically formulated as a multi-objective MINLP problem which was proved to be NP-hard. Two approximation algorithms were proposed to give practical solutions to the decomposed sub-problems. Simulation results have shown that the proposed algorithm achieves pronounced energy conservation for one round data gathering compared with benchmark algorithms. Moreover, the proposed

algorithm achieves lower data gathering latency compared with the benchmark algorithms. Finally, the proposed algorithm significantly extends network lifetime.

Chapter 5

Cost-Aware Stochastic Compressive Data Gathering for Wireless Sensor Networks

Chapter 4 have investigated applying CS technique in deterministic routing scenario. However, data gathering algorithms with deterministic path are not suitable for WSNs deployed in hostile environment and subjected to frequent topology changes. Furthermore, the repeated use of the same path will introduce load unbalance problem, which shortens network lifetime. In addition, deterministic path is more vulnerable to attacks. In such cases, it would be beneficial to allow certain randomness in the routing path. This chapter presents a cost-aware stochastic compressive data gathering for WSNs. In contrast to traditional CS based algorithms that implicitly assume uniform transmission cost, our proposed scheme takes the cost diversity into consideration of the stochastic CS data gathering framework. Markov chain based model is used to characterize the stochastic data gathering process. An optimization problem is formulated to minimize the total expected energy cost subjected to the constraints on global routing randomness and the recovery error. Our proposed algorithm requires less total expected energy cost to achieve certain level of recovery accuracy. It prolongs network lifetime due to its load balancing features. Extensive simulations on both synthetic and real data show that the proposed algorithm significantly outperforms benchmark algorithms.

5.1 Introduction

Reliable delivery of critical information is vital for successful operation of WSNs. However, the unreliable nature of wireless communication links in hostile environments poses a major challenge for reliable delivery of information to the sink nodes [101, 102]. Tree-based algorithms work well in small static networks. Nevertheless, as network size grows, the following problems surface. First, the tree-based data gathering inherently introduces the load unbalance problem. More specifically, nodes near to the sink node will have more traffic load since they have to forward packets for their downstream nodes. In addition, static tree-based algorithm cannot deal with changing environment and node failure. Furthermore, deterministic routing topology of tree-based algorithms makes it vulnerable to malicious attacks and eavesdropping.

In order to reduce data transmissions and achieve energy-efficiency, data correlation can be exploited. In real life, spatial and temporal correlation can be commonly found, such as in the environmental data [8]. Cooperative in-network compression algorithms such as [103] require information exchange for data compression, which introduces significant communication and computation overhead. DSC such as Slepian-Wolf coding [32, 104] reduces the computation complexity in the sensor nodes. However, it requires prior information about the data correlation structure and performs well only when the correlation structure is static. As we introduced in Chapter 2, CS promises to recover the data vector with high probability with less measurements than the data vector dimension. CS technique imposes light computation load for the sensor, the resource-demanding recovery process is performed at the powerful sink node. In addition, CS technique requires no prior information about the data correlation structure. Hence, it can deal with changing correlation patterns. Chapter 4 has proposed a method to apply CS technique in tree-based data gathering network. In this chapter, we investigate CS with

sparse measurement matrix to further reduce the number of transmissions in the network [51, 53, 55, 105].

An implicit assumption for CS is that the transmission cost for each node is the same. Plenty of previous works focus on reducing the number of transmission while achieving acceptable recovery accuracy. However, in practical sensor networks, the transmission cost may vary due to factors such as the link quality, distance, etc. The cost diversity should be considered in the CS framework in order to achieve cost-efficiency [61, 106]. Xu *et al.* [106] propose a cost-aware CS data gathering scheme for networked sensing systems. They aimed to achieve a good trade-off between the total sampling cost and the recovery accuracy. Regularized column sum (RCS) was used as a predictive metric for recovery accuracy. Chen *et al.* [61] proposed a cost-aware activity scheduling scheme for sleeping WSNs. They aimed to minimize the sampling cost with constraint on the regularized mutual coherence. Nevertheless, both papers considered a single-hop network.

For sensor nodes working in hostile environment where frequent topology changes exist, stochastic routing can be used together with CS technique in order to balance traffic load in the network. Stochastic routing refers to a routing process where a node will randomly choose a successor nodes as next hop with certain probability. By allowing certain randomness in routing, stochastic compressive data gathering brings the following advantages. First, the randomness brought by stochastic routing can promote load balancing in the network, which results in prolonged network lifetime. In addition, stochastic compressive data gathering is more robust to node failure. Furthermore, stochastic compressive data gathering is inherently more robust against malicious attack because the routing path and the data content is unpredictable.

This chapter develops a cost-aware stochastic compressive data gathering scheme for multi-hop WSNs. The objective is to collect enough random projections for reliable data recovery through stochastic compression with minimized total expected

energy cost. To achieve this, we formulate an optimization problem with the objective to minimize expected energy cost subjected to the constraints on recovery error and global routing randomness. We seek to construct an optimized transition probability matrix so that the equivalent measurement matrix constructed from the absorbing Markov chain can guarantee the recovery of data with certain error bound. Stochastic routing is used to deliver randomly selected compressed sensor readings to the sink node through several dynamic routes. This approach brings benefits such as load balancing and security. Nonetheless, the resulting equivalent CS measurement matrix will be a sparse non-uniform random matrix, which is different from traditional uniform sparse random CS measurement matrix whose recovery performance is guaranteed using ℓ_1 -minimization recovery. Hence, a recovery algorithm based on sketching decoder is proposed. More specifically, the contributions of this chapter can be described as the following,

- A cost-aware stochastic compressive data gathering scheme for multi-hop WSNs is developed.
- A Markov chain based model is exploited to characterize the stochastic compressive data gathering process and formulate an optimization problem.
- The formula to describe the relationship between the transition probability matrix of the Markov chain and the equivalent measurement matrix is formulated.
- An optimized equivalent sparse measurement matrix is constructed to minimize the total expected energy cost to collect the random projections for accurate recovery of the data vector.
- Extensive simulation results are provided to show that our proposed algorithm outperforms benchmark algorithms.

The remainder of the chapter is organized as follows. Section 5.2 sets up the

network model and describes the process of stochastic CS. In Section 5.3, the cost-aware stochastic compressive data gathering problem is formulated as a constrained optimization problem. Section 5.4 presents the procedure to solve for optimal compression probability. In Section 5.5, the low complexity sketching decoder to recover the CS data is presented. Section 5.6 evaluates the performance of the proposed algorithm through simulation. Finally, Section 5.7 concludes the chapter.

5.2 System Model

5.2.1 Network model

We consider a multi-hop WSN consisting of $n - 1$ nodes and a sink. The WSN can be modeled as a directed acyclic graph (DAG) [107], $G = (V, E)$, where V is the set of nodes in the network, E denotes the wireless links. Each node is assumed to know its geographical information and neighbors. Neighbors of node i include the predecessors set M_i and the successors set N_i . Successors set of node i can be defined as $N_i = \{j | d_{jn} \leq d_{in}, d_{ij} \leq r_0, j \in V\}$, where n is the sink node, d_{ij} is the Euclidean distance between node i and node j , r_0 is the radius of communication range. Similarly, the predecessors set of node i can be defined as $M_i = \{j | d_{jn} \geq d_{in}, d_{ij} \leq r_0, j \in V\}$. The outgoing traffic of node i will be transmitted to its successors with certain probability. Moreover, we assume that, there is at least one path from each node to the sink node. We define a cost matrix $\mathbf{C}$ with entries c_{ij} denotes the cost to transmit one packet on link (i, j) .

The sensor nodes are deployed around a sensing field to sense the environment and report the sensory readings to the sink node through multi-hop. The sensor reading for node i is denoted as u_i . The data vector collected by all the sensor nodes is denoted as $\mathbf{u} = [u_1, \dots, u_n]^T$. We assume that the data vector $\mathbf{u} \in \mathbb{R}^n$ is compressible in some transform domain Ψ . More specifically, there is an orthonormal basis Ψ , under which $\mathbf{u}$ can be represented as $\mathbf{u} = \Psi\theta$. Ψ can be

Fourier or wavelet transform basis [91]. The transform coefficient vector can be denoted as $\boldsymbol{\theta} = \boldsymbol{\Psi}^T \mathbf{u}$. Sort the magnitude of θ decreasingly, such that $|\theta|_{(1)} \geq |\theta|_{(2)} \geq \dots \geq |\theta|_{(n)}$. Then each of the i th coefficient decays according to power law [108] as follows,

$$|\theta|_{(i)} \leq D_0 i^{-\frac{1}{q}}, \quad i = 1, 2, \dots, n, \quad (5.1)$$

where D_0 is a constant, q controls the rate of decay of the transform coefficients.

The error for keeping the largest k coefficients and setting the remaining to zero is bounded by the following,

$$\|\mathbf{u} - \hat{\mathbf{u}}_k\|_2 = \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_k\|_2 \leq \alpha_q D_0 k^{-\frac{1}{q} + \frac{1}{2}}, \quad (5.2)$$

where α_q is a constant that only depends on q .

$\boldsymbol{\theta}$ is k -sparse if the largest k transform coefficients of $\boldsymbol{\theta}$ in magnitude capture most of the signal energy. In this case, one can say that $\mathbf{u}$ is k -sparse under transform basis $\boldsymbol{\Psi}$. Numerous previous research works [55, 109] have shown that, various natural data, such as temperature data vector for a smooth sensing field, are compressible.

5.2.2 Recovery of Compressible Data

CS theory states that, if a data vector $\mathbf{u}$ is k -sparse under a orthonormal transform basis $\boldsymbol{\Psi}$, then, by obtaining L random linear projections $\mathbf{y} = \boldsymbol{\Phi} \mathbf{u}$, where $\boldsymbol{\Phi}$ is a $L \times n$ measurement matrix, traditional CS theory guarantees that data vector $\mathbf{u}$ can be recovered from $\mathbf{y}$ with high probability [37]. Data recovery through ℓ_1 -minimization algorithm is introduced in Chapter 2. Dense random measurement matrices can guarantee the recovery accuracy using ℓ_1 -minimization algorithm. All the elements of a dense random matrix are nonzero.

On the other hand, sparse random projections rely on sparse random matrices

Figure 5.1: Stochastic compressive data gathering.

where only a few entries are nonzero. Some special sparse random matrices also satisfy RIP condition [93, 110], which guarantees data recovery using ℓ_1 -minimization. Whereas, the equivalent measurement matrix of the proposed stochastic compressive data gathering algorithm (which will be introduced later) is a non-uniform sparse random matrix, which doesn't satisfy the RIP condition. Therefore, a low complexity sketching decoder [55] with optimal error bound is used in our proposed algorithm for recovery.

5.2.3 Stochastic Compressive Data Gathering

In a dynamic environment, sensor nodes send their data to the sink node via stochastic routing. In contrast to tree-based shortest path routing, stochastic routing is more robust to node failure and can balance the energy load along the routing path.

The data gathering starts with an initiator node. From the initiator node, L independent random walks are performed towards the sink node. In each random walk, the initiator node randomly selects a successor from its neighbors with certain

probability. Then it transmits its compressed data to the selected node. The selected node compresses its own reading into the packet and transmits towards the sink node. Every random walk towards the sink node will produce one random projection. It can be observed that the random selection of the successor node depends only on the assigned transition probability and is not related to previous packet transmission. Moreover, all the compressed data packets are routed towards a single destination. Consequently, the stochastic routing of one random projection can be modeled as a discrete absorbing Markov chain [111], with a set of states $S = \{s_1, s_2, \dots, s_n\}$ and a transition probability matrix $\mathbf{P}$. The entry p_{ij} of $\mathbf{P}$ gives the transition probability from state s_i to s_j between time slot t and $(t+1)$. Node i is represented by the state s_i . Without loss of generality, we assume that the sink node n corresponds to the absorbing state s_n while other nodes correspond to transient states $s_1, s_2, \dots, s_{n-1}$. With this model, the transition probability p_{ij} specifies the probability that a packet is transmitted from node i to node j .

To be more specific, the stochastic compressive data gathering process can be described as follows. An initiator node initiates an independent random path to the sink node, resulting in one random projections at the sink node. Hence, the L independent compressive data gathering paths will produce L random projections. As shown in Fig. 5.1, the collection of one random projection corresponds to one random walk from the initiator node towards the sink node. For example, the random projection y_l received at the sink node is produced by the l -th random walk, which corresponds to the product of the l -th row of the measurement matrix and the data vector, $y_l = \Phi_l \mathbf{u}$, where Φ_l denotes the l -th row of the measurement matrix. In the following, we will describe how to obtain the l -th random projection y_l . We assume that the l -th random walk consists of the following nodes $\{\beta_1, \dots, \beta_{d_l}\}$, where d_l is the total length of the l th random walk, β_1 is the initiator node and β_{d_l} is the sink node.

To begin with, the initiator node β_1 generates a packet with the reading $y_l^1 =$

u_{β_1} . Then, one of its neighbors is randomly chosen as β_2 according to the absorbing Markov chain, and the generated packet is transmitted to β_2 .

Then, upon receiving the packet, β_2 generates a packet aggregating its own sensor reading with the received data by either adding or subtracting its own reading. $y_l^2 = y_l^1 + u_{\beta_2}$ or $y_l^2 = y_l^1 - u_{\beta_2}$. Then, y_l^2 is transmitted to a randomly selected neighbors.

Accordingly, for the i -th node of path l , β_i , it generates the packet with aggregated data $y_l^i = y_l^{i-1} \pm u_{\beta_i}$.

The packet will be aggregated until it reaches the sink node. This will produce the l -th linear random projection $y_l = y_l^{d_l}$. L random walks result in L random projections. Once the sink node obtains all the L random projections $\mathbf{y} = [y_1, \dots, y_L]^T$, it can recover the data vector with the recovery algorithm, which will be introduced later in Section 5.5.

The long term behavior of the random walk determines the compression probability of each node, which also affects the recovery quality. We define the compression probability π_j as the probability that the sensor reading of node j is aggregated to form the random projection. In other words, compression probability is the probability that the j -th entry of the measurement matrix is nonzero.

5.2.4 Equivalent Measurement Matrix and Compression Probability

As discussed above, given an initiator node, the long term behavior of the corresponding absorbing Markov chain determines the compression probability of each node. Therefore, given the L random walks to the sink node, we can construct the

equivalent measurement matrix as the following.

$$\phi_{lj} = \begin{cases} +1 & \text{with prob. } \frac{\pi_j}{2}, \\ 0 & \text{with prob. } 1 - \pi_j, \\ -1 & \text{with prob. } \frac{\pi_j}{2}, \end{cases} \quad (5.3)$$

where $l = 1, \dots, L$ corresponds to the number of random walks. The l -th random projection is generated by $\sum_{j=1}^n \phi_{lj} u_j$. The probability that $\phi_{lj} \neq 0$ is defined as the compression probability π_j , which is related to the frequency of node j in the absorbing Markov Chain. We assume that node 1 is the initiator node. Starting from node 1, L independent random walks to the sink node will produce L random projections. Then the sink node recovers the original data from the random projections with a data recovery algorithm based on sketching decoder.

Since the compression probability relates to the long term behavior of the absorbing Markov chain, in order to derive the compression probability, the transition probability matrix should be written in the following canonical form [111],

$$\mathbf{P} = \begin{bmatrix} \mathbf{Q} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{bmatrix},$$

where $\mathbf{Q}$ is an $(n-1) \times (n-1)$ matrix representing the transition probabilities of the transient nodes, $\mathbf{r}$ is an $(n-1) \times 1$ matrix representing the transition probabilities of the links terminating at the sink node and $\mathbf{0}$ is an $(n-1) \times 1$ column vector with all zero entries.

Given the transition probability matrix $\mathbf{P}$, the *fundamental matrix* of $\mathbf{P}$ can be computed as $\mathbf{F} = (\mathbf{I} - \mathbf{Q})^{-1}$. The entry f_{ij} of $\mathbf{F}$ denotes the expected number of times the process is in transient state s_j if it is initiated in state s_i . In other words, f_{ij} denotes the average occurrence of node j if the initiator node is i . Assume node 1 is the initiator node. Then, if started from node 1, the expected number of occurrence of node j is f_{1j} . Then the probability of compression π_j can be given

by the following formula,

$$\pi_j = f_{1j}. \quad (5.4)$$

The above equation (5.4) establishes the relationship between the transition probability of the Markov chain and the equivalent measurement matrix of the compressive data gathering. π_j controls the sparsity of the equivalent measurement matrix, larger π_j indicates denser measurement matrix and possibly more energy consumption.

5.2.5 Expected Energy Cost

The energy cost is a key performance metric for a data gathering WSN. In deterministic routing, the route to deliver the data to the sink node with the least energy cost can be found by shortest path algorithm. In the context of stochastic routing, where all the available paths between the source and the sink node are utilized, the expected energy cost is used as a performance metric.

The expected energy cost to transmit a packet from any initiator node i to the sink node is denoted as

$$w_i = \begin{cases} \sum_{j \in N_i} p_{ij}(c_{ij} + w_j) & \text{if } i \neq n \\ 0 & \text{if } i = n \end{cases} \quad (5.5)$$

In matrix form, the expected energy cost vector in (5.5) can be written as

$$\mathbf{w} = \mathbf{diag}(\mathbf{P}\mathbf{C}^T) + \mathbf{P}\mathbf{w}, \quad (5.6)$$

where $\mathbf{w} = [w_1, w_2, \dots, w_{n-1}, 0]^T$ is the expected energy cost vector.

5.2.6 The Randomness of the Data Gathering Process

A deterministic shortest path to the sink node will result in the minimum cost. However, the deterministic routing path is vulnerable to malicious attacks and cannot deal with changing environment. Introducing stochastic routing can improve the robustness of the network. On the other hand, the randomness of the data gathering process affects the total expected energy cost. The total expected energy cost increases as the randomness increases. Shannon entropy [112] is exploited in this chapter to measure the randomness of the data gathering process. The Shannon entropy of a node is used to measure the local randomness of the next hop choice for a node. For instance, the Shannon entropy of the transition probabilities can be given as

$$h_i = - \sum_{j \in N_i} p_{ij} \log p_{ij}, \quad \forall i \neq n. \quad (5.7)$$

Moreover, the global entropy H is defined as the sum of weighted individual values as

$$H = \sum_{i=1}^{n-1} f_{1i} h_i = - \sum_{i=1}^{n-1} f_{1i} \sum_{j \in N_i} p_{ij} \log p_{ij}, \quad (5.8)$$

where each individual entropy is weighted by the expected number of visits to this node. The global entropy H measures the global routing randomness. For better readability, in the following, f_{1i} is denoted as f_i since the initiator node is always 1.

5.3 Problem Formulation

By modeling the stochastic compressive routing as an absorbing Markov chain, the expected energy cost vector is given by (5.5).

With sparse CS technique, $L \ll n$ random walks to the sink node are sufficient for data recovery. This chapter aims to derive the optimal transition probability

matrix that minimizes the total expected energy cost to collect L compressed measurement at the sink node subjected to the constraint on the bound on CS recovery error and the global routing randomness.

Node 1 is selected as the initiator node, then w_1 is the expected energy cost to transmit a coded packet from node 1 towards the sink node. It takes L random walks to produce L random projections. Consequently, the total expected energy cost to collect L random projections at the sink node is given by Lw_1 . Then, the problem can be formulated as the following.

$$\underset{\mathbf{P}}{\text{minimize}} \quad Lw_1 \quad (5.9)$$

$$\text{subject to} \quad -\sum_{i=1}^{n-1} f_i \sum_{j \in N_i} p_{ij} \log p_{ij} = H_0, \quad (5.10)$$

$$L = O(k^2 \log n), \quad (5.11)$$

$$\mathbf{w} = (\mathbf{I} - \mathbf{P})^{-1} \mathbf{diag}(\mathbf{P}\mathbf{C}^T), \quad (5.12)$$

$$0 \leq p_{ij} \leq 1, \forall (i, j) \in E, \quad (5.13)$$

$$\sum_{j \in N_i} p_{ij} = 1, \quad \forall i \neq n, \quad (5.14)$$

where constraint (5.11) is the global routing randomness constraint, constraint (5.12) specifies the number of random walks to guarantee data recovery, constraint (5.13) establishes the relationship between the expected energy cost of one random walk and the transition probability, constraint (5.15) states that the summation of the probability to neighbors is 1. We solve the above optimization problem for the optimal transition probability, then the compression probability can be obtained by equation (5.4).

The framework of our cost-aware stochastic compressive data gathering can be described by the flow chart shown in Fig. 5.2.

Figure 5.2: The solution framework of cost-aware stochastic compressive data gathering.

5.4 Optimal Compression Probability

This section derives the formula to compute the optimal probability to minimize the total expected data gathering cost by solving the optimization problem. With the optimal transition probability, the optimal compression probability can be computed with equation (5.4). The optimization problem is solved with Lagrange multiplier method. To begin with, the Lagrangian for the optimization problem can be obtained as the following.

$$\begin{aligned}
 \mathcal{L} = & Lw_1 + \sum_{i=1}^{n-1} \lambda_i [w_i - \sum_{j \in N_i} p_{ij} (c_{ij} + w_j)] + \lambda_n (w_n) \\
 & + \sum_{i=1}^{n-1} \mu_i (\sum_{j \in N_i} p_{ij} - 1) + \eta [\sum_{i=1}^{n-1} f_i \sum_{j \in N_i} p_{ij} \log p_{ij} + H_0] \\
 & + \rho (L - k^2 \log n),
 \end{aligned}$$

where λ_i, η and ρ are the Lagrangian multipliers.

Setting $\frac{\partial \mathcal{L}}{\partial p_{kl}} = 0$, we obtain

$$\begin{aligned}
 & -\lambda_k (c_{kl} + w_l) + \mu_k + \eta f_k (\log p_{kl} + 1) \\
 & + \eta \sum_{i=1}^{n-1} \left(\frac{\partial f_i}{\partial p_{kl}} \right) \sum_{j \in N_i} p_{ij} \log p_{ij} = 0.
 \end{aligned}$$

Substituting $h_i = -\sum_{j \in N_i} p_{ij} \log p_{ij}$, the above equation can be simplified into

$$p_{kl} = \exp \left[\frac{\lambda_k (c_{kl} + w_l)}{\eta f_k} - \frac{\mu_k + f_k}{\eta f_k} + \frac{\eta \sum_{i=1}^{n-1} \left(\frac{\partial f_i}{\partial p_{kl}} \right) h_i}{\eta f_k} \right]. \quad (5.15)$$

Use the fact that the summation of the probability to neighbors is 1, $\sum_{l' \in N_k} p_{kl'} = 1$, with some manipulation, we obtain

$$\begin{aligned} & \exp \left[\frac{-\mu_k - \eta f_k}{\eta f_k} \right] \\ &= \frac{1}{\sum_{l' \in N_k} \exp \left[\frac{\lambda_k (c_{kl'} + w_{l'}) + \eta \sum_{i=1}^{n-1} \left(\frac{\partial f_i}{\partial p_{kl'}} \right) h_i}{\eta f_k} \right]}. \end{aligned} \quad (5.16)$$

Substitute equation (5.16) into (5.15), we can have the update equation for probability as the following,

$$p_{kl} = \frac{\exp \left[\frac{\lambda_k (c_{kl} + w_l) + \frac{1}{f_k} \sum_{i=1}^{n-1} \left(\frac{\partial f_i}{\partial p_{kl}} \right) h_i}{\eta f_k} \right]}{\sum_{l' \in N_k} \exp \left[\frac{\lambda_k (c_{kl'} + w_{l'}) + \eta \sum_{i=1}^{n-1} \left(\frac{\partial f_{1i}}{\partial p_{kl'}} \right) h_i}{\eta f_{1k}} \right]}. \quad (5.17)$$

Then, with $\frac{\partial \mathcal{L}}{\partial w_k} = 0$, we obtain

$$\lambda_k - \sum_{i \in M_k} \lambda_i p_{ik} = 0, \text{ for } k \neq 1, \quad (5.18)$$

where M_k is the set of predecessors of node k . Since the initiator node has no predecessors, we obtain $\lambda_1 = -1$. Defining $\tau_k = -\lambda_k$, then we have

$$\tau_k = \begin{cases} \sum_{i \in M_k} \tau_i p_{ik} & \text{if } k \neq 1 \\ 1 & \text{if } k = 1 \end{cases} \quad (5.19)$$

Defining $\zeta_l = \frac{1}{f_k} \sum_{i=1}^{n-1} \left(\frac{\partial f_i}{\partial p_{kl}} \right) h_i$, the optimal transition probability from node k to node l is given by

$$p_{kl} = \frac{\exp \left[\frac{-\tau_k (c_{kl} + w_l) + \zeta_l}{\eta f_k} \right]}{\sum_{l' \in N_k} \exp \left[\frac{-\tau_k (c_{kl'} + w_{l'}) + \zeta_{l'}}{\eta f_k} \right]}. \quad (5.20)$$

Parameter $\eta > 0$ is an indicator of global routing randomness, which is a monotonically increasing function of H , larger η indicates larger global routing

randomness. Given η , the optimal transition probability can be found with iterative method.

After obtaining the optimal transition probability, the optimal compression probability for node j is given by $\pi_j = f_{1j}$, which corresponds to the occurrence frequency of node j when the initiator node is node 1. Algorithm 5 summarizes the procedure of solving optimal compression probability.

Algorithm 5: Solve for optimal compression probability

Input : Graph $G = (V, E)$, initiator node = 1, sink node = n , cost matrix $\mathbf{C}$, randomness parameter η
Output: Optimal compression probability $\boldsymbol{\pi}^*$

- 1 Initialize the transition probability as pure random, $p_{ij} = \frac{1}{deg(i)}$, where $deg(i)$ is the outdegree of node i ;
- 2 **repeat**
- 3 Compute h_i using equation (5.7);
- 4 Solve $\mathbf{w} = \mathbf{diag}(\mathbf{P}\mathbf{C}^T) + \mathbf{P}\mathbf{w}$;
- 5 Compute global entropy H ;
- 6 Compute ζ_j for each node j ;
- 7 **for** $j \in N_i$ **do**
- 8 Update p_{ij} with the equation (5.20) ;
- 9 **end**
- 10 **until** P converges;
- 11 Compute the corresponding fundamental matrix of $\mathbf{P}$, $\mathbf{F} = (\mathbf{I} - \mathbf{Q})^{-1}$;
- 12 $\pi_j = f_{1j}$

5.5 Data Recovery Using Sketching Decoder

According to [53, 55], with the measurement matrix $\boldsymbol{\Phi}$, transform matrix $\boldsymbol{\Psi}$, and the linear random projections $\mathbf{y}$, the transform coefficients vector can be recovered and recorded as $\hat{\boldsymbol{\theta}} = [\hat{\theta}_1, \dots, \hat{\theta}_n]^T$. Then, the data vector can be recovered as $\hat{\mathbf{u}} = (\hat{\boldsymbol{\theta}}\boldsymbol{\Psi}^{-1})^T$. The detailed low complexity sketching decoder is described as in Algorithm 6.

Some previous study has shown that, with $L = O(k^2 \log n)$ random projections, the recovery algorithm can achieve error bound that is comparable to k

Algorithm 6: Sketching decoder for non-uniform sparse random projection recovery.

Input : Transform basis matrix Ψ , $L_1 = C_1 \frac{1+\xi+\chi H^2}{\epsilon^2}$ and $L_2 = C_2(1 + \gamma) \log n$ and $L = L_1 L_2$

Output: Recovered data vector $\hat{\mathbf{u}}$

- 1 Construct measurement matrix Φ with optimized compression probability ;
- 2 Divide the measurement matrix Φ into L_2 sub-matrices, $\{\Phi_1, \Phi_2, \dots, \Phi_{L_2}\}$, each of size $L_1 \times n$. ;
- 3 Divide transform matrix Ψ into n column vector $\{\psi_1, \dots, \psi_n\}$, where $\psi_j \in \mathbb{R}^n$ is the j -th column of the original transform matrix. ;
- 4 Divide the random projection of the data vector into L_2 sub-matrices, $\{\mathbf{a}_1 = \frac{1}{\sqrt{L_1}} \Phi_1 \mathbf{u}, \dots, \mathbf{a}_{L_2} = \frac{1}{\sqrt{L_1}} \Phi_{L_2} \mathbf{u}\}$, each of size $L_1 \times 1$. ;
- 5 **for** $j = 1 : n$ **do**
- 6 **for** $l = 1 : L_2$ **do**
- 7 Project the basis vector to obtain $\mathbf{b}_l^j = \Phi_l \psi_j$;
- 8 Compute the inner product of $\mathbf{a}_l$ and $\mathbf{b}_l^j$, obtaining $z_l^j = (\mathbf{a}_l)^T \mathbf{b}_l^j$;
- 9 **end**
- 10 Compute estimate of the j -th element of the transformation coefficients, $\hat{\theta}_j = \text{median}\{z_1^j, \dots, z_{L_2}^j\}$. ;
- 11 **end**
- 12 Keep the largest k coefficients in $\hat{\theta}$, and set the remaining to zero ;
- 13 The data vector can be recovered as $\hat{\mathbf{u}} = (\hat{\theta} \Psi^{-1})^T$;

best approximation [53, 55]. More specifically, the recovery error bound of sparse compressive data gathering is described by the Theorem 5.1 as the following.

Theorem 5.1. *Suppose data vector $\mathbf{u} \in \mathbb{R}^{n \times 1}$ satisfies $\frac{\|\mathbf{u}\|_\infty}{\|\mathbf{u}\|_2} \leq M$. A sparse random measurement matrix $\Phi \in \mathbb{R}^{L \times n}$ with probability distribution π , and $L = B(\delta, \epsilon)k^2 \log n$. Let $\mathbf{y} = \frac{1}{\sqrt{L}}\Phi\mathbf{u}$, consider an orthonormal transform $\Psi \in \mathbb{R}^{n \times n}$ and the corresponding transform coefficients $\theta = \Psi^T\mathbf{u}$. If the k largest transform coefficients in magnitude give an approximation with error $\|\mathbf{u} - \hat{\mathbf{u}}_{opt}\|_2^2 \leq \delta\|\mathbf{u}\|_2^2$, then given only $\mathbf{y}$, Φ and Ψ , the sparse stochastic data gathering gives an estimate $\hat{\mathbf{u}}$ with error*

$$\|\mathbf{u} - \hat{\mathbf{u}}\|_2^2 \leq (1 + \epsilon) \delta \|\mathbf{u}\|_2^2, \quad (5.21)$$

with probability at least $1 - n^{-\gamma}$. $\gamma > 0$ is a constant.

The proof of theorem 5.1 can be found in [53].

Theorem 5.1 states that, with the given L random projections, the recovery error of proposed stochastic compressive data gathering can achieve an error which is only $(1+\epsilon)$ times of the optimal error bound for a k -sparse data vector. Moreover, π_j controls the sparsity of the equivalent measurement matrix, larger π_j indicates denser measurement matrix and possibly more energy consumption. Whereas, larger π_j also indicates smaller L .

5.6 Performance Evaluation

In this section, the performance of the proposed scheme is evaluated on both synthetic data and real data. For synthetic data, the sparsity of the data vector can be controlled. A DAG with $n = 200$ nodes is generated, where each connected link (i, j) is associated with a normalized cost c_{ij} ranges from 0 to 1. Then, a corresponding k -sparse synthetic data vector is generated. For example, Fig. 5.3 shows a synthetic data vector for a network of 200 nodes, which is 4-sparse on the frequency domain. In our evaluation, the real data vector is the temperature data imported from Intel Lab [113].

The performance of the proposed algorithm is evaluated from three aspects:

Figure 5.3: Synthetic data vector and the corresponding transform coefficients using Fourier basis.

the recovery error, the total expected energy cost and the network lifetime. The recovery error is defined as

$$e = \frac{\|\mathbf{u} - \hat{\mathbf{u}}\|_2^2}{\|\mathbf{u}\|_2^2}, \quad (5.22)$$

where $\mathbf{u}$ is the original data vector, and $\hat{\mathbf{u}}$ is the recovered data vector. Assume that each node has a certain amount of initiate energy. The network lifetime is defined as the time duration until the first node runs out of energy, which is measured in time slots.

The random walk based routing algorithm in [114] is chosen as the benchmark. In the benchmark algorithm, each node assumes that all the links have the same energy cost, hence, it transmits to its neighbors with equal probability.

First of all, the number of random projections required to achieve different recovery error under different degree of global routing randomness is evaluated.

(a) The trade-off between number of random projections and the recovery error. Synthetic data, network size $n = 200$.

(b) The trade-off between total expected energy cost and the recovery error. Synthetic data, network size $n = 200$.

Figure 5.4: Performance of proposed algorithm on synthetic data.

(a) The trade-off between number of random projections and the recovery error. Real data, DCT transform, network size $n = 54$.

(b) The trade-off between total expected energy cost and the recovery error. Real data, DCT transform, network size $n = 54$.

Figure 5.5: Performance of proposed algorithm on real data.

Fig. 5.4a and Fig. 5.5a plot the number of random projections to achieve a certain level of accuracy for synthetic data and real data, respectively. For both our proposed and benchmark algorithm, the recovery error reduces as the number of random projections increases. Smaller value of global routing randomness results in larger number of random projections. For example, in Fig. 5.4a, when there are 200 nodes in the network, to achieve approximately 0.4 recovery error, it takes 140, 80, and 60 random projections for the proposed algorithms with global routing randomness of $H = 4.29$, $H = 5.43$ and $H = 7.31$. And it takes 80 random projections for the benchmark algorithm to achieve the same level of accuracy. This may be due to the fact that the equivalent measurement matrix of an absorbing Markov chain with smaller global routing randomness is more sparse, which requires more random projections for data recovery.

Fig. 5.4b and Fig. 5.5b investigate the total expected energy cost for stochastic compressive data gathering to achieve a certain level of accuracy under different global routing randomness. As shown in both Fig. 5.4b and Fig. 5.5b, for both synthetic data and real data, for a given degree of global routing randomness, in order to achieve lower recovery error, the total expected energy cost is increased. On the other hand, in order to achieve the same recovery error, our proposed cost-aware algorithm requires less energy cost than the benchmark algorithm. For example, in Fig. 5.4b, to achieve recovery error less than 0.2, the total expected energy cost for the proposed algorithm is about 750, while it requires about 1250 for the benchmark algorithm to achieve the same level of recovery accuracy. The proposed algorithm minimizes the total expected energy cost for a certain degree of global routing randomness. Therefore, although when $H = 4.29$, the proposed algorithm requires more random projections than the benchmark algorithm to achieve recovery error of 0.2, the corresponding total expected energy cost of $H = 4.29$ is less than the benchmark algorithm. When cost diversity exists, the total expected energy cost gives more accurate measure of the total cost to collect and

recover the data using stochastic CS.

Figure 5.6: Network lifetime vs the recovery error. Synthetic data, network size $n = 200$.

Fig. 5.6 and Fig. 5.7 plot the network lifetime as a function of the recovery error for both synthetic data and real data. It can be observed from the figures that, for a given algorithm, in order to achieve smaller recovery error, network lifetime will decrease as the target recovery error reduces. Moreover, for the same level of recovery error, algorithms with larger degree of global routing randomness achieve longer network lifetime. For instance, in Fig. 5.6, when recovery error is 0.4, the network lifetime values for $H = 4.29, 5.40, 7.31$ and the benchmark are 170, 274, 400 and 289. The reason is that larger degree of global routing randomness indicates that the traffic load in the network is distributed more evenly. Moreover, the proposed algorithm minimizes the total expected energy cost subjected to the constraint on recovery error and global routing randomness. Therefore, when $H = 7.31$, the proposed algorithm achieves a good trade-off between the recovery error and network lifetime. We design the process of stochastic CS data gathering such that the equivalent sparse random measurement matrix is optimized. Extensive

Figure 5.7: Network lifetime vs the recovery error. Real data, DCT transform, network size $n = 54$.

simulations on both synthetic data and real data show that our proposed algorithm minimizes the total expected energy cost and prolongs network lifetime.

5.7 Concluding Remarks

In this chapter, a cost-aware stochastic compressive data gathering scheme was proposed for dynamic multi-hop WSNs. A Markov chain based method was proposed to analyze the relationship between the total expected energy cost and recovery error to collect and recover the data vector. An optimization problem to minimize the total expected energy cost for data gathering subjected to the recovery error and global routing randomness constraints was formulated. Simulation results have shown that the proposed algorithm prolongs network lifetime, and achieves a good trade-off between total expected energy cost and the recovery error.

Chapter 6

Harvesting-Aware Compressed Data Gathering for Optimal Sustainable Information Throughput in EH-WSNs

In this thesis, previous chapters have shown that data aggregation can prolong network lifetime for traditional WSNs powered by batteries. Traditional WSNs are powered by tiny, non-rechargeable batteries, which limits their network lifetime. Hence, energy conservation is the crucial concern in order to prolong network lifetime. Even so, the sensor nodes will eventually die out once their energy is depleted. Further, there are applications where the sensor networks are required to operate for decades while replacing the battery is impractical. Recent advances in EH techniques provide a possible alternative to power the sensor nodes, which may unlock the possibility of perpetual operation of the WSNs. In EH-WSNs, network lifetime is no longer a critical concern. Instead, given a certain EH rate, it's crucial to adapt the design of the algorithm to achieve sustainable operation while maximizing information throughput. This chapter applies CS techniques into EH-WSNs to improve the information throughput while achieving energy neutrality. An optimization problem is formulated to maximize the throughput. Numerical results are provided to show that the proposed CS based algorithm significantly improves the sustainable information throughput for EH-WSNs. Furthermore,

our proposed algorithm requires smaller EH material size to achieve the same information throughput.

6.1 Introduction

Traditional sensor nodes are powered by limited-sized batteries which are hard to replace after deployment. Hence, conventional data gathering algorithms focus on conserving energy and extending network lifetime. However, there are applications like structural health monitoring that requires the WSNs to operate for longer period, such as years or even decades. In addition, battery leakage will drain out the battery even when they are not in use [115]. Due to the above reasons, EH has emerged to be a promising alternative to traditional non-rechargeable batteries.

EH is to acquire renewable energy from ambient environment and convert to electricity to power the devices [65]. Common ambient energy sources can be: solar energy, indoor lighting, RF signal, and vibration [66,67]. Each of the energy sources associates with different technologies. The benefits of powering the sensor nodes with EH techniques are manifold. First, EH techniques provide the node with capabilities such as autonomous operation, which significantly reduces the cost of maintenance. Moreover, the network lifetime is extended or even sustainable operation can be achieved. While EH techniques have the potential to enable perpetual operation, the instantaneous harvested energy is limited by parameters such as size, weight, and cost. For example, the energy storage system for renewable energy in sensor nodes is comparatively small, which leads to sporadic energy supply. The harvested energy should be wisely managed such that information can be successfully delivered while perpetual operation of WSNs can be achieved.

With renewable energy sources, network lifetime is no longer a critical concern when designing the communication protocol. Instead, given source-dependent energy availability, the objective becomes achieving sustainable operation at a satisfy-

ing performance level. The change in objective calls for novel communication protocols that are tailored for EH-WSNs. Several algorithms for EH-WSNs have been proposed to improve various performance requirements [116–118]. Lattanzi *et al.* proposed a new method to assess the energetic sustainability of routing algorithms for EH-WSNs [116]. The authors defined the maximum energetically sustainable workload (MESW) for a given EH-WSN as a metrics to indicate whether a routing algorithm allows for sustainable operation. A tool based on network simulation and graph algorithms was developed to test the optimality of the sustained work load of a routing protocol. Jakobsen *et al.* proposed a distributed energy harvesting aware routing algorithm (DEHAR) for EH-WSNs to find energy optimized routes [117]. In DEHAR, each sensor node finds a sustainable path to the sink node based on the hop count and the energy availability. The authors showed that DEHAR can adapt to changes in harvested and stored energy. Eu *et al.* proposed a probabilistic polling protocol for EH-WSN to improve the performance in terms of throughput, fairness, and scalability [118].

Information throughput is an important performance metric in traditional WSNs powered by non-rechargeable batteries. It measures the amount of information collected at the sink, subjected to energy constraints [11, 119]. In our proposed work, the information throughput is defined as the ratio of the total amount of received information at the sink node and the system delay. In the context of EH-WSNs, the objective would be changed to maximizing information throughput while achieving sustainable operation of the sensor network. Given the same amount of harvested energy, if the sensory information can be compressed into less packets for transmission, then the information throughput for the sustainable system can be improved. CS technique can condense the raw information into less data packets while guaranteeing the recovery quality at the sink node [37]. Therefore, CS technique can be exploited in EH-WSN to improve information throughput while maintaining sustainable operation.

This chapter introduces CS technique into EH-WSNs to improve the information throughput for the system while satisfying energy neutrality. An EH-WSN is considered where sensors are powered solely by renewable energy. The sink node is connected to the power plug with abundant power supply. More specifically, the main contributions of this chapter can be summarized as the following:

- A harvesting-aware compressed data gathering algorithm is proposed. The heterogeneous EH rates of the nodes are taken into consideration in order to maximize information throughput while maintaining sustainable operation.
- An optimization problem is formulated with the objective to maximize the information throughput subjected to the constraints on energy neutrality and compressive data scheduling.
- The NP-hard optimization problem is divided into two sub-problems. Two approximation algorithms are proposed to solve the problem.
- The performance bounds for the two approximation algorithms are given.
- The performance of our proposed scheme is evaluated through simulation, compared with several state-of-the-art methods. Simulation results show that the proposed algorithm outperforms benchmark algorithms.

The remainder of the chapter is organized as follows. Section 6.2 establishes the framework of harvesting-aware compressed data gathering. In Section 6.3, the optimization problem is formulated as a MINLP problem. Section 6.4 analyzes the complexity of our problem and proves its NP-hardness. In Section 6.5, the NP-hard problem is decomposed into two sub-problems, and approximation algorithms are developed to solve them. In Section 6.6, the performance bounds on the tree construction algorithm and scheduling algorithm are given. Section 6.7 evaluates the performance of our proposed algorithm. Section 6.8 concludes the chapter.

6.2 Harvesting-Aware Compressed Data Gathering Framework

6.2.1 Network Model

Assume that 1 sink node and $n - 1$ half-duplex sensor nodes are randomly deployed in a sensing field. We assume that the sink node has abundant power supply and better computation capability. Assume that other sensor nodes are powered by renewable energy with limited storage. The energy storage is subjected to energy half-duplex constraint. That is, the sensor node cannot be charged and discharged simultaneously [120]. Even with the state-of-the-art energy harvester, the EH rate could be much lower than typical power consumption level of a sensor node [118]. Hence, a sensor node has to accumulate energy in the energy storage until it reaches the level to power the transmission of a sensor nodes. Accordingly, three states are defined for the sensor nodes: the harvesting state, the active state and the idle state. In the harvesting state, the sensor node harvests and stores energy without performing other activity. In the active stage, the node has accumulated enough energy, it can receive, process and transmit data. In the idle state, the sensor node doesn't perform any activity. As shown in Fig. 6.1, the sensor node enters harvesting state to accumulate energy until it obtains enough for data transmission, then it enters active state to perform computation and transmission. Between the active states, when the sensor node needs to wait for other node's data, it enters the idle state. Once it depletes energy, it switches to harvesting state. An average EH rate λ_i is defined for each node such that $\lambda_i t_i = \int_0^{t_i} \hat{\lambda}_i(t) dt$, where t_i is the time for node i to accumulate energy, $\hat{\lambda}_i$ denotes the instantaneous harvesting rate for node i . Heterogeneous average EH rates are assumed for all the sensor nodes.

A sensor node i can communicate to node j that resides within its communication range r_0 . Protocol model/graph-based model [121] is used to define interference. In this model, the interference range of a node is assumed to be equal to its

Figure 6.1: Two states of an EH node.

transmission range. Accordingly, two links interfere with each other if the receiver of either link locates within the range of the transmitter of the other link. In this case, these two links cannot be scheduled concurrently. The above network can be modeled as a connected graph $G = (V, E)$, where V is the set of nodes that are randomly deployed in certain region, E denotes the set of wireless links between sensor nodes in the network. $\mathbf{C}$ is an $n \times n$ cost matrix defined on edges of the network with entry $c_{ij} : (i, j) \in E$ denoting the energy cost to transmit one data packet from node i to node j .

6.2.2 Energy Neutrality and Cost-Harvest Ratio (CHR)

In order to achieve sustainable operation, the harvested energy should be more than what to be consumed, which can be expressed as

$$\lambda_i \sigma_i t_i \geq E_c^i, \quad (6.1)$$

where λ_i is the average EH rate for node i , σ_i is the EH material size for node i , E_c^i is the total energy consumption for node i . E_c^i can be computed using the energy consumption model proposed in [18]. Assume that one packet includes L bits. The energy cost to transmit a packet from node i to node j can be computed as $E_{tx} = E_{elec}L + \epsilon_{amp}Ld_{ij}^2$, where E_{elec} is the energy used to run transmitter or receiver circuitry, ϵ_{amp} is the energy used to achieve acceptable $\frac{E_b}{N_0}$, the energy per

bit to noise power spectral density ratio. The energy cost of receiving one packet is $E_{rx} = E_{elec}L$. Overall, define $c_{ij} = E_{tx} + E_{rx}$, the total energy consumption for a node can be computed as

$$E_c^i = f_{ij}c_{ij}, \quad (6.2)$$

where f_{ij} is the number of packets transmitted from node i to node j .

Moreover, we define the CHR as

$$\varpi_{ij} = \frac{c_{ij}}{\lambda_i \sigma_i}, \quad (6.3)$$

which serves as a routing metric to incorporate the influence of both EH and energy consumption.

6.2.3 Sustainable Information Throughput

Sustainable information throughput measures the average amount of information collected at the sink node per unit time subjected to energy neutrality constraints. A network of n sensor nodes will produce n units of information to be delivered to the sink node. The information throughput ρ is defined as,

$$\rho = \frac{n}{\Gamma}, \quad (6.4)$$

where Γ is the system delay. The system delay consists of two parts: the time to harvest energy for the whole network, denoted as Γ_e , and the time to transmit data to the sink node, denoted as Γ_t . In order to fuel continuous data transmission, each sensor node should spend a period of time to harvest and store enough energy. The amount of energy to be harvested determines the EH time. Assume that all the nodes are synchronized. All the nodes start to harvest energy at the same time. Then the EH time is determined by the node with the largest EH time. Hence, $\Gamma_e = \max_i \{t_i\}$.

The information amount is fixed. With CS technique, the same amount of

information can be delivered using less packets, which leads to reduced system delay. For example, as shown in Fig. 6.2, in the CS-base scheme, the 6 units of information is delivered to the sink node with 12 packets, while in Non-CS cases, the same amount of information is delivered in 13 packets.

6.2.4 Harvesting-Aware Compressed Data Gathering and Recovery

The operation of the network can be described as follows. The sensor nodes are randomly deployed in a sensing field. The sink node runs the proposed algorithm to compute a harvesting-aware routing topology and a scheduling scheme. Then, the topology and scheduling scheme will be disseminated to all the sensor nodes. The sensor nodes first enter harvesting state, and harvest ambient energy at various EH rates. The harvested energy is stored for later use. Then, the sensor nodes enter active state. Each sensor node senses and produces a sensory reading, u_i . To facilitate compressive data gathering, node i sends out a vector $u_i\Phi_i$ instead of the original data for aggregation. The vector will be aggregated along the way as shown in Fig. 6.2. The sink node receives $\mathbf{y} = \sum_{i=1}^n u_i\Phi_i$ and recovers the original data vector with CS recovery algorithms.

6.2.5 CS Recovery

CS theory states that if an $n \times 1$ data vector is compressible in some domain, then the data vector can be recovered from far less observations than the original signal dimension [37]. Assume a data vector of length n can be represented as $\mathbf{u} = [u_1, \dots, u_n]^T$. If the data vector is sparse, which means that we can find a k -sparse representation for $\mathbf{u}$ under a proper basis Ψ such that $\mathbf{u} = \Psi\boldsymbol{\theta}$, where Ψ is an $n \times n$ transform matrix, $\boldsymbol{\theta}$ is a k -sparse column vector, only k out of n elements of $\boldsymbol{\theta}$ are nonzero, and $k \ll n$. Then, according to CS theory [37], by obtaining $\mathbf{y} = \Phi\mathbf{u}$, a column vector of length m , the sink node can reconstruct the

(a) CS-based data gathering, CS factor $m = 2$, node 0 is the sink node.

(b) Non-CS data gathering, node 0 is the sink node.

Figure 6.2: Comparison of CS-based and Non-CS data gathering scheme.

original data vector $\mathbf{u}$ with high probability. Here, $m \ll n$ is defined as CS factor in this chapter, Φ is an $m \times n$ measurement matrix whose entries are i.i.d and can be generated from random distribution.

With aggregated observation $\mathbf{y}$, and basis matrix Ψ and sensing matrix Φ , the sink node can recover the sparse representation of $\mathbf{u}$, θ by solving the convex optimization problem as shown below in (6.5).

$$\min_{\theta} \|\theta\|_{l_1} \quad \text{subject to} \quad \mathbf{y} = \Phi \Psi \theta, \quad (6.5)$$

where $\|\theta\|_{l_1} = \sum_i |\theta_i|$. Let $\hat{\theta}$ be the optimal solution for the above convex optimization problem. The original data $\mathbf{u}$ can be recovered by $\hat{\mathbf{u}} = \Psi \hat{\theta}$ using CS recovery algorithms [36, 40, 41].

6.3 MINLP Formulation

This section formulates an optimization problem with the objective of maximizing information throughput subjected to the energy neutrality constraints and CS constraints. The notations used in this section is given in Table 6.1. The EH time is $\Gamma_e = \max_i \{t_i\}$. The data gathering time can be computed as $\Gamma_t = \Delta \sum_{s \in S} l_s$. The combinatorial optimization problem is then formulated as a MINLP problem as shown in the following,

$$\begin{aligned} \text{maximize} \quad & \rho = \frac{n}{\Gamma_e + \Gamma_t} \\ \text{subject to} \quad & \text{constraints} \quad (6.7) - (6.18) \end{aligned} \quad (6.6)$$

where all the constraints will be derived as follows.

The first constraint is energy neutrality constraint. For each node, the harvested energy should be more than the energy to be consumed, which can be

Table 6.1: Notations in MINLP formulation

Parameters	
S	A set of time frames that is large enough to deliver packets from all sensor nodes to the sink.
V	The set of nodes in the network.
E	The set of edges in the network.
r	The sink node.
n	Total number of nodes in the network.
m	CS factor.
σ_i	EH material size.
λ_i	The average EH rate for node i .
$c_{ij} > 0$	The link energy cost to transmit one data packet over link (i, j) .
Δ	Duration of one time frame.
Variables	
$a_{ij}^s \in \{0, 1\}$	Indicate whether link (i, j) is activated in time frame s .
$x_{ij} \in \{0, 1\}$	Indicate whether link (i, j) is selected in the data collection tree.
$f_{ij} \geq 0$	The data traffic load (number of packets transmitted) on link (i, j) .
$\bar{f}_{ij} \geq 0$	The virtual data traffic load on link (i, j) , used to specify connectivity.
$l_s \in \{0, 1\}$	Indicate that there is at least one transmission in time frame s .

described by the following constraint.

$$\lambda_i \sigma_i t_i \geq E_c^i \quad i \in V \setminus \{r\}. \quad (6.7)$$

Constraint (6.8) describes the data flow of the network. Introduction of CS techniques changes the traditional flow conservation constraints. A set of extended flow conservation constraints are given to characterize this phenomena. If i is the sink node, then the traffic flow out from node i is 0, while the data traffic flow into node i is the indegree of node i times m . Whereas, for nodes other than the sink

node, the outgoing traffic is m , the incoming traffic is m times indegree of a node.

$$\sum_{j:(i,j) \in E} f_{ij} - \sum_{j:(j,i) \in E} f_{ji} = \begin{cases} -m \sum_{j:(j,i) \in E} x_{ji} & i \in \{r\} \\ m(1 - \sum_{j:(j,i) \in E} x_{ji}) & \forall i \in V \setminus \{r\} \end{cases} \quad (6.8)$$

All the sensor nodes contribute to the information collected at the sink node. Therefore, the data collection tree is a spanning tree. Constraint (6.9) follows directly from the property that the number of edges for an n -node spanning tree is $n - 1$.

$$\sum_{(i,j) \in E} x_{ij} = n - 1. \quad (6.9)$$

Constraint (6.10) and constraint (6.11) use virtual link flow $\bar{f}_{ij}$ to ensure connectivity from every node to the sink node,

$$\sum_{j:(i,j) \in E} \bar{f}_{ij} - \sum_{j:(j,i) \in E} \bar{f}_{ji} \geq \frac{1}{n-1}, \quad \forall i \in V \setminus \{r\}. \quad (6.10)$$

$$x_{ij} - \bar{f}_{ij} \geq 0, \quad \forall (i, j) \in E. \quad (6.11)$$

The following constraints state that only links that carry positive traffic flow are selected to form the data collection tree.

$$f_{ij} - x_{ij} \geq 0, \quad \forall (i, j) \in E. \quad (6.12)$$

$$x_{ij} - \frac{f_{ij}}{n} \geq 0, \quad \forall (i, j) \in E. \quad (6.13)$$

Constraints (6.14) - (6.17) are imposed to achieve collision-free transmission as well as to facilitate compressive operation in each sensor node. TDMA is assumed as the access technique.

$$a_{ij}^s + a_{jk}^s \leq 1, \quad \forall s \in S, \quad \forall (i, j) \in E, \quad \forall (j, k) \in E. \quad (6.14)$$

$$\sum_{j:(i,j) \in E} a_{ij}^s \leq 1, \quad \forall i \in V, \quad \forall s \in S. \quad (6.15)$$

$$\sum_{s \in S} a_{ij}^s = x_{ij}, \quad \forall (i, j) \in E. \quad (6.16)$$

Constraint (6.14) is the half-duplex constraint ensuring that a node cannot transmit and receive at the same time. Constraint (6.15) is used to avoid collision for each time frame when a node j has multiple children node. Constraint (6.16) ensures that each link in the data collection tree is scheduled to transmit exactly once.

Furthermore, to facilitate CS, a node needs to collect all the data from all the downstream nodes. Therefore, the following constraint is given to guarantee transmission order. For example, in time frame s , if node i wants to transmit its data via link (i, k) , it needs to satisfy that all its children have already transmitted in previous time frames.

$$\sum_{\bar{s}=1}^{s-1} \sum_{j:(j,i) \in E} a_{ji}^{\bar{s}} + M(1 - a_{ik}^s) \geq \sum_{j:(j,i) \in E} x_{ji}, \quad (6.17)$$

$$\forall s \in S, \quad \forall (i, k) \in E,$$

in which M is a sufficiently large constant that is larger than the total number of links in the spanning tree T .

Finally, the total number of time frames used in data transmission is counted. The following constraints ensure that $l_s = 1$ only when at least one link is activated at time frame s .

$$l_s \geq a_{ij}^s, \quad \forall s \in S, \quad (i, j) \in E. \quad (6.18)$$

6.4 Hardness of the Problem

The following lemma establishes the NP-hardness of the above optimization problem.

Lemma 6.1. *The optimization problem to maximize information throughput subjected to energy neutrality constraint and compressive data gathering constraints is NP-hard.*

Proof. Observing the objective of the optimization problem, the information amount is fixed. The objective actually involves two aspects: to minimize EH time Γ_e and to minimize data gathering time Γ_t . Our problem can be then expressed as the following: Among all the spanning trees that can minimize EH time, find the tree and corresponding scheduling that leads to the least data delivery time. Hence, the problems consists of two parts: (1) find a set of spanning trees that minimize EH time; (2) among the trees, further find the tree and scheduling that minimize data delivery time.

First, the complexity of the first part of our problem is analyzed as the following. The EH time can be given as $\Gamma_e = \max\{t_i\}$, where $t_i = \frac{f_{ij}c_{ij}}{\lambda_i\sigma_i} = f_{ij}\varpi_{ij}$. With CS technique, the traffic load for each link is always m . Then, the problem of finding a spanning trees to minimize EH time can be converted to the problem of finding MBST. The problem of finding minimum bottleneck spanning tree asks for a spanning tree which minimizes edge cost of the bottleneck edge. The bottleneck edge of a spanning tree is the edge with the highest edge weight. In our problem, the edge weight is the CHR ϖ_{ij} . A graph may have many MBSTs. The problem of finding MBST can be efficiently solved by Camerini's algorithm [98].

Next, the hardness of the second part of our problem is proved by reducing from minimum edge coloring problem [122], which has been proven to be NP-hard [123]. The minimum edge coloring problem is to assign a color to each edge such that adjacent edges have different colors, and the total colors used is minimized. The second part of our problem can be described as follows: given a MBST, to find

the time frame allocation to facilitate compressed data transmission as well as to minimize data gathering time. In order to facilitate CS operation, nodes should transmit in order: a node needs to wait until all its children finish transmission so that it can compress for its own transmission. Hence, this second part of our problem can be decomposed into a series of minimum edge coloring problems. Different colors of the edges represent different time frames. In each edge coloring problem, an auxiliary graph is constructed, edges that are connected to the leaf nodes are added to the auxiliary graph. After solving minimum edge coloring problem in the auxiliary graph, the colored edges are removed from the MBST. This step is repeated until no edge is left in MBST. Then the number of iterations is the data gathering slots of the MBST. The second part of our optimization problem is the combination of a series of minimum edge coloring problems, which makes it NP-hard. To sum up, the original problem of maximizing information throughput subjected to energy neutrality constraint and compressive data gathering constraints is NP-hard.

6.5 Approximation Algorithms

In the MINLP formulation proposed in Section IV, the number of constraints grows exponentially as the number of nodes increases, which makes it prohibitive to solve this problem optimally in large scale network. To overcome the NP-hardness of the problem in large scale network, we break the problem into two sub-problems: 1) Construct a routing topology that achieves energy neutrality at each node and minimizes EH time; 2) Schedule data transmissions so that data can be successfully compressed en-route while minimizing total data delivery time.

6.5.1 CHR Based Compressed Data Gathering Tree

The heterogeneous average EH rates as well as energy consumption rate should be considered when constructing the routing topology. This section presents a CHR

based compressed data gathering tree construction algorithm. First, EH time is lower bounded by $f_{ij}\varpi_{ij}$. With CS technique, the traffic load f_{ij} is fixed at m . Hence, in order to minimize EH time, links with smaller CHR is preferred. In the proposed algorithm, the nodes are divided into different layers according to CHR. Specifically, each node finds a shortest path to the sink node based on CHR. Then, a hop count h_v is assigned to each node as the layer number. This method guarantees that the data packets are transmitted through the route with least CHR. Next, start from layer 1, each node connects to a parents selected from previous layers with the least descendants. This prevents a node from having too many children which increases data gathering latency. Layer by layer, This operation is repeated until all layers are examined and all the nodes find a parent. The algorithm can be summarized in Algorithm 7.

The time complexity of our proposed tree construction problem is analyzed as follows. It takes $O(|V|)$ to find a shortest path for every node. Assume the maximum hop count is H , and the number of nodes in level h is n_h . We have $n_1 + n_2 + \dots + n_H = |V| - 1$. In the worst case, it takes $O(n_h n_{h-1})$ to select parents for nodes in layer h . The time complexity for constructing the tree is $O(n_1 n_2 + n_2 n_3 + \dots + n_{H-1} n_H) = O((|V| - 1)^2)$. Hence, the overall time complexity of our proposed algorithm is $O(|V| + (|V| - 1)^2) = O(|V|^2)$.

- *Practical Implementation:* During the setup stage, each node acquires information about CHR distributedly. Then all the information will be collected at the sink node. The sink node runs Algorithm 7 to compute the sustainable data gathering tree T and disseminates the information to the whole network.

Algorithm 7: CHR based compressed data gathering tree construction.

Input : $G = (V, E)$, CHR ϖ , sink node r .
Output: Sustainable data gathering tree $T = (V, E')$.

```

1 foreach node  $v \in V$  do
2   | find the shortest path from node  $v$  to the sink node based on CHR  $\varpi$ ,
   | record hop count  $h_v$  to the sink node ;
3   | node  $v$  learns a set of neighbors  $N(v)$ ;
4 end
5 Connect the nodes in layer 1 to the sink node ;
6 for  $layer = 2$  to  $maxhop$  do
7   | Each node selects a parents from previous layers with least descendants
   | as parents ;
8 end

```

6.5.2 Graph-Based Priority-Aware Link Scheduling (GPLS)

Given a sustainable data gathering tree, this section presents a GPLS algorithm. Graph-based model is used as the interference model. Nodes are scheduled to transmit from the bottom of the tree up to the sink. At the beginning, each node will be assigned a priority value. Two major factors determine the priority. Namely, the parent's node degree and the distance in hop count to the sink node. In general, nodes that have more 'siblings' and are far away from the sink are given the priority to transmit. In each time frame, unscheduled nodes that are ready to transmit can be added to a *ReadytoTransmit* set. *ReadytoTransmit* set contains nodes that satisfy the following condition: 1) the node is a leaf node; 2) the node has received all the sensor readings from its children. From this *ReadytoTransmit* set, nodes are scheduled to transmit in current time frame according their priority without causing conflict with each other. This procedure is repeated until all the nodes are scheduled. The procedure of GPLS can be summarized as in Algorithm 8.

The time complexity of our link scheduling algorithms is analyzed as follows. It takes $O(|V|)$ to assign priority to each node. Then, it takes $O(|V|)$ to traverse every node and judge whether it can be added to *ReadytoTransmit* set. Assume

that on average, there are α nodes in the *ReadytoTransmit* set, then the sort step takes $O(\alpha)$. Assume that for each node, the size of the conflicting set is β , then the scheduling step takes $O(\alpha\beta)$. In the worst case, the outer loop may run $|V| - 1$ times. The overall time complexity for our algorithm is $O((2|V| + \alpha\beta)(|V| - 1))$. α, β are bounded by $|V|$, therefore, the time complexity of our algorithm is $O(|V|^3)$.

- *Practical Implementation:* Given the data gathering tree T from Algorithm 7, the sink node runs Algorithm 8 to obtain time frame assignment S and then disseminates S to all other nodes. Then, sensor nodes transmit their coded reading through the given topology in a determined time frame.

Algorithm 8: GPLS algorithm

Input : A sustainable data gathering tree T .
Output: A feasible schedule S to facilitate compressed data collection.

```

1 Initialization: assign priority to each node ;
2  $t = 1$  ;
3 repeat
4   forall the node  $v$  that is unscheduled do
5     if  $v$  is a leaf node or all the children of  $v$  have been scheduled then
6       add  $v$  into ReadytoTransmit set;
7     end
8   end
9   Sort the ReadytoTransmit set based on the priority of the nodes, get
   sortedReady set ;
10  forall the node  $u \in \text{sortedReady}$  do
11    if adding  $u$  to  $S_t$  won't introduce collision then
12      add node  $u$  into  $S_t$  ;
13      mark node  $u$  as scheduled;
14    end
15  end
16   $t = t + 1$  ;
17 until all the nodes are scheduled;
18  $T_{tx} = t$  ;
19  $S = S_1 \cup S_2 \cup \dots \cup S_{n-1}$  ;
```

6.6 Performance Analysis

This section presents the performance analysis of our algorithm by evaluating the approximation ratio of the proposed tree construction algorithm and giving the bound for our scheduling algorithm.

6.6.1 Approximation Ratio of the Tree Construction Algorithm

Lemma 6.2. *Given a graph G , if the maximum CHR is $\varpi_{\max}$, the minimum CHR is $\varpi_{\min}$. Define the ratio of $\varpi_{\max}$ and $\varpi_{\min}$ as $\chi = \frac{\varpi_{\max}}{\varpi_{\min}}$. Then our proposed tree construction algorithm is better than χ -approximation to the optimum.*

Proof. The objective is to construct a tree with minimum EH time. Write the optimum value as $OPT = \min\{\max_{(i,j) \in T^*} \{f_{ij}\varpi_{ij}\}\}$, where T^* is the optimal data gathering tree. For CS-based data gathering, the bottleneck traffic of the network is m . Therefore, $OPT = m \min\{\max_{(i,j) \in T^*} \{\varpi_{ij}\}\}$. The lower bound for OPT can be obtained as $OPT \geq m\varpi_{\min}$. Our proposed tree construction algorithm will give the performance of $m\varpi_{ij}$, where $(i,j) \in T$ and T is the proposed tree. The approximation ratio is $\frac{\varpi_{ij}}{\varpi_{\min}} \leq \frac{\varpi_{\max}}{\varpi_{\min}} = \chi$. Therefore, our proposed tree construction algorithm is better than χ -approximation to the optimum.

6.6.2 Bound of Scheduling Algorithm

Lemma 6.3. *Given a sustainable data gathering tree T , the lower bound on the time frames to schedule all transmissions is*

$$\max_{i \in V} \left\{ \sum_{j \in D_i} f_{ji} + f_{ik} h_i \right\}, \quad (6.19)$$

where k is the parent node of node i , D_i is the set of children node of node i , h_i is the hop count of node i to the sink node.

Proof. In the data gathering tree, every node needs to wait for its children to finish transmission so that data can be compressed and transmitted up towards the sink node. Denote the set of children of node i as D_i . For each node j in the set

D_i , the traffic from j to i as f_{ji} . Then, it takes $\sum_{j \in D_i} f_{ji}$ to complete gathering data from children. Similarly, after collecting all the data from its children nodes, node i needs to send the aggregated data through h_i hops to the sink node, which requires at least $f_{ik}h_i$ frames. If all transmissions can be done in parallel without collision with each other, then the possible lower bounds on the schedule time frames is (6.19).

6.7 Performance Evaluation

This section evaluates the performance of our proposed algorithm through simulations. First of all, the variations of maximum traffic load under different network size, node density and CS factor are investigated. The information throughput under different network size and node densities is examined. Next, the impact of CS factor to information throughput is explored. Finally, the influence of EH material size is also investigated. For comparison, Non-CS data gathering algorithms are also implemented. For instance, LLHC [124] and DCAT [125] tree construction algorithms are implemented with MWF [124] scheduling algorithm for EH-WSNs.

In this simulation, the sensor nodes are randomly deployed in a square field of 500×500 unit distance such that the network is connected. The average EH rate is assumed to be within the range of 10-15 mW/cm² (which is easily achievable by state-of-the-art solar cell [66]). Except for the last simulation, the EH material size is 20 cm². The packet length is assumed to be 2000 bits/packet with the transmission rate at 100 Mbps. The performance of our algorithm is considered under two scenarios: sparse network where the average node degree of each node is 5 and dense network where average node degree of each node is 10. The average node degree can be adjusted by changing the communication range of each node.

First of all, the performance of our proposed algorithm is evaluated in terms of bottleneck traffic, which manifests the benefit brought by the load balancing

(a) Bottleneck traffic as a function of the network size in sparse network . The number of nodes in the network grows from 100 to 450. CS factor $m = 10\%n$

(b) Bottleneck traffic as a function of the network size in dense network . The number of nodes in the network grows from 100 to 450. CS factor $m = 10\%n$

Figure 6.3: Bottleneck traffic for different network sizes and node densities

features of CS technique. The bottleneck traffic for both sparse network and dense network are plotted in Fig. 6.3a and 6.3b. It can be observed that our algorithm significantly reduces the bottleneck traffic. For example, in dense network, when CS factor $m = 0.2n$, when number of nodes n grows from 100 to 500, the bottleneck traffic of our proposed algorithm ranges from 20 to 100. The bottleneck traffic of Non-CS-LLHC ranges from 35 to 210, and the bottleneck traffic for Non-CS-DCAT increase from 39 to 250. Moreover, the advantage of our proposed algorithm is more obvious in sparse network. The reason is that in sparse network, the constructed data gathering tree tends to be taller compare to one in dense network. Therefore, in Non-CS-LLHC and Non-CS-DCAT network, nodes that are near to the sink node may incur more traffic load. While in the proposed algorithm, given certain number of nodes in the network, the traffic load for each link is the same. Consequently, our proposed algorithm can reduce bottleneck traffic, which leads to reduced EH time and improved sustainable information throughput.

Next, the information throughput of our proposed algorithm is compared with benchmark algorithms in both sparse and dense network. The information throughput for both sparse and dense network are plotted in Fig. 6.4a and 6.4b. It is notable that our proposed scheme achieves higher information throughput while maintaining the sustainable operation in both sparse and dense network. For example, when CS factor $m = 0.2n$, $n = 300$, our proposed algorithm achieves information throughput of 1500 in a dense network, which is about 3 times of the other compared algorithms. Furthermore, the benefits of our proposed algorithm is more significant in sparse network. It can be observed that our proposed algorithm can significantly improve the information throughput in large scale network.

Then, the impact of CS factor m to the information throughput is investigated. In this simulation, the number of nodes in the network is fixed to be 200, and the CS factor is varied from $5\%n$ to $40\%n$ to observe the information throughput. The results are plotted in Fig. 6.5a and 6.5b. As shown in the figure, when the number

(a) Information throughput as a function of the network size in sparse network. The number of nodes in the network grows from 100 to 450. CS factor $m = 10\%n$

(b) Information throughput as a function of the network size in dense network. The number of nodes in the network grows from 100 to 450. CS factor $m = 10\%n$

Figure 6.4: Information throughput for different network sizes and node densities.

(a) Information throughput as a function of the CS factor. The number of nodes in the network is 200. CS factor changes from $m = 10\%n$ to $m = 40\%n$

(b) Information throughput as a function of the CS factor in dense network. The number of nodes in the network is 200. CS factor changes from $m = 10\%n$ to $m = 40\%n$

Figure 6.5: Information throughput under different CS factor and node densities.

of nodes in the network is fixed, the information throughput decreases with the growth of CS factor. The increase in CS factor indicates more traffic load in each wireless link in our proposed algorithm. Therefore, the same amount of information is condensed into more data packets, which leads to the decrease of information throughput. Moreover, it can be concluded that the advantage of our proposed algorithm is dependent on the CS factor which is affected by the sparsity of the original sensory data. When the CS factor is large, the advantages of compressed data gathering for a sustainable system is less significant. For example, in dense network, the information throughput for our proposed algorithm is about 100 times of Non-CS-LLHC-MWF for $m = 5\%n$ but 1.2 times for $m = 40\%n$.

Finally, the impact of EH material size on the information throughput is studied. In this simulation, the number of nodes in the network is fixed to be 200, and the EH material size is varied from 10 cm² to 80 cm². The corresponding information throughput is observed. The results are plotted in Fig. 6.6. It can be observed from Fig. 6.6 that as the EH material size increases, information throughput increases for all the listed algorithms. However, to achieve the same information throughput, our proposed algorithm requires smaller EH material size, which is favorable for a more compact hardware design.

6.8 Concluding Remarks

This chapter presented a harvesting-aware compressed data gathering for optimal sustainable information throughput in EH-WSNs. An NP-hard optimization problem was formulated. To overcome the NP-hardness of the problem, the problem was decoupled into two sub-problems and solved by approximation algorithms. Finally, the performance of our algorithm was evaluated through simulation and compared with some state-of-the-art algorithms. The results showed that our proposed algorithm can significantly improve information throughput while maintain-

(a) Information throughput v.s. various EH material size in sparse network.

(b) Information throughput v.s. various EH material size in dense network.

Figure 6.6: Information throughput under different EH material size and node densities.

ing sustainable operation. The significance of performance gain is dependent on CS factor which is related to the sparsity of the original signal. Furthermore, to achieve the same sustainable information throughput, our proposed algorithm requires smaller EH material size, which is more favorable for compact hardware design.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis has provided solutions from different aspects to the challenges brought by the data deluge of pervasive sensor networks. When sensor nodes are powered by traditional non-rechargeable batteries, energy efficiency and network lifetime are the two primary considerations. Therefore, a function-based data aggregation algorithm has been proposed to reduce energy consumption as well as system latency. Moreover, a hybrid CS based algorithm for energy-efficient and low latency data gathering has been proposed. Further, a stochastic compressive data gathering has been proposed for sensor networks that work in dynamic environments. Finally, CS technique was applied into EH-WSNs to maximize the information throughput while achieving sustainable operation.

Chapter 3 has studied a function-based data aggregation algorithm to significantly reduce data transmissions and energy consumption. Specifically, a priority-aware hybrid scheduling algorithm was proposed to compute the MAX function in a single-hop WSN. In the proposed algorithm, a schedule assignment was generated for each node to determine its time and transmission sequence to transmit. Priority-aware TDMA scheme was used to first schedule the transmission. Then, in order to improve the performance with the influence of collision, CDMA is introduced to recover information whenever a collision occurs. The hybrid schedule

assignment is designed so that the node with the maximum value has the priority to transmit. The expression for computation failure probability was derived. The optimal value was found using an iterative method. Further, an optimization problem was formulated to minimize average computation time subjected to the constraints on failure probability. The synergy effect of priority-aware TDMA and CDMA scheme contributes to a novel technique for the reduction of average computation time and computation failure probability. Numerical results have shown that the proposed algorithm reduces average computation time and energy consumption for one-shot MAX function computation with good scalability.

Chapter 4 has studied a correlation-based data aggregation algorithm. An energy-efficient, low latency, and general data gathering algorithm based on hybrid CS with a dense measurement matrix was proposed. The problem was mathematically formulated as a multi-objective MINLP problem which was proved to be NP-hard. Two approximation algorithms were proposed to give practical solutions to the decomposed sub-problems. On one hand, an energy and latency aware hybrid CS based tree was constructed to minimize link energy cost as well as to reduce data gathering latency. Sensor nodes are categorized as aggregators and forwarders according to the tree topology and a predefined CS factor, which introduces hybrid traffic into the network. Aggregators transmit coded sensor readings while forwarders just transmit the original sensor readings. On the other hand, SINR-based priority-aware scheduling was proposed to reduce data gathering latency while guaranteeing successful concurrent transmissions. Simulation results have shown that the proposed algorithm achieves pronounced energy conservation for one round data gathering compared with benchmark algorithms. Moreover, the proposed algorithm achieves lower data gathering latency compared with the benchmark algorithms. Finally, the proposed algorithm significantly extends network lifetime.

Chapter 5 has studied a correlation-based data aggregation algorithm for sen-

sors working in dynamic environments. A cost-aware stochastic compressive data gathering scheme was proposed for multi-hop WSNs. The cost diversity was considered in the design framework. A Markov chain based method was proposed to analyze the relationship between the expected energy cost and recovery error. An optimization problem to minimize the total expected energy cost for data gathering subjected to the recovery error and global routing randomness constraints was formulated. The goal is to construct an optimized transition probability matrix so that the equivalent measurement matrix constructed from the absorbing Markov chain can guarantee the recovery of data with a certain error bound. Simulation results have shown that the proposed algorithm prolongs network lifetime and achieves a good trade-off between total expected energy cost and the recovery error.

Chapter 6 has explored the possibility of applying CS into EH-WSNs. For EH-WSNs with heterogeneous EH rate, a harvesting-aware compressed data gathering scheme was developed. CS technique was exploited to condense the data for transmission, which improved the information throughput. An optimization problem to maximize the information throughput subjected to energy neutrality and compressive data gathering constraints was formulated with its complexity proven. To overcome the NP-hardness of the problem, the problem was decoupled into two sub-problems and solved by two approximation algorithms. Finally, the performance of our proposed scheme was evaluated through simulation, comparing with state-of-the-art algorithms. The results showed that our proposed scheme can significantly improve sustainable information throughput. The significance of performance gain is dependent on CS factor which is related to the sparsity of the original data vector. Furthermore, to achieve the same information throughput, our proposed scheme requires smaller EH material size, which is more favorable for compact hardware design.

7.2 Future Work

In Chapter 3, a fast and reliable MAX function computation has been proposed in a single-hop network through a proper assignment of transmission time and code sequence. Numerical results are provided to evaluate the performance. Apart from the numerical results, further investigation on how to capture the average performance using a network simulator can be done in the future. In this work, fixed transmission power and AWGN channel were assumed. Consequently, the following extensions are possible for our future work.

- First, the work can be extended into multi-hop networks. For example, to extend the work into a two-hop network and investigate the delay performance and energy consumption performance.
- Second, another dimension of control can be added to further conserve the power consumption. For example, the transmit power of each node can be adjusted and optimized to minimize the total power consumption.
- Third, more complicated channel model such as Rayleigh fading channel can be considered.
- Fourth, the in-network computation of more functions, such as the median, can be explored.

In Chapter 4, a hybrid CS based data gathering scheme was proposed for multi-hop network to minimize data gathering delay and energy consumption while guaranteeing the data recovery quality. In the proposed work, the NP-hard problem was solved by two approximation algorithms. One interesting direction is that we can explore possible algorithm to solve for the exact solution of the NP-hard problem and compare with the result of approximation algorithms. Moreover, in the proposed work, uniform transmit power was assumed. Another possible extension

is to investigate the performance under heterogeneous transmit power and adopt some power control algorithms to further optimize the total power consumption. Furthermore, in the proposed work, we evaluated the recovery quality using synthetic data. It would be more interesting to apply CS technique on real world data and access the recovery quality.

In Chapter 5, a cost-aware stochastic compressive data gathering scheme was proposed with reliable links. However, in practical environment, it's common that the wireless link is unreliable. In unreliable wireless networks, packet delivery ratio serves as a key metric to measure the resiliency of routing in WSNs. Therefore, one future direction is to consider various reliability of the wireless links and investigate the transmission policy to maximize the expected packet delivery ratio. The robustness of the algorithm is also an important consideration. In the future, we would like to conduct experiments to evaluate the performance of the proposed scheme under the condition of node failure.

In Chapter 6, a harvesting-aware CS-based transmission scheme was proposed to maximize sustainable information throughput for EH-WSNs. In the proposed work, the NP-hard problem was solved by two approximation algorithms. One possible extension is to investigate some efficient methods to solve for the exact solution and compare with the result of the approximation algorithms. Another possible extension would be introducing an energy forecasting model to predict energy arrival and estimate the expected energy intake. The energy prediction model will help the EH system to wisely manage energy usage and storage. There are several types of models in the literature such as Markov chain based models [126], leaky bucket model [127], and ETH predictor [128] for different types of EH devices and applications.

List of Publications

Journal Papers

- [1] Jiajia Huang and Boon-Hee Soong. “Priority-aware hybrid scheduling for fast, energy-efficient max function computation in single-hop networks.” *IET Communications*, 2016, 10(18): 2606-2612.
- [2] Jiajia Huang and Boon-Hee Soong “Low latency data gathering for sensor networks based on energy-aware hybrid compressive sensing.” submitted to *IEEE Internet of Things Journal*, 2017.
- [3] Jiajia Huang and Boon-Hee Soong “Cost-aware stochastic compressive data gathering for wireless sensor networks.” submitted to *IEEE Transactions on Vehicular Technology*, 2017.
- [4] Jiajia Huang and Boon-Hee Soong “Harvesting-aware compressed data gathering for optimal sustainable information throughput in sensor networks” submitted to *Computer Communications*, 2017.

Conference Papers

- [1] Jiajia Huang, Boon-Hee Soong and Shiyong Han “Energy-conserving timer-based max function computation for wireless sensor network,” in *IEEE Asia Pacific Wireless Communications Symposium (APWCS)*, 2015.
- [2] Jiajia Huang and Boon-Hee Soong “Fast timer-based maximum function com-

List of Publications

- putation over noisy multiple access channels” in *IEEE Consumer Communications & Networking Conference (CCNC)*, 2016.
- [3] Jiajia Huang and Boon-Hee Soong “Hybrid compressive sensing for delay-efficient sustainable data gathering.” in *2017 IEEE 85th Vehicular Technology Conference (VTC)*, 2017.

Bibliography

- [1] “Is this the future of the internet of things?” 2015, accessed: 2017-06-28. [Online]. Available: <https://www.weforum.org/agenda/2015/11/is-this-future-of-the-internet-of-things/>
- [2] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context aware computing for the internet of things: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.
- [3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
- [4] D. Evans, “The internet of things: How the next evolution of the internet is changing everything,” *CISCO white paper*, vol. 1, no. 2011, pp. 1–11, 2011.
- [5] R. G. Baraniuk, “More is less: Signal processing and the data deluge,” *Science*, vol. 331, no. 6018, pp. 717–719, 2011.
- [6] Y. Chen and Q. Zhao, “On the lifetime of wireless sensor networks,” *IEEE Communications Letters*, vol. 9, no. 11, pp. 976–978, 2005.
- [7] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, “A line in the sand:

- A wireless sensor network for target detection, classification, and tracking,” *Computer Networks*, vol. 46, no. 5, pp. 605–634, 2004.
- [8] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [9] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [10] O. Omeni, A. C. W. Wong, A. J. Burdett, and C. Toumazou, “Energy efficient medium access protocol for wireless medical body area sensor networks,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 2, no. 4, pp. 251–259, Dec 2008.
- [11] R. Rajagopalan and P. K. Varshney, “Data-aggregation techniques in sensor networks: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 8, no. 4, pp. 48–63, Fourth 2006.
- [12] J. Kim, X. Lin, N. B. Shroff, and P. Sinha, “Minimizing delay and maximizing lifetime for wireless sensor networks with anycast,” *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 2, pp. 515–528, 2010.
- [13] S. Guo, L. He, Y. Gu, B. Jiang, and T. He, “Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links,” *IEEE Transactions on Computers*, vol. 63, no. 11, pp. 2787–2802, 2014.
- [14] M. Younis and K. Akkaya, “Strategies and techniques for node placement in wireless sensor networks: A survey,” *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.
- [15] N. A. Pantazis and D. D. Vergados, “A survey on power control issues in wireless sensor networks.” *IEEE Communications Surveys & Tutorials*, vol. 9, no. 1-4, pp. 86–107, 2007.

- [16] F. K. Shaikh and S. Zeadally, “Energy harvesting in wireless sensor networks: A comprehensive review,” *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 1041–1054, 2016.
- [17] M. T. Penella, J. Albesa, and M. Gasulla, “Powering wireless sensor nodes: Primary batteries versus energy harvesting,” in *Instrumentation and Measurement Technology Conference, 2009. I2MTC’09. IEEE*. IEEE, 2009, pp. 1625–1630.
- [18] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [19] C. M. Sadler and M. Martonosi, “Data compression algorithms for energy-constrained devices in delay tolerant networks,” in *Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, 2006, pp. 265–278.
- [20] H. Yousefi, M. Malekimaajid, M. Ashouri, and A. Movaghar, “Fast aggregation scheduling in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 3402–3414, 2015.
- [21] R. Van Renesse, “The importance of aggregation,” in *Future Directions in Distributed Computing*. Springer, 2003, pp. 87–92.
- [22] P. Jesus, C. Baquero, and P. S. Almeida, “A survey of distributed data aggregation algorithms,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 381–404, 2015.
- [23] A. Giridhar and P. Kumar, “Computing and communicating functions over sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 755–764, Apr. 2005.

- [24] N. Freris, H. Kowshik, and P. Kumar, “Fundamentals of large sensor networks: connectivity, capacity, clocks, and computation,” *Proc. IEEE*, vol. 98, no. 11, pp. 1828–1846, Nov. 2010.
- [25] L. Ying, R. Srikant, and G. Dullerud, “Distributed symmetric function computation in noisy wireless sensor networks,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4826–4833, Dec. 2007.
- [26] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “Tag: A tiny aggregation service for ad-hoc sensor networks,” *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 131–146, 2002.
- [27] C. Li and H. Dai, “Efficient in-network computing with noisy wireless channels,” *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2167–2177, Nov. 2013.
- [28] X. Cao, X. Wang, and S. Lu, “Function computation over heterogeneous wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1756–1766, Jul. 2014.
- [29] S. Kamath, D. Manjunath, and R. Mazumdar, “On distributed function computation in structure-free random wireless networks,” *IEEE Transactions on Information Theory*, vol. 60, no. 1, pp. 432–442, 2014.
- [30] J. Sacha, J. Napper, C. Stratan, and G. Pierre, “Adam2: Reliable distribution estimation in decentralised environments,” in *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*. IEEE, pp. 697–707.
- [31] M. Jelasity, A. Montresor, and O. Babaoglu, “Gossip-based aggregation in large dynamic networks,” *ACM Transactions on Computer Systems (TOCS)*, vol. 23, no. 3, pp. 219–252, 2005.

Bibliography

- [32] D. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Transactions on information Theory*, vol. 19, no. 4, pp. 471–480, 1973.
- [33] J. Chou, D. Petrovic, and K. Ramachandran, “A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks,” in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2. IEEE, 2003, pp. 1054–1062.
- [34] S. S. Pradhan, J. Kusuma, and K. Ramchandran, “Distributed compression in a dense microsensor network,” *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 51–60, 2002.
- [35] Z. Xiong, A. D. Liveris, and S. Cheng, “Distributed source coding for sensor networks,” *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80–94, 2004.
- [36] E. J. Cands, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [37] E. J. Candes and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [38] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [39] M. Alodeh, S. Chatzinotas, and B. Ottersten, “Spatial dct-based channel estimation in multi-antenna multi-cell interference channels,” *IEEE Transactions on Signal Processing*, vol. 63, no. 6, pp. 1404–1418, 2015.
- [40] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

- [41] E. J. Candes and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?” *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.
- [42] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [43] J. A. Tropp, “Greed is good: Algorithmic results for sparse approximation,” *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [44] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications*. Cambridge University Press, 2012.
- [45] S. K. Sharma, E. Lagunas, S. Chatzinotas, and B. Ottersten, “Application of compressive sensing in cognitive radio communications: A survey,” *IEEE Communications Surveys & Tutorials*, vol. PP, no. 99, pp. 1–1, 2016.
- [46] W. Chen and I. J. Wassell, “Optimized node selection for compressive sleeping wireless sensor networks,” *IEEE Transactions on Vehicular Technology*, 2016.
- [47] D. L. Donoho, M. Elad, and V. N. Temlyakov, “Stable recovery of sparse overcomplete representations in the presence of noise,” *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 6–18, 2006.
- [48] C. Luo, F. Wu, J. Sun, and C. W. Chen, “Compressive data gathering for large-scale wireless sensor networks,” in *Proceedings of the 15th annual international conference on mobile computing and networking*. ACM, pp. 145–156.

- [49] X. Liu, L. Jun, and C. Rosenberg, “Compressed data aggregation: Energy-efficient and high-fidelity data collection,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1722–1735, 2013.
- [50] G. Chen, X. Y. Liu, L. Kong, J. L. Lu, and M. Y. Wu, “Multi-attribute compressive data gathering,” in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, 2014, pp. 2178–2183.
- [51] D. Ebrahimi and C. Assi, “On the interaction between scheduling and compressive data gathering in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2015.
- [52] W. Chen, Y. Andreopoulos, I. Wassell, and M. R. Rodrigues, “Towards energy neutrality in energy harvesting wireless sensor networks: A case for distributed compressive sensing?” in *Global Communications Conference (GLOBECOM), 2013 IEEE*. IEEE, 2013, pp. 474–479.
- [53] X. Y. Liu, Y. Zhu, L. Kong, C. Liu, Y. Gu, A. V. Vasilakos, and M. Y. Wu, “Cdc: Compressive data collection for wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2188–2197, 2015.
- [54] X. Liu, L. Jun, and A. Vasilakos, “Compressed data aggregation for energy efficient wireless sensor networks,” in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*, 2011, pp. 46–54.
- [55] W. Wang, M. Garofalakis, and K. Ramchandran, “Distributed sparse random projections for refinable approximation,” in *Proceedings of the 6th international conference on information processing in sensor networks*. ACM, 2007, pp. 331–339.

- [56] Y. Shen, W. Hu, R. Rana, and C. T. Chou, “Nonuniform compressive sensing for heterogeneous wireless sensor networks,” *IEEE Sensors Journal*, vol. 13, no. 6, pp. 2120–2128, 2013.
- [57] R. Rana, W. Hu, and C. T. Chou, “Optimal sampling strategy enabling energy-neutral operations at rechargeable wireless sensor networks,” *IEEE Sensors Journal*, vol. 15, no. 1, pp. 201–208, 2015.
- [58] A. Amini, V. Montazerhodjat, and F. Marvasti, “Matrices with small coherence using p -ary block codes,” *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 172–181, Jan 2012.
- [59] S. Li and G. Ge, “Deterministic sensing matrices arising from near orthogonal systems,” *IEEE Transactions on Information Theory*, vol. 60, no. 4, pp. 2291–2302, April 2014.
- [60] G. Yang, V. Y. F. Tan, C. K. Ho, S. H. Ting, and Y. L. Guan, “Wireless compressive sensing for energy harvesting sensor nodes,” *IEEE Transactions on Signal Processing*, vol. 61, no. 18, pp. 4491–4505, 2013.
- [61] W. Chen and I. Wassell, “Cost-aware activity scheduling for compressive sleeping wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. PP, no. 99, pp. 1–1, 2016.
- [62] X. Xu, R. Ansari, A. Khokhar, and A. V. Vasilakos, “Hierarchical data aggregation using compressive sensing (hdacs) in wsns,” *ACM Transactions on Sensor Networks*, vol. 11, no. 3, pp. 1–25, 2015.
- [63] L. Quan, S. Xiao, X. Xue, and C. Lu, “Neighbor-aided spatial-temporal compressive data gathering in wireless sensor networks,” *IEEE Communications Letters*, vol. 20, no. 3, pp. 578–581, 2016.
- [64] A. Ravelomanantsoa, H. Rabah, and A. Rouane, “Compressed sensing: A simple deterministic measurement matrix and a fast recovery algorithm,”

- IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 12, pp. 3405–3413, Dec 2015.
- [65] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, “Energy harvesting wireless communications: A review of recent advances,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 3, pp. 360–381, 2015.
- [66] S. Sudevalayam and P. Kulkarni, “Energy harvesting sensor nodes: Survey and implications,” *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 443–461, 2011.
- [67] J. A. Paradiso and T. Starner, “Energy scavenging for mobile and wireless electronics,” *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 18–27, 2005.
- [68] A. Kurs, A. Karalis, R. Moffatt, J. D. Joannopoulos, P. Fisher, and M. Soljačić, “Wireless power transfer via strongly coupled magnetic resonances,” *Science*, vol. 317, no. 5834, pp. 83–86, 2007.
- [69] N. S. Shenck and J. A. Paradiso, “Energy scavenging with shoe-mounted piezoelectrics,” *IEEE Micro*, vol. 21, no. 3, pp. 30–42, 2001.
- [70] W. K. Seah, Z. A. Eu, and H.-P. Tan, “Wireless sensor networks powered by ambient energy harvesting (wsn-heap)-survey and challenges,” in *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on*. IEEE, pp. 1–5.
- [71] O. Orhan and E. Erkip, “Throughput maximization for energy harvesting two-hop networks,” in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, pp. 1596–1600.

- [72] G. K. Ottman, H. F. Hofmann, A. C. Bhatt, and G. A. Lesieutre, “Adaptive piezoelectric energy harvesting circuit for wireless remote power supply,” *IEEE Transactions on Power Electronics*, vol. 17, no. 5, pp. 669–676, 2002.
- [73] F. I. Simjee and P. H. Chou, “Efficient charging of supercapacitors for extended lifetime of wireless sensor nodes,” *IEEE Transactions on Power Electronics*, vol. 23, no. 3, pp. 1526–1536, 2008.
- [74] Z. A. Eu, W. K. Seah, and H.-P. Tan, “A study of mac schemes for wireless sensor networks powered by ambient energy harvesting,” in *Proceedings of the 4th Annual International Conference on Wireless Internet*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), p. 78.
- [75] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on*. IEEE, 2000, p. 10 pp. vol. 2.
- [76] H. Kowshik and P. Kumar, “Optimal computation of symmetric boolean functions in collocated networks,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 639–654, 2013.
- [77] C.-Y. Wang, S.-W. Jeon, and M. Gastpar, “Multi-round computation of type-threshold functions in collocated gaussian networks,” in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 2154–2158.
- [78] N. Ma, P. Ishwar, and P. Gupta, “Information-theoretic bounds for multi-round function computation in collocated networks,” in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*. IEEE, 2009, pp. 2306–2310.

- [79] M. Goldenbaum and S. Stanczak, “Robust analog function computation via wireless multiple-access channels,” *IEEE Transactions on Communications*, vol. 61, no. 9, pp. 3863–3877, 2013.
- [80] A. Giridhar and P. Kumar, “Toward a theory of in-network computation in wireless sensor networks,” *IEEE Communications Magazine*, vol. 44, no. 4, pp. 98–107, 2006.
- [81] S. K. Udgata, G. Sindhura, and S. L. Sabat, “A novel architecture for intelligent and adaptive wireless sensor network based alarming system,” in *Wireless Communication and Sensor Networks (WCSN), 2009 Fifth IEEE Conference on*. IEEE, 2009, pp. 1–6.
- [82] N. Khude, A. Kumar, and A. Karnik, “Time and energy complexity of distributed computation of a class of functions in wireless sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 617–632, 2008.
- [83] M. Ye, C. Li, G. Chen, and J. Wu, “Eecs: an energy efficient clustering scheme in wireless sensor networks,” in *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*. IEEE, 2005, pp. 535–540.
- [84] O. Younis and S. Fahmy, “Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.
- [85] A. Anand and N. B. Mehta, “Quick, decentralized, energy-efficient one-shot max function computation using timer-based selection,” *IEEE Transactions on Communications*, vol. 63, no. 3, pp. 927–937, 2015.
- [86] J. Tang, X. Jiao, and W. Xiao, “Minimum-latency data aggregation in duty-cycled wireless sensor networks under physical interference model,” in *Wire-*

- less and Optical Communication Conference (WOCC), 2013 22nd.* IEEE, 2013, pp. 309–314.
- [87] X.-Y. Li, X. Xu, S. Wang, S. Tang, G. Dai, J. Zhao, and Y. Qi, “Efficient data aggregation in multi-hop wireless sensor networks under physical interference model,” in *Mobile Adhoc and Sensor Systems, 2009. MASS’09. IEEE 6th International Conference on.* IEEE, 2009, pp. 353–362.
- [88] R. J. Vanderbei and D. F. Shanno, “An interior-point algorithm for nonconvex nonlinear programming,” *Computational Optimization and Applications*, vol. 13, no. 1, pp. 231–252, 1999.
- [89] D. Son, B. Krishnamachari, and J. Heidemann, “Experimental study of concurrent transmission in wireless sensor networks,” in *Proceedings of the 4th international conference on Embedded networked sensor systems.* ACM, 2006, pp. 237–250.
- [90] I. Dietrich and F. Dressler, “On the lifetime of wireless sensor networks,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, p. 5, 2009.
- [91] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, “Compressed sensing for networked data,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 92–101, 2008.
- [92] R. Xie and X. Jia, “Transmission-efficient clustering method for wireless sensor networks using compressive sensing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 806–815, 2014.
- [93] W. Wang, M. J. Wainwright, and K. Ramchandran, “Information-theoretic limits on sparse signal recovery: Dense versus sparse measurement matrices,” *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2967–2979, 2010.

- [94] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, “A compressive sensing approach to urban traffic estimation with probe vehicles,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2289–2302, 2013.
- [95] X. Wu, Q. Wang, and M. Liu, “In-situ soil moisture sensing: Measurement scheduling and estimation using sparse sampling,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, no. 2, p. 26, 2015.
- [96] D. Gong and Y. Yang, “Low-latency sinr-based data gathering in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 6, pp. 3207–3221, 2014.
- [97] M. Andrews and M. Dinitz, “Maximizing capacity in arbitrary wireless networks in the sinr model: Complexity and game theory,” in *INFOCOM 2009, IEEE*, pp. 1332–1340.
- [98] P. M. Camerini, “The min-max spanning tree problem and some extensions,” *Information Processing Letters*, vol. 7, no. 1, pp. 10–14, 1978.
- [99] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer, “Complexity in geometric sinr,” in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2007, pp. 100–109.
- [100] L. B. Le, E. Modiano, C. Joo, and N. B. Shroff, “Longest-queue-first scheduling under sinr interference model,” in *Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing*. ACM, pp. 41–50.
- [101] U. S. Wijetunge, R. Subramanian, A. Pollok, and S. Perreau, “Maximizing packet delivery ratio in stochastic routing: An analytical perspective,” in *2014 IEEE International Conference on Communications (ICC)*, Conference Proceedings, pp. 415–420.

- [102] M. Zhao, P. H. J. Chong, and H. C. Chan, “An energy-efficient and cluster-parent based rpl with power-level refinement for low-power and lossy networks,” *Computer Communications*, 2016.
- [103] R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer, “Network correlated data gathering with explicit communication: Np-completeness and algorithms,” *IEEE/ACM Transactions On Networking*, vol. 14, no. 1, pp. 41–54, 2006.
- [104] J. Zheng, P. Wang, and C. Li, “Distributed data aggregation using slepi-anwolf coding in cluster-based wireless sensor networks,” *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2564–2574, 2010.
- [105] Z. Haifeng, Y. Feng, T. Xiaohua, G. Xiaoying, W. Xinbing, and X. Shilin, “Data gathering with compressive sensing in wireless sensor networks: A random walk based approach,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 35–44, 2015.
- [106] L. Xu, X. Hao, N. Lane, X. Liu, T. Moscibroda, and S. Nath, *Cost-aware compressive sensing for networked sensing systems Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, 2015.
- [107] N. Christofides, *Graph theory: An algorithmic approach*. Academic Press, Inc., New York, 1975.
- [108] A. Cohen, W. Dahmen, and R. DeVore, “Compressed sensing and best,” *Journal of the American Mathematical Society*, vol. 22, no. 1, pp. 211–231, 2009.
- [109] M. Vetterli and J. Kovacevic, *Wavelets and subband coding*. Prentice-Hall, 1995.
- [110] A. Gilbert and P. Indyk, “Sparse recovery using sparse matrices,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 937–947, 2010.

Bibliography

- [111] C. M. Grinstead and J. L. Snell, *Introduction to probability*. American Mathematical Soc., 2012.
- [112] M. Saerens, Y. Achbany, F. Fouss, and L. Yen, “Randomized shortest-path problems: Two related models,” *Neural Computation*, vol. 21, no. 8, pp. 2363–2404, 2009.
- [113] “Intellab data,” 2004, accessed: 2017-04-18. [Online]. Available: <http://db.csail.mit.edu/labdata/labdata.html>
- [114] I. Mabrouki, X. Lagrange, and G. Froc, “Random walk based routing protocol for wireless sensor networks,” in *Proceedings of the 2Nd International Conference on Performance Evaluation Methodologies and Tools*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007, pp. 71:1–71:10.
- [115] S. Basagni, M. Y. Naderi, C. Petrioli, and D. Spensa, “Wireless sensor networks with energy harvesting,” *Mobile Ad Hoc Networking: The Cutting Edge Directions*, pp. 701–736, 2013.
- [116] E. Lattanzi, E. Regini, A. Acquaviva, and A. Bogliolo, “Energetic sustainability of routing algorithms for energy-harvesting wireless sensor networks,” *Computer Communications*, vol. 30, no. 14, pp. 2976–2986, 2007.
- [117] M. K. Jakobsen, J. Madsen, and M. R. Hansen, “Dehar: A distributed energy harvesting aware routing algorithm for ad-hoc multi-hop wireless sensor networks,” in *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a*. IEEE, 2010, pp. 1–9.
- [118] Z. A. Eu, H.-P. Tan, and W. K. Seah, “Design and performance analysis of mac schemes for wireless sensor networks powered by ambient energy harvesting,” *Ad Hoc Networks*, vol. 9, no. 3, pp. 300–323, 2011.

Bibliography

- [119] N. Sadagopan and B. Krishnamachari, “Maximizing data extraction in energy-limited sensor networks,” in *IEEE INFOCOM 2004*, vol. 3, pp. 1717–1727 vol.3.
- [120] S. Luo, R. Zhang, and T. J. Lim, “Optimal save-then-transmit protocol for energy harvesting wireless transmitters,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 3, pp. 1196–1207, 2013.
- [121] O. D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, “Fast data collection in tree-based wireless sensor networks,” *Mobile Computing, IEEE Transactions on*, vol. 11, no. 1, pp. 86–99, 2012.
- [122] D. Jungnickel and T. Schade, *Graphs, networks and algorithms*. Springer, 2005.
- [123] I. Holyer, “The np-completeness of edge-coloring,” *SIAM Journal on Computing*, vol. 10, no. 4, pp. 718–720, 1981.
- [124] D. Gong and Y. Yang, “Low-latency sinr-based data gathering in wireless sensor networks,” in *INFOCOM, 2013 Proceedings IEEE*, pp. 1941–1949.
- [125] J. Gagnon and L. Narayanan, “Efficient scheduling for minimum latency aggregation in wireless sensor networks,” in *Wireless Communications and Networking Conference (WCNC), 2015 IEEE*, pp. 1024–1029.
- [126] C. K. Ho, P. D. Khoa, and P. C. Ming, “Markovian models for harvested energy in wireless communications,” in *Communication Systems (ICCS), 2010 IEEE International Conference on*. IEEE, 2010, pp. 311–315.
- [127] A. Kansal, D. Potter, and M. B. Srivastava, “Performance aware tasking for environmentally powered sensor networks,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 32, no. 1, pp. 223–234, 2004.

Bibliography

- [128] C. Moser, L. Thiele, D. Brunelli, and L. Benini, “Adaptive power management in energy harvesting systems,” in *Proceedings of the conference on Design, automation and test in Europe*. EDA Consortium, 2007, pp. 773–778.