# Stony Brook University


OFFICIAL COPY

**The official electronic file of this thesis or dissertation is maintained by the University Libraries on behalf of The Graduate School at Stony Brook University.**

**© All Rights Reserved by Author.**

# Mobile Data Gathering and Energy Replenishment in Wireless Sensor Networks: Theoretical and Experimental Approaches

A Dissertation Presented

by

**Ji Li**

to

The Graduate School

in Partial Fulfillment of the Requirements

for the Degree of

**Doctor of Philosophy**

in

**Electrical Engineering**

Stony Brook University

May 2017

**Stony Brook University**

The Graduate School

**Ji Li**

We, the dissertation committee for the above candidate for the Doctor of Philosophy degree, hereby recommend acceptance of this dissertation.

Yuanyuan Yang – Dissertation Advisor
Professor, Department of Electrical and Computer Engineering

Sangjin Hong – Chairperson of Defense
Professor, Department of Electrical and Computer Engineering

Dmitri Donetski
Associate Professor, Department of Electrical and Computer Engineering

Samir R. Das
Professor, Department of Computer Science

This dissertation is accepted by the Graduate School.

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

# Mobile Data Gathering and Energy Replenishment in Wireless Sensor Networks: Theoretical and Experimental Approaches

by

**Ji Li**

**Doctor of Philosophy**

in

**Electrical Engineering**

Stony Brook University

2017

Energy constraint is one of the major constraints in the design and deployment of conventional wireless sensor networks. In such networks, usually powered by batteries with limited capacity, one or few number of static data sinks are deployed to collect sensory data from the network through multi-hop relay in non-uniform pattern. Such a methodology inevitably causes quick depletion of battery energy in the sensors which transmit large amount of data, e.g., the sensors around data sinks. This results in the formation of energy holes that may disconnect sensors from the networks, and even terminates the operation of the networks.

Mobile data gathering provides a reasonable approach to alleviate this problem as one or more mobile data collectors roam over the sensing field and work as data sinks to collect data from surrounding sensors.

By virtually increasing the number of data sinks and careful calculate their locations in the networks, the length of relays are shortened and the amount of data being transmitted are greatly decreased, thus energy can be significantly saved in sensors.

In the meanwhile, energy harvesting techniques have been applied in wireless sensor networks to supply the sensors with the energy obtained from ambient environment. Recent breakthrough in wireless power transfer based on inductance resonance emerges as a promising method to relieve energy limitation in wireless sensor networks. As mobile energy transporters are employed for energy injection into the networks, the sensors are provided with sustained energy through wireless recharge charging, and perpetual network operation can be achieved.

This dissertation focuses on scheme design and performance optimization of mobile data gathering and energy replenishment, and experimental evaluation in WSNs. We present a joint design of mobile data gathering and energy replenishment to maximize the network utility in wireless sensor networks. We develop an efficient NDN-based protocol to collect real-time battery information from the network so that the trajectories of mobile data collectors or energy transporters as well as routing scheme for the sensory data can be carefully calculated based on the energy distribution. We studies energy neutral problem and designed algorithms for recharge scheduling of the sensors based on their energy situation to achieve perpetual network operation. We propose a mobility assisted data gathering scheme with solar irradiance awareness which utilizes data sinks with limited mobility in solar-powered sensor networks to maximize the amount of sensory data that can be collected from the network. We also design a versatile platform for experimental research on mobile data gathering in wireless sensor networks. Our testbed includes both wireless sensor node and mobile data collector on which algorithms and protocols can run in real world. Compared with mathematical analysis and software simulation, which may neglect the impact of many real factors due to the limitation of the

models and present inaccurate results of performance evaluation. Our testbed provides a more convincing method for performance evaluation by, instead of modeling the real world, running the algorithms and protocols in the same environment the application will be deployed. It also offers the assistance of deep insight for system modeling, which could help the development of better solutions. We implement mobile data gathering on our testbed and conduct experiment for wildlife surveillance. It successfully captures the impact of many real factors which are usually omitted in theoretical analysis and software simulation due to the limitation of the models.

*To my beloved wife, my son, and my parents for their love and support.*

# Contents

# List of Figures

# List of Tables

# Acknowledgements

I would like to show my gratitude to numerous people, who helped and supported me during my Ph.D. study.

First and foremost, I am heartily thankful to my supervisor, Prof. Yuanyuan Yang, for her enthusiastic supervision and strong support to my research and professional growth throughout the course of dissertation. Her broad vision, deep insight, extensive experience, valuable advice and strong support are essential to make my research possible. I can never forget the countless times she worked so hard and carefully on our papers. I hope I could learn more from her professionalism, persistence and strong-mindness, which will greatly benefit my future career development.

I would also like to thank my defense committees, Prof. Sangjin Hong and Prof. Dmitri Donetski in Department of Electrical and Computer Engineering, and Prof. Samir R. Das in Department of Computer Science. Thanks for their precious time and valuable suggestions to improve my dissertation quality.

I am grateful to the previous and current members of High Performance Computing and Networking Research Laboratory, Dr. Min Yang, Dr. Lin Liu, Dr. Miao Zhao, Dr. Xi Deng, Dr. Dawei Gong, Dr. Zhiyang Guo, Dr. Zhemin Zhang, Zhexi Pan, Cong Wang, Jun Duan, Zhenhua Li, Pengzhan Zhou, Yaodong Huang, for their friendship and all of their help.

I would also like to express my gratitude to many individuals in the department who have made my stay at Stony Brook pleasant and memorable. Prof. Fan Ye, Prof. Kenneth Short, Staff Anthony Olivo, Scott Tierno, and Graduate Program Director, Prof. Leon Shterengas, have been especially helpful.

Finally and specially, I would like to deeply thank my beloved wife, Shan Chu, who has always been a great source of strength to me. She has provided me constant love and support all through my study. I am forever indebted to my parents for their

unyielding love, endless patience and great encouragement. I also would like to thank my son, Yiteng Li, for bringing me the delight. The achievements of my dissertation are not possible without their tremendous support.

# Publications

## Conference Publications

- J. Li, C. Wang and Y. Yang, "A versatile platform for mobile data gathering experiments in wireless sensor networks," *The 37th IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2017.

- C. Wang, J. Li, Y. Yang and F. Ye, "A hybrid framework combining solar energy harvesting and wireless charging for wireless sensor networks", *The 35th Annual IEEE International Conference on Computer Communications (INFOCOM)*, 2016.

- C. Wang, J. Li, F. Ye, and Y. Yang, "Improve charging capability for wireless rechargeable sensor networks using resonant repeaters", *IEEE 35th International Conference on Distributed Computing Systems (ICDCS)*, 2015.

- C. Wang, J. Li and Y. Yang, "Low-latency mobile data collection for Wireless Rechargeable Sensor Networks", *IEEE International Conference on Communications (ICC)*, 2015.

- J. Li, Y. Yang and C. Wang, "Mobility assisted data gathering in heterogeneous energy replenishable wireless sensor networks", *The 23rd International Conference on Computer Communications and Networks*, 2014. (Best Paper Runner-Up)

- C. Wang, J. Li, F. Ye and Y. Yang, "Recharging schedules for wireless sensor networks with vehicle movement costs and capacity constraints", *IEEE International Conference on Sensing, Communication and Networking (IEEE SECON)*, 2014.

- J. Li, C. Wang, F. Ye and Y. Yang, "NETWRAP: An NDN based real time wireless recharging framework for wireless sensor networks", *The 10th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2013. (Best Paper Runner up, System Track)

- C. Wang, J. Li, F. Ye and Y. Yang, "Multi-vehicle coordination for wireless energy replenishment in sensor networks," *The 27th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, 2013.

- C. Wang, Y. Yang and J. Li, "Stochastic mobile energy replenishment and adaptive sensor activation for perpetual wireless rechargeable sensor networks," *IEEE Wireless Communications and Networking Conference (WCNC)*, 2013.

- J. Li, M. Zhao and Y. Yang, "OWER-MDG: A novel energy replenishment and data gathering mechanism in wireless rechargeable sensor networks", *IEEE Global Communications Conference (GLOBECOM)*, 2012.

- M. Zhao, J. Li and Y. Yang, "Joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks," *The 23rd International Teletraffic Congress (ITC)*, 2011.

## Journal Publications

- J. Li, C. Wang and Y. Yang, "A versatile platform for network modeling and performance evaluation for mobile data gathering in wireless sensor networks," under submission.

- C. Wang, J. Li, F. Ye, and Y. Yang, "A mobile data gathering framework for wireless rechargeable sensor networks with vehicle movement costs and capacity constraints", *IEEE Transactions on Computers*, vol. 65, no. 8, 2016.

- C. Wang, J. Li, F. Ye, and Y. Yang, "A novel framework of multi-hop wireless charging for sensor networks using resonant repeaters", *IEEE Transactions on Mobile Computing*, no. 1, 2016.

- Y. Yang, C. Wang, and J. Li, "Wireless Rechargeable Sensor Networks - Current Status and Future Trends", *Journal of Communications*, vol. 10, 2015.

- J. Li, Y. Yang and C. Wang, "Mobility assisted data gathering with solar irradiance awareness in heterogeneous energy replenishable wireless sensor networks", *Computer Communications*, vol. 69, 2015.

- Miao Zhao, J. Li and Y. Yang, "A framework of joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks", *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, 2014.

- C. Wang, J. Li, F. Ye and Y. Yang, "NETWRAP: An NDN based real-time wireless recharging framework for wireless sensor networks", *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, 2014.

# Chapter 1

# Introduction

This chapter explains the motivation, design goals, challenges, and contributions of the dissertation.

## 1.1 Motivation

Wireless sensor networks (WSNs) have emerged as an efficient tool to collect spatial and temporal information from the ambient environment[1], and have found widespread application in smart home, industry automation, precision agriculture, environmental monitoring, wildlife study, battlefield surveillance and many other scopes [2–8]. Wireless sensor nodes, usually low-cost and low-power, are deployed to capture the temperature, humidity, vibration and other parameters in the field of interest.

In traditional WSNs, one or more data sinks are statically deployed to collect sensory information from the sensors which are scattered with or without predefined topologies. With the absence of infrastructure for data transmission, the sensors are self-organized and work in ad-hoc manner. The sensory data, generated in the sensors all over the network, is sent to the data sinks via multi-hop relays. Such methods induce significant energy consumption on wireless transmission, especially when sensory data travels long distance (number of hops), which results in unbalanced energy consumption in the networks. Since the sensors are usually powered by batteries with limited capacity, consequently, the energy consumption on sensors varies: the more data transmitted, the faster batteries deplete. When

the sensors that transport large amount of data, such as the sensors close to the data sinks, deplete their energy, data can no longer be transported through the area where these sensors are deployed, thus *energy holes* forms. The formation of energy holes aggravates

The energy constraint exerted by the transmission of sensory data is one of the major problems faced when designing and deploying WSNs as energy holes may disconnect other sensors from the whole networks and whereas decreases network lifetime. Many efforts have been made to tackle the challenge of energy constraint.

A conventional technique is to improve the efficiency of data gathering to reduce energy consumption for data gathering in WSNs from both node and network aspects, such as low power hardware architecture and software implementation[9, 10], low power wireless transmission at physical layer and energy efficient data connection in data link layer[11, 12], end-to-end (sensor-to-data sink) delivery at network layer[13–23] and hierarchical network structure[24–28]. In the past decades, mobile data gathering method has been proposed as a promising solution to such problems[29–54]. One of the earliest applications of mobile data gathering traces back to Operation Igloo White to detect the activities along the Ho Chi Minh Trail during the Vietnam War in 1960's. Aircrafts were dispatched to pick up signals from the 20,000 sensors along the trail and retransmit them to the surveillance center in Thailand. Such methods had not been widely applied in wireless sensor networks due to the high cost of hardware manufacturing and operation (Operation Igloo White costed 1 billion dollars per year). Recent advancement in technologies and reduction in cost, e.g., for semiconductor manufacturing, wireless communications algorithms and network protocols, gave rise to the employment of WSNs with mobile data gathering for many scenarios, such as plant surveillance and environment monitoring. The benefit of mobile data gathering is well recognized for connectivity guaranty, reduced network cost, increased reliability, and improved energy efficiency. Mobile data collectors are employed and move in the networks to obtain data from wireless sensor networks. Since mobile data collectors can move close to the sensors, the path for sensory data relay can be greatly shortened, and consequently less energy is consumed for wireless communication. On the other hand, mobile data collectors can be dispatched to any areas, including the area where the disconnected sensors are connected, thus all the sensors can be utilized to perform sensing tasks.

Energy efficient data gathering methodologies are able to relieve energy constraint in WSNs. However, the batteries in the sensors will deplete ultimately due to the limited energy supply from batteries, thus these methodologies can only extend network lifetime to some extend. In comparison, the application of renewable energy technologies has been considered as a promising solution in tackling with energy constraint in WSNs as it is able to provide sustained energy source. Various energy harvesting methods, either through RF based energy transmission, or by harvesting mechanical, thermal, photovoltaic or electromagnetic energy from the ambient environment, such as mechanical, thermal, photovoltaic or electromagnetic energy, are used to charge the sensors in the network. However, a main problem of energy harvesting is its low efficiency recharging, since the power output of the energy harvesting devices is relatively low [55] compared to power consumption of the node for sensing and communications. Moreover, some of the energy harvesting devices are subject to the variation of the environment, and the power supply cannot be guaranteed. For example, in a solar harvesting system, the power output from the solar panel depends on the solar radiation arrived at the panel which varies according to the time and weather. When power output is low and the battery has little energy left, the sensor node is prone to run out of energy. RF radiation based wireless energy replenishing technique also suffers from low efficiency due to highly lossy transmissions of RF radiation even if directional antennas are used for energy transmitting. Recent breakthrough in wireless energy transmission has brought a renewed interest to renovate the designs for traditional battery-powered sensor networks. Wireless energy transmission technique [56, 57], as an alternative to provide reliable energy to sensor nodes, eliminates the wires and plugs for recharging sensors and circumvents unpredictable behavior of the power source in environmental energy harvesting techniques [73, 74]. By opting wireless energy transmission, sensor network can have indefinitely long operations without battery replacements and we call such a network Wireless Rechargeable Sensor Network (WRSN) [81]. In WRSNs, one or more mobile energy transporters are usually adopted to recharge sensor nodes at different locations[82, 83]. Hence, the mobile energy transporters become the power source in WRSN and the designs of recharge policy is vital to the entire network.

As the result of aforementioned research work, complexity of algorithms and network protocols is growing rapidly in theory, which introduce a great amount of

complexities into network design and deployment. Ammari et al. [119] developed an energy-aware protocol for disseminating data to the mobile sink in WSNs using an information theoretic approach. Zhao et al. [120] applied Space-Division Multiple-Access (SDMA) technique by launching multiple antennas on an MDC such that (distinct) compatible sensors can make concurrent data uploading thereby speeding up data gathering process. Xu et al. [126] addressed event collection problem by leveraging the mobility of the sink node and the spatial-temporal correlation of the event. Most of such works rely on theoretical analysis for system modeling and software simulators for performance evaluation. Although mathematical tools and software simulators are straightforward and easy to utilize, they are generally "inaccurate" given the vast majority of simplifications in mathematical models and imperfection to characterize real world dynamics in software simulators. They often overlook the impact from multiple practical factors so the results may deviate significantly in reality. To this end, several testbeds have been built for more accurate evaluations [131–133, 137]. Nevertheless, many of such testbeds were designed for a particular project in an application specific domain, and it may be infeasible to use the testbeds for other applications.

## 1.2 Design Goals and Research Challenges

Our work is in two categories. We execute theoretical analysis of mobile data gathering and energy. We also design a versatile platform and conduct experiment for mobile data gathering in WSNs.

### 1.2.1 Theoretical analysis and design of mobile data gathering and energy replenishment

The application of mobile data gathering and energy replenishment is an effective method to improve network performance, which is achieved through systematical design of networks by considering the characteristics of sensors, mobile data collectors and energy transporters, and careful scheduling of their actions.

Our design of the framework of mobile data gathering and energy replenishment are targeted to improve WSNs performance in the following categories.

- **High Network Utility:** Network utility characterizes how well the network nodes are utilized as data sources. Different from network throughput which measures the overall amount of data from the network, network utility is a increasing function of the amount of data that is collected from each sensor, thus is a is a properly defined function to characterize the data gathering. In WSNs where sensors usually have limited transmission capability (data rate can be hundreds of, or tens of kilobytes per second, or even lower), and these sensors may become the bottlenecks for data transmission, especially where data aggregates. Data is discarded when congestions occur at these aggregation points, which result in low network utility. Thus network utility is a direct metric to evaluate the effectiveness of data gathering, and a well designed mobile data gathering scheme should be able to reduce such congestion to maximize network utility.

- **Balanced Energy Consumption:** The unbalanced energy consumption due to multi-hop data relay results in energy holes which prohibit transportation of sensory data in the area. By utilizing mobile collectors which work as virtual data sinks, and carefully calculating the locations of these virtual data sinks for different migration tours, routing patterns can be changed accordingly to relieve the sensors with lower energy levels from heavy burden of data relay, thus energy consumption can be balanced throughout the network.

- **Prolonged Network Lifetime:** Mobile energy transporters are scheduled to charge the sensors with little residual energy. When energy neutral condition is satisfied to guarantee that sufficient energy is injected into the network, and the sensors in critical energy situation are charged at higher priority, perpetual operation of the networks can be achieved.

- **High Network Scalability:** The number of sensors charged by energy transporters is limited due to various constraints, such as the time needed to the charge batteries, the distance between the sensors, and the moving velocity of the energy transporters. In wireless sensor networks with high scalability, an energy transporter is able to charge a large number of sensors. Since energy transporters usually have much higher hardware cost than sensors, it is desirable to employ a small number of such expensive devices so as to keep

overall system cost low.

- **Low Operating Cost:** Operating cost is an important metric to characterize the efficiency of network operation. It includes the overhead for data transmission to achieve the benefits brought by mobile data gathering and energy replenishment, which consumes extra energy on sensors. Operating cost also includes the distance of the mobile data collectors' movement which is associated with data latency, and the distance of the energy transporters' movement which impact their responding time to recharge demands. Such distance induces extra energy consumer on mobile data collectors and energy transporters. When operating cost results in performance deterioration that the requirement cannot be satisfied, remedies such as deploying more mobile data collectors or energy transporters is necessary. It is always expected to lower the operating cost as much as possible.

- **Heterogeneous Sensor Networks:** Energy harvesting sensor networks, whose energy availability is constraint, is not capable for large amount of data transmission. However, these networks don't require attendance for battery refill. In the contrary, wireless rechargeable sensor networks can provide higher capability for data forwarding, but needs the attendance of energy transporters, which limits their scalability. We design heterogeneous sensor networks consisting of both energy harvesting and wireless rechargeable sensors, and address efficient data gathering scheme to combine the advantages of two techniques while reducing the impact of their perspective constraints.

Given the unique characteristics, what we have to handle includes, but not limited to, the following open technical challenges and issues in the scheme design and performance optimization for mobile data gathering in WSNs.

- **Limited Energy Capacity at Sensors**. Many sensor with the capability to be recharged use conventional rechargeable batteries (Li-ion batteries, Ni-MH batteries or Ni-Cd batteries) for energy storage. These batteries have limited capacity (less than several thousand Joules) due to the limit of physical size. With such constraint, the time a sensor can work after each recharge is limited. How to calculate energy efficient routing plan according to the dynamic

energy distribution is one of the important factor to be considered for mobile data gathering in WSNs.

- **Limited Charging Capability of Energy Transporters**. The charging capacity of energy transporters is limited by two factors. The first is the limited charging current for the sensors using conventional rechargeable batteries for storage. Due to the charging characteristic of the batteries, the charging current has to be limited to ensure the safety and health of batteries, which implies unnegligible charging time. The second is the limited moving velocity of the energy transporters which costs energy transporters some time to reach to a destination. The combination of these two factors decide the number of sensors that can be charged during a given period, after which some sensors could deplete their energy. How to schedule the charging sequence according to the limited charging capacity is an interesting issue for wireless energy replenishment in WSNs.

- **Energy Distribution Dynamics**. Sensors consumes energy when performs sensing and wireless transmission, and the amount of consumed energy depends on the sensing tasks performed and the amount of data transmitted. The variance of sensing tasks, e.g., in duty-cycle WSNs, and the variance of amount of data being transmitted due to the change of routing scheme, gives rise to the energy distribution dynamic during network operation, which is also impacted by the energy transporters who charges the sensors at different locations. Such dynamics increases the complexity of design of routing scheme as well as charging sequences. How to schedule the movement of mobile data collectors and energy transporters and adjust the data relay in the network to achieve satisfactory performance is critical for network design.

- **Realtime Energy Information Collection**. Energy transporters make decision for recharge scheduling based on the energy information of the network. Such energy information has be delivered to the energy transporters in-time so that they can calculate optimal charging sequence. Moreover, energy levels on some sensors may change drastically, e.g., due to abrupt transmission of large amount of data. If the battery levels drops below a threshold which implies quick depletion of energy, these sensors need to be charged at high-

er priority. Such emergencies should be immediately reported to the energy transporters who are moving in the network. Aggregation and transmission of energy information, especially the emergencies, in an efficient way, is challenging due to the mobility of the energy charger whose locations vary.

### 1.2.2 Experimental platform design for WSNs

The objective of design of the experimental platform is to provide a versatile tool for network modeling and verification. In order to satisfy the requirement from different applications, the testbed is designed to present:

- **High flexibility**: In order to provide a general purpose testbed that is able to implement different application, it should provide flexibility to support customization in data communication in all layers (physical layer, MAC layer and network layer). Mobile platforms, for example, could be robot, vehicle or UAVs for applications in different environment. The control methodologies for different mobile platforms vary, which demands the feasibility to implement different functionality on the mobile element.

- **Extended computing and caching capability on sensor**: Conventional sensors use 8-bit microcontrollers as the computing units, and the memory space is limited to the on-chip RAM. This may be not sufficient to facilitate the increasing on-node computing and storage demand for data processing and caching, such as network coding and data compression. It is necessary to provide more computing power and memory space on the node.

- **High computing capability on the mobile element**: The autonomous operation of the mobile elements demands complex computing, such as physical layer signal processing, optimal routing calculation and autonomous driving. Furthermore, it is beneficial to execute computing tasks on MDCs whenever possible since sensor nodes usually have low computing capability and constrained energy supply due to cost/size limit. As an example, optimal routing and data flow control can be achieved using distributed algorithms by exchanging information among sensors, however, this may require several rounds of message exchange in the network, which could consume a lot

of energy. By migrating the computation to MDC, sensor nodes can reduce energy cost thus the network lifetime can be prolonged. Such migration is important since sensing tasks usually require real-time processing and high computing capability in many applications. Such processing makes a high demand on the computing capability on the mobile element.

## 1.3 Contributions

In this dissertation, we focus on the topic of mobile data gathering and energy replenishment in WSNs. We address several important topics through both theoretical analysis and experiments. Our contributions can be summarized as follows.

- **Joint Design of Mobile Data Gathering and Energy Replenishment in WSNs with Controlled Topology**. We propose an optimal wireless energy replenishment and mobile data gathering mechanism (OWER-MDG) which charges sensor nodes effectively and collects data from the network using a mobile data collector. We study the application of OWER-MDG in WSNs to enable long term operation while maintaining high network utility, and provide an efficient algorithm for optimal rate allocation in such networks. Our numerical results demonstrate that OWER-MDG is effective in guaranteeing network lifetime and network utility. We also provide a guidance on parameter selection for system design.

- **Realtime Energy Information Collection in Wireless Sensor Networks**. We apply *Named Data Networking (NDN)* techniques to gather and deliver energy information to the SenCar. To scale to large network sizes, we divide the network in a hierarchical fashion and the energy information is aggregated bottom-up through different levels. NDN uses names instead of locations to address data, which is a natural match for aggregated energy information that belongs to an area instead of any particular node. Thus the aggregated energy information can be addressed by the area's name. NDN also supports mobile receivers because the routing states in intermediate nodes are constantly updated to follow the movements of receivers, which is important for the SenCar to receive the energy information timely after it changes its location.

- **Optimal Energy Replenishment in Wireless Sensor Networks with Random Topology**. We derive analytic results on *energy neutral* conditions that give rise to perpetual operation in such a recharging framework. We discover that optimal recharging of multiple emergencies using one SenCar is an Orienteering problem with Knapsack approximation. We further address the scalability of the networks by using multiple SenCars. We define the problem of scheduling and coordinating the SenCars to recharge the sensors within their residual lifetimes while minimizing the cost of SenCars as the *Emergency Recharge Optimization with Multiple SenCars* (EROMS) problem. Our extensive simulations demonstrate the effectiveness and efficiency of the proposed framework and validate the theoretical analysis.

- **Mobility Assisted Data Gathering with Solar Irradiance Awareness in Heterogeneous Energy Replenishable Wireless Sensor Networks**. We propose a **Mobility Assisted Data Gathering with Solar Irradiance Awareness** (MADG-SIA)scheme, to achieve balanced energy consumption in WSNs and prolong network lifetime. We construct a MADG-SIA enabled network with three types of devices: the static sensor nodes that are powered by solar panels, the mobile cluster heads that can be wirelessly recharged, and the energy transporters. The network has a hierarchical architecture where sensor nodes are clustered and send data to their corresponding cluster heads. The cluster heads sojourn at different positions in each data gathering period to collect data from surrounding sensor nodes in a one-hop or multi-hop manner. Based on this network architecture, we find optimal positions for anchor points and moving paths for cluster heads to balance energy consumption on sensors and prolong network operating time, and develop a clustering approach and determine the routes for sensor nodes to upload data to cluster heads.

- **A Versatile Platform for Mobile Data Gathering Experiments in Wireless Sensor Networks**. we present a versatile platform that enables performance validation for various mobile data gathering algorithms and network protocols in wireless sensor networks. We focus on architectural support for high performance computing and customizability on mobile data collector and use high-performance Field Programmable Gate Array (FPGA) to handle mobile

computing tasks. Our design also considers wireless communication, memory management, localization, time synchronization and mobility control. As another part of our platform, we enable low-power designs on sensors and enhance their computing capability, memory cache and resource expansions. We provide a complete experimental platform where customization can be achieved in different layers (application, network, MAC, or physical layer), and different mobile data gathering algorithms can be executed efficiently.

- **Experiments of Mobile Data Gathering in Wireless Sensor Networks**. We implement a wildlife monitoring system on our platform. We analyze the reasons that the experiment behaves different from theoretical analysis. Our experimental results demonstrate that real implementations can evaluate many practical performance factors which would have a great impact on the sensing results and are very difficult to fully capture by theoretical models and simulations.

## 1.4 Dissertation Outline

The rest of the dissertation is organized as follows. Chapter 2 exhibits a joint design of wireless energy replenishment and mobile data gathering mechanism using a mobile vehicle in wireless sensor networks with controlled topology. Chapter 3 develops an NDN-based real-time energy information collection protocol for dynamic recharging in wireless sensor networks. Chapter 4 propose a novel real time recharging framework that optimizes the recharging policy under dynamic network conditions. Chapter 5 proposes mobility assisted data gathering with solar irradiance awareness in heterogeneous energy replenishable wireless sensor networks. Chapter 6 presents the design and implementation of a general purpose, flexible platform for mobile data gathering in wireless sensor networks to evaluate network performance and algorithms in a practical setting. Chapter 7 implements a wildlife monitoring system on our platform, and evaluates practical performance factors which would have a great impact on the sensing results and are difficult to fully capture by theoretical models and simulations. Finally, Chapter 8 concludes the dissertation.

The mobile collectors and energy transporters used for data gathering and ener-

gy replenishment could be mobile robots, autonomous vehicles or UAVs equipped with wireless transceivers, energy transmitters, computing units and power supplies. In the rest of this dissertation, they are denoted as *SenCars* for convenience of presentation.

# Chapter 2

# Joint Design of Mobile Data Gathering and Energy Replenishment in WSNs with Controlled Topologies

Current study on prolonging lifetime for wireless sensor networks (WSNs) mainly focuses on two techniques. The first technique is to reduce energy consumption of sensor nodes, while the second technique is to recharge sensor nodes by harvesting energy from the ambient environment or RF based energy transmission. However, neither of these two techniques are able to guarantee network lifetime and network performance. In order to achieve perpetual operation for WSNs while providing high network utility, in this chapter we propose an optimal wireless energy replenishment and mobile data gathering mechanism (OWER-MDG) which charges sensor nodes effectively and collects data from the network using a mobile vehicle (SenCar). We study the application of OWER-MDG in WSNs and provide an efficient algorithm which maximizes network utility. Our numerical results demonstrate the performance advantage of OWER-MDG and provide a guidance on parameter selection for system design.

## 2.1 Introduction

Wireless sensor networks (WSNs) have received considerable attention due to their applications in many areas, such as agriculture, industry and military. Challenging problems arise in the design and deployment of WSNs, among which energy consumption is one of the most important factors to be considered. As most conventional sensor nodes are powered by batteries, the networks can only work for the limited period of battery lifetime. The nodes that have depleted their energy have to be replaced to enable the network to work as long as desired.

Current study on prolonging lifetime for wireless sensor networks mainly focuses on two techniques. A conventional technique is to reduce energy consumption in WSNs from both node and network aspects, such as low power hardware architecture and software implementation, power efficient wireless communications techniques, MAC protocols, routing strategies and battery aware designs. Energy replenishment for wireless sensor nodes has emerged as a new technique to relief the power constraint by recharging sensor nodes through RF based energy transmission, or by harvesting mechanical, thermal, photovoltaic or electromagnetic energy from the ambient environment. However, a main problem of energy harvesting is its low efficiency recharging, since the power output of the energy harvesting devices is relatively low [55] compared to power consumption of the node for sensing and communications. Moreover, some of the energy harvesting devices are subject to the variation of the environment, and the power supply cannot be guaranteed. For example, in a solar harvesting system, the power output from the solar panel depends on the solar radiation arrived at the panel which varies according to the time and weather. When power output is low and the battery has little energy left, the sensor node is prone to run out of energy. RF radiation based wireless energy replenishing technique also suffers from low efficiency due to highly lossy transmissions of RF radiation even if directional antennas are used for energy transmitting.

It is foreseeable that energy replenishment could be a good solution for energy constrained WSNs to achieve perpetual operation with performance guaranty if the efficiency of energy replenishment for sensor nodes could be improved. We have found two new techniques that would make such improvement possible. The first technique is the high efficiency non-radiative wireless power transmission over

midrange[1]. Different from RF radiation based wireless energy transmission, this technique transfers energy through strongly coupled magnetic resonances, and the efficiency of transferring 60 watts of power over 2 meters is as high as $40\%$ [56, 57]. Intel has also demonstrated that 60 watts of power can be transferred over a distance of up to 2 to 3 feet with efficiency of $75\%$ [58]. The second technique is a novel battery material for ultra-fast charging. Scientists from MIT has implemented ultra-fast charging in $LiFePO_4$ by creating a fast ion-conducting surface phase through controlled off-stoichiometry [59]. Without the problem of slow charging for Nickel-Metal Hydride (NiMH) and Lithium-ion (Li-ion) batteries[2] and the problems of high leakage and limited capacity for supercapacitors, the new material brings high energy density and can be charged at as high as 400C, thus shortens the time to fully charge a battery to few seconds.

Inspired by these breakthroughs, in this chapter we propose an Optimal Wireless Energy Replenishment and Mobile Data Gathering mechanism, which is abbreviated as OWER-MDG. In this mechanism, a mobile vehicle, referred to as a *SenCar*, travels through the area where the network is deployed, visits some locations, named as *anchor points*, stays for a period of *sojourn time* to charge neighboring sensor nodes and collect data from the network. Compared with energy harvesting WSNs, OWER-MDG is immune to environment change and achieves higher network performance. It also overcomes the low efficiency and short range energy transmission of RF radiation based wireless energy replenishing technique. To be specific, we focus on WSNs with regular topologies, such as equilateral triangles, squares and hexagons. Compared with random topologies, regular topologies have many advantages, including wider coverage, simple and efficient wireless communication and networking protocols, and significantly increased end-to-end throughput in multi-hop connections [60–62]. Thus for applications where the deployment of sensor nodes is controllable, e.g., precision agriculture, WSNs with regular topologies are commonly used for better performance. Without loss of generality, we focus on the WSNs with a square topology in this chapter.

[1]Midrange refers to that the distance between the transmitter and the receiver is larger than the size of devices by a factor of at least 2 to 3 [56].

[2]For Li-ion batteries, the optimum charging rate that achieves the best performance is 1C, while for NiMH batteries, the optimum charging rate is even lower, where C is determined by the nominal capacity of the battery. For a battery of 1000 mAh, 1C = 1000mA.

## 2.2 Related Work

In this section, we briefly review some related work in the literature, which includes the work on energy replenishment in wireless sensor networks, and the work on mobile data gathering in wireless sensor networks.

### 2.2.1 Energy Replenishment

In recent years, energy replenishment has been considered for wireless sensor networks in the literature. Wireless sensor networks with redundantly deployed rechargeable sensors were studied in [63], which also addressed the problem of maximizing a global coverage metric by dynamically activating sensor nodes. In the work, a distributed threshold based sensor node activation policy was proposed and the performance was demonstrated to be very close to that of the globally optimal policy for both overlapped and non-overlapped sensors. The performance of multi-hop radio networks with renewable energy sources was modeled and characterized in [64], where an energy-aware routing algorithm was provided and proved to be asymptotically optimal with respect to the number of nodes in the network. A joint study of energy management and resource allocation problem for energy-harvesting sensors was presented in [65]. In this work, the optimal sampling rate based on the average energy replenishment rate was explored, and a local algorithm was presented for each sensor to adapt the sampling rate according to short term fluctuations in recharging, with the objective of maintaining the battery at a target level. An adaptive control theory based approach that achieves energy neutral operation, performance maximization and duty cycle stability was proposed in [67] by taking into account of the variability of the harvested environmental energy. RF radiation based energy replenishment for sensor networks was studied in [68] by developing and implementing a wireless charging system which charges sensor nodes by a mobile charger. Wireless recharging with a mobile vehicle in sensor networks was also studied in [69], in which the ratio of the vehicle's vacation time over the cycle time is maximized using a near-optimal solution.

### 2.2.2 Mobile Data Gathering

Mobile data gathering is especially suitable for WSNs where the terrains of the networks can be reached by mobile data collectors (MDCs). There has been much work in this area. Based on the fact that sensors close to the static data sink deplete energy much faster than others and become the bottleneck for data uploading, it was proposed in [33, 43, 70, 71] to use mobile data collectors to achieve the uniformity of energy consumption. A mobile data gathering scheme was proposed in [33] in which mobile collectors move along parallel straight lines and gather data from nearby sensors with multi-hop transmissions. A rendezvous design was provided in [43], which aims to find a set of rendezvous points to be visited by the mobile collector within a delay bound, while the network cost incurred in transmitting data from sources to rendezvous points is minimized. A joint design of space-division multiple access technique and mobile data gathering was explored in [70], and the algorithms proposed significantly improve data gathering efficiency. A distributed routing algorithm was presented in [71] to maximize network lifetime, in which the mobile base station visits specified locations for multi-hop data gathering. While demonstrated to improve network performance, mobile data gathering generally requires extra resources to build MDCs. However, the cost is not significant compared with the overall cost of WSNs which usually consist of thousands of sensor nodes, and thus it is reasonable to utilize mobile data gathering in WSNs.

## 2.3 Framework of OWER-MDG

In this section, we provide an overview of the proposed OWER-MDG mechanism. The structure of an OWER-MDG enabled wireless sensor network is illustrated in Fig. 2.1. The entire network consists of two types of devices: SenCars that work as energy transporters and data collectors, and wireless sensor nodes that perform surveillance. Each SenCar is equipped with a high capacity rechargeable battery, an oscillator that converts the direct current (DC) from the battery into high frequency alternating current (AC), and a resonant coil as the transmitter. The battery can provide sufficient energy for the operation of the SenCar as well as fully charging the selected sensor nodes. Each sensor has a receiver which consists of a receiver coil that is tuned to resonate at exactly the same frequency as the transmitting coil

on SenCar, an AC/DC converter that is controlled by the battery charger, and a rechargeable battery with ultra-fast charging rate.

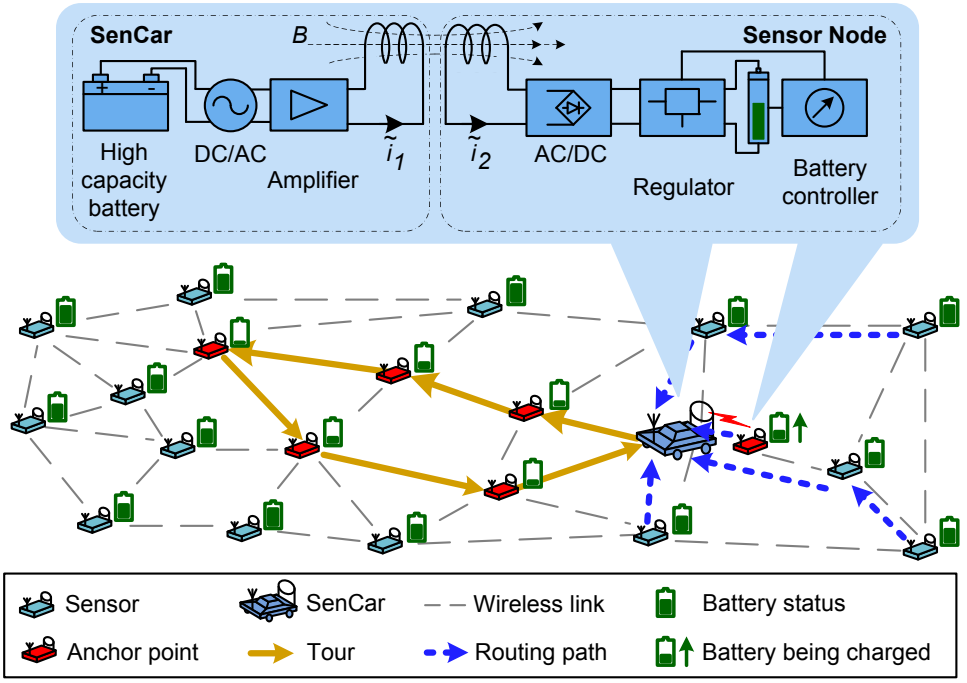



Figure 2.1: System structure of an OWER-MDG enabled wireless sensor network.

For each charging and data collecting tour, which is referred to as a *migration tour*, the SenCar travels along a planned path consisting of several anchor points, as illustrated by the yellow edges in Fig. 2.1. In this chapter, positions of some sensor nodes are selected as anchor points so that these nodes can be charged with high efficiency and upload sensing data to the SenCar at the same time. When the SenCar arrives at an anchor point, DC current is fed to the oscillator and converted to high frequency AC current. Driven by this AC current, an oscillating magnetic field is induced around the transmitter coil. This magnetic field then induces an AC current in the receiver coil of the sensor node placed at the anchor point. The AC/DC converter generates DC current and regulates it to properly charge the rechargeable battery of the sensor node. The charging process is performed concurrently with data collection. When the SenCar returns to the base station to upload data to the data analyzer, the high capacity battery can be charged with constantly available energy sources. By carefully selecting the nodes to be charged for each migration

tour, the sensor nodes can always be supplied with energy when necessary and the system is able to run as long as desired.

Note that since ultra-fast charging material is applied for the rechargeable batteries, the charging time is relatively shorter compared with the time for the SenCar to gather data from nearby sensor nodes. Therefore, the additional charging task does not affect the data gathering operation of the SenCar.

## 2.4 Optimal Wireless Energy Replenishment and Mobile Data Gathering

In this section, we present a possible implementation for OWER-MDG in WSNs. Our objective is to jointly select the sensors to be charged and find the optimal data gathering scheme, such that energy consumption on data relay is minimized for perpetual operations of the network, while maximizing network performance. This is achieved in two steps: we first find anchor points for each migration tour to charge sensor nodes, then we allocate data rates according to the energy status of the nodes for network performance improvement.

### 2.4.1 System Model

Consider an area covered by $N^2$ sensor nodes which are regularly deployed at square grids with distance $d$, and each node is within its one-hop neighbors' transmission range $r$. Since the data are sent to the anchor point in a multi-hop manner, which consumes a lot of energy on data relay, reducing the number of hops can extend sensors' battery lives. As the deployment of the wireless sensor networks is regular, we can plan paths for the SenCar to charge the nodes and collect data, as illustrated in Fig. 2.2.

Initially, the area is divided into $p$ *regions* (labeled as 1, 2, $\cdots$, $p$) by $p$ parallel *tracks* (labeled as 1, 2, $\cdots$, $p$) with track 1 positioned on the left boundary of the area. The $i$-th region ($1 \leq i < p$) is bounded by the $i$-th track, the $(i+1)$-th track and the boundaries of the area, while the $p$-th region is bounded by the $p$-th track and the boundaries of the area. The $p$ tracks, which go along the sensor nodes with the same longitude, are connected by $(p-1)$ *segments* and form the SenCar's moving

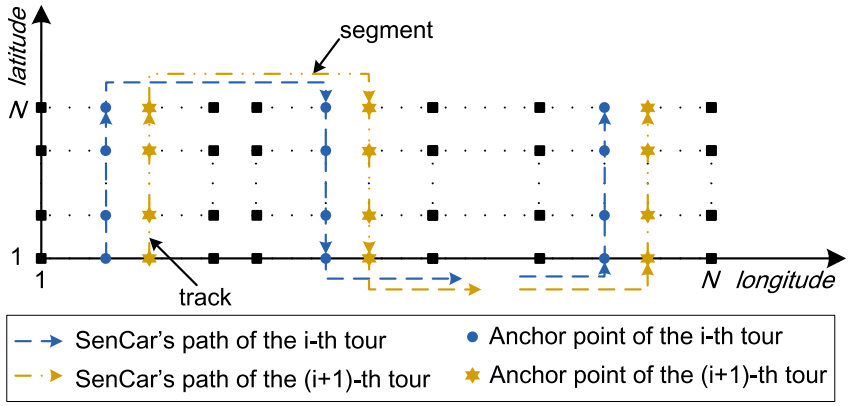



Figure 2.2: Energy replenishment and data gathering in a network with square topology.

path with a continuous square wave shape, e.g., the blue dashed lines in Fig. 2.2. The width of region $i$, $w_i$, is calculated as follows so that the area is divided into as many regions as possible to minimize energy consumption on packet relay.

$$w_i = \begin{cases} w_1, & i = 1, 2, \ldots, t \\ w_2, & i = t+1, \ldots, p \end{cases}$$

where $w_1$, $w_2$, $t$ and $p$ satisfy the following constraints

$$(tw_1 + (p - t - 1)w_2 + p(N-1))\, d \leq L \tag{2.1}$$

$$1 + tw_1 + (p - t)w_2 = N \tag{2.2}$$

$$w_1 - w_2 \leq 1 \tag{2.3}$$

Eq. (2.1) ensures that the tour length is bounded by the given maximum tour length $L$, and Eq. (2.2) and Eq. (2.3) ensure that nodes are separated into groups as evenly as possible. The positions of nodes on the tracks are selected as anchor points where the SenCar stops to charge the nodes and collect data at the same time. In each consecutive migration tour, the SenCar's path is shifted to the increasing direction of the longitude by 1 unit, and a track which is shifted out of the region is shifted in from the left and reconnected to the path. In order to reduce energy consumption on packet relay, each node is scheduled to transmit data to the SenCar when it visits the nearest anchor point. As the network topology is regular, the route for each

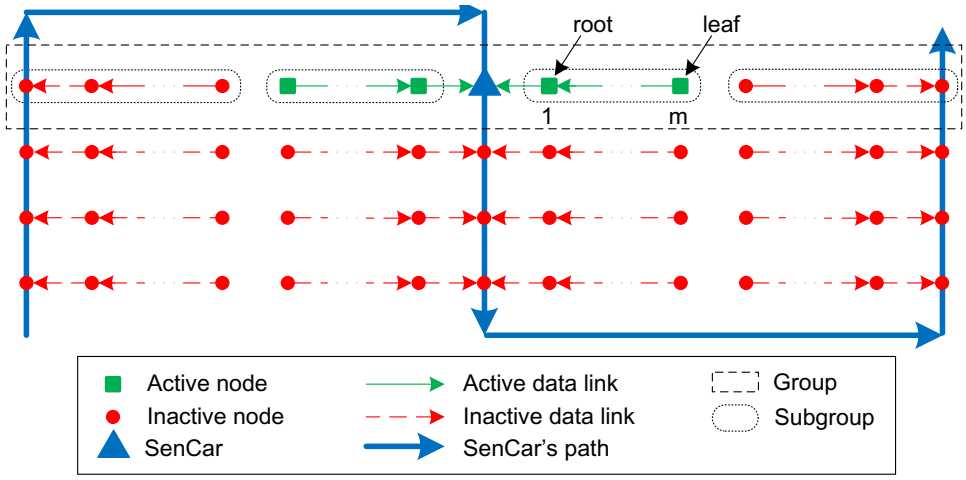



Figure 2.3: Each node transmits data to the nearest anchor point. In a network with a square topology, such a scheme results in parallel data transmission.

node to relay its packets to the nearest anchor point is a straight line vertical to the tracks, and the nodes with the same longitude transmit data in the same pattern, as shown in Fig. 2.3. Therefore, the study of the entire network can be simplified to the study of the nodes with the same latitude, referred to as a *group* and denoted as $\mathcal{G}$. Depending on the data transmission activity of nodes, the nodes in a group can be divided into the nodes located at anchor points and non-anchor-point nodes. The non-anchor-point nodes are further divided into *subgroups* and the nodes in a subgroup send data to the SenCar at the same anchor point in the same direction. To ease the presentation, a subgroup consisting of $m$ nodes is considered as a tree, in which the node that is one-hop away from the SenCar is the $root$ and denoted as node 1, while the node that is farthest to the root is the $leaf$ and denoted as node $m$. Since the subgroups operate independently, the optimal activity of a group can be achieved by optimizing the activity of each subgroup belonging to the group.

### 2.4.2 Problem Formulation

We have seen that the proposed data transmission scheme has the potential to reduce the number of hops, thus energy consumption, in multi-hop data relay. Now we further consider how to achieve perpetual operation of the network. We formulate this problem as a network utility maximization problem. The notations are listed in

Table 2.1.

Table 2.1: Notations used in formulation of OWER-MDG.

| Notation | Definition |
|---|---|
| $\mathcal{S}$ | Set of sensors in the network. |
| $\mathcal{G}_i$ | Set of sensors in group $i$. |
| $\mathcal{V}_i^{(k)}$ | Set of sensors in subgroup $i$ in the $k$-th migration tour. |
| $\mathcal{X}_i^{(k)}$ | Set of sensors located at the anchor points in group $i$ in the $k$-th migration tour. |
| $root$ | The node that is one-hop away from the anchor point in a subgroup. |
| $leaf$ | The node that is farthest to the anchor point in a subgroup. |
| $\mathcal{A}_i^{(k)}$ | Set of ancestor nodes of sensor $i$ in the $k$-th migration tour, i.e., the nodes that relay data from node $i$ towards the anchor point. |
| $\bar{B}_i$ | Full battery energy state of sensor $i$. |
| $b_i^{(k)}$ | Battery status of sensor $i$ in the $k$-th migration tour. |
| $\mathcal{K}$ | Set of migration tours. |
| $\mathcal{P}^{(k)}$ | Set of anchor points selected for the $k$-th migration tour. |
| $L$ | Maximum length of SenCar's migration tour. |
| $e_{i,j}$ | Energy consumed for node $i$ to transmit a unit flow on link $(i, j)$. |
| $r_i^{(k)}$ | Data rate of node $i$ in the $k$-th migration tour. |
| $r_{max}$ | Maximum data rate for all the nodes. |
| $f_{i,j}^{(k)}$ | Flow rate over link $(i, j)$in the $k$-th migration tour. In a subgroup with $m$ nodes, $f_{1,0}^{(k)}$ refers to the flow on the link between the root and the SenCar, and $f_{m+1,m}^{(k)}$ is always set to 0. |
| $l_{i,j}$ | Link capacity of the link between node $i$ and node $j$. |
| $\tau^{(k)}$ | Sojourn time of SenCar at each anchor point in the $k$-th migration. |
| $\sigma$ | Portion of remaining energy on a node that can be used for data transmission in each migration tour. |

We use *data utility* function $U_i(\cdot)$ to characterize the contribution of node $i$ to the overall network performance. $U_i(\cdot)$ is defined as a monotonically increasing, strictly concave and twice-differentiable function of the total amount of data collected from node $i$ in a migration tour. There are several typical forms that can be

used for $U_i(\cdot)$, such as $U_i(\cdot) = q_i \log\left(1 + r_i^{(k)}\tau\right)$, where $q_i$ is the weight of utility at node $i$. Without loss of generality, we consider a homogeneous network and all the sensor nodes have the same importance. The utility function can be written as $U_i(\cdot) = \log\left(1 + r_i^{(k)}\tau\right)$. Our objective can be achieved by solving the network utility maximization problem (NUM)s, which is described by

$$NUM : \sum_{i \in \mathcal{S}} U_i(\cdot)$$

where $\mathcal{S}$ is the set of sensor nodes in the network, and $r_i$ is the data rate of node $i$. Since the groups have exactly the same behavior and the subgroups in the groups work independently, network utility can be calculated by

$$\begin{aligned}\sum_{i \in \mathcal{S}} U_i(\cdot) &= N \times \sum_{i \in \mathcal{G}_j} U_i(\cdot) \\ &= N \times \left(\sum_{\mathcal{X}_j} U_i(\cdot) + \sum_{\mathcal{V}_k \subset \mathcal{G}_j} \sum_{i \in \mathcal{V}_k} U_i(\cdot)\right)\end{aligned}$$

where $j \leq N$ is a positive integer. According to our data gathering scheme, the sensor nodes located at the anchor points upload data to the SenCar directly and their data rates are constants, and $\sum_{\mathcal{X}_j} U_i(\cdot)$ is also a constant. The NUM problem can be solved by maximizing the utility of each subgroup separately. Denote the node that is $m$-hops away from the anchor point as node $m$. Refer to node $(i-1)$ as the *parent* of node $i$, and nodes $1, 2, \ldots, (i-1)$ as the *ancestors* of node $i$. Then the maximization of subgroup utility can be rewritten as

$$\textbf{Maximize:} \quad \sum_{i \in \mathcal{V}_j^{(k)}} U_i(\cdot) \tag{2.4}$$

**Subject to**

$$r_i^{(k)} + f_{i+1,i}^{(k)} = f_{i,i-1}^{(k)}, \forall i \in \mathcal{V}_j^{(k)}, \forall k \in \mathcal{K} \tag{2.5}$$

$$f_{i,i-1}^{(k)} \tau^{(k)} e_{i,i-1}^{(k)} < \sigma b_i^{(k)}, \forall i \in \mathcal{V}_j^{(k)}, \forall k \in \mathcal{K} \tag{2.6}$$

$$r_i^{(k)} \in \mathcal{R}+, r_i^{(k)} < r_{max}, \forall i \in \mathcal{V}_j^{(k)}, \forall k \in \mathcal{K}$$

where $b_i^{(k)}$ is the battery status of sensor $i$ in the $k$-th migration tour and can be

calculated as follows

$$b_i^{(k)} = \begin{cases} \bar{B}_i & i \in \mathcal{P}^{(k)} \\ b_i^{(k-1)} - f_{i,i-1}^{(k-1)} \tau^{(k-1)} e_{i,i-1}^{(k-1)} & \text{otherwise} \end{cases} \tag{2.7}$$

In the above formulation, flow conservation constraint (2.5) states that at each sensor, the aggregated outgoing link flow rate equals the local data rate plus the incoming link flow rates, and energy constraint (2.6) specifies that the energy cost of each sensor in a time interval should be bounded by its energy budget, which is part of the current battery energy.

### 2.4.3 The Deterministic Solution

In this subsection, we provide a solution for the network utility maximization problem defined above. Due to the regularity of the network topology, both routing and scheduling in the network are deterministic in each migration tour, thus we are able to find a deterministic solution for this network utility maximization problem. Based on the formulation in the previous subsection, the problem can be simplified to data rate allocation for the sensor nodes to maximize the utility of each subgroup. Flow constraint (2.5) and energy constraint (2.6) show that the sum of data rates of a node and its descendants are bounded by its energy budget and link capacity. We have the following theorem regarding the optimal data rate allocation under such constraint.

**Theorem 1.** *Let $\overline{X} = [x_1, x_2, \cdots, x_n]$ be an $n$-dimensional vector, where $\forall i$, $x_i \leq a$ ($a$ is a constant). Then we have $\sum_i x_i \leq K \leq n \cdot a$. Let $f(x)$ be a monotonically increasing, strictly concave and twice differentiable function on its feasible region. Let $g(\overline{X}) = \sum_{x_i \in \overline{X}} f(x_i)$. Let $g_{max}$ denote the maximum value of $g(\overline{X})$. Then $g(\overline{X}) = g_{max}$ if and only if $x_i = \frac{K}{n}, \forall x_i \in \overline{X}$.*

*Proof.* Since $f(x)$ is monotonically increasing, $g(\overline{X}) = g_{max}$ only if $\sum_i x_i = K$. We have the Lagrange function as follows

$$\begin{aligned} \Lambda(\overline{X}, \lambda) &= g(\overline{X}) + \lambda(\sum_{x_i \in \overline{X}} x_i - K) \\ &= \sum_{x_i \in \overline{X}} f(x_i) + \lambda(\sum_i x_i - K). \end{aligned}$$

Taking derivative of $x_i$, $\forall x_i \in \overline{X}$, we have

$$\begin{aligned}\frac{\partial \Lambda}{\partial x_i} &= \frac{\sum_{x_i \in \overline{X}} f(x_i)}{x_i} + \frac{\partial(\lambda(\sum_i x_i - K))}{\partial x_i} \\ &= \frac{\partial f(x_i)}{\partial x_i} + \lambda = 0.\end{aligned}$$

Since $f(x)$ is strictly concave and twice differentiable, $g(\overline{X}) = g_{max}$ when $x_i = \frac{K}{n}, \forall x_i \in \overline{X}$. □

According to constraint (2.5) and (2.6), the maximum number of $f_{i,i-1}^{(k)}$, denoted as $\tilde{f}_{i,i-1}^{(k)}$, is limited by the link capacity of link $(i, i-1)$ and battery status of sensor $i$, i.e.,

$$\tilde{f}_{i,i-1}^{(k)} = \min\left(l_{i,i-1}, \frac{\sigma b_i^{(k)}}{\tau^{(k)} e_{i,i-1}}\right). \tag{2.8}$$

Let node $j$ be an ancestor of node $i$. Given the maximum flow rate on the outgoing link of node $j$, i.e., $\tilde{f}_{j,j-1}^{(k)}$, and the flow rate on the incoming link of node $i$, i.e., $f_{i+1,i}^{(k)}$, by Theorem 1, the optimal data rate of node $i$ determined by this ancestor is the average of $f_{i+1,i}^{(k)}$ (excluding $f_{i+1,i}^{(k)}$ which node $j$ must forward) among nodes $j, j+1, \ldots, i-1, i$. The optimal data rate of node $i$ can be calculated as follows:

$$r_i^{(k)} \leq \min\left(\min_{j \in \mathcal{A}_i \bigcup \{i\}} \left(\frac{\tilde{f}_{i,i-1}^{(k)} - f_{i+1,i}^{(k)}}{|i-j|+1}\right), r_{max}\right). \tag{2.9}$$

In order to maximize network utility, the upper bound of $r_i^{(k)}$ is set. Let node $n$ be the leaf node of a subgroup. Set $f_{n+1,n}^{(k)}$ to 0 as node $n$ does not have a child. Starting from node $n$, the data rate of the nodes in the subgroup can be induced by calculating the rates of the parent nodes of those already determined nodes iteratively. The deterministic solution is described in Algorithm 1. It calculates the data rates for network utility maximization in subgroups. In the loop from line 7 to line 9, the maximum flow rates on the outgoing links of all the nodes in subgroups are calculated. The loop starting from line 10 calculates the data rates accordingly. Flow to the next node and the battery status are updated after the data rate of each node is calculated.

**Algorithm 1**: Calculating data rates for sensor nodes in a group to maximize network utility for $K$ successive migration tours

**Input:** Full battery status of sensors $\{\bar{B}_i, i \in \mathcal{G}_1\}$; energy to transmit a unit flow over link $(i, i-1)$, $\{e_{i,i-1}, i \in \mathcal{G}_1\}$; link capacity between a node and its parent $\{l_{i,i-1}, i \in \mathcal{G}_1\}$; maximum tour length $L$; number of nodes in a group $N$; sojourn time of SenCar $\tau$; portion of remaining node energy $\sigma$.

**Output:** Sensor data rates in $K$ migration tours $\{r_i^{(k)}, i \in \mathcal{G}_1, k = 1, 2, \cdots, K\}$.

$b_i^{(1)} \leftarrow \bar{B}_i$, $i = 1, 2, \cdots, N$
Calculate $w_1, w_2, t$ and $p$
Initialize $\mathcal{P}$
**for** $k = 1$ to $K$ **do**
  **for all** subgroups **do**
    $f_{leaf+1,leaf}^{(k)} \leftarrow 0$
    **for** $i$ from leaf to root **do**
      $\tilde{f}_{i,i-1}^{(k)} = \min(\frac{\sigma b_i^{(k)}}{\tau^{(k)} e_{i,i-1}}, l_{i,i-1})$
    **end for**
    **for** $i$ from leaf to root **do**
      $r_i^{(k)} = \min\left(\min_{j \in \mathcal{A}_i \bigcup \{i\}} \left(\frac{\tilde{f}_{j,j-1}^{(k)} - f_{i+1,i}^{(k)}}{|i-j|+1}\right), r_{max}\right)$
      $f_{i,i-1}^{(k)} \leftarrow f_{i+1,i}^{(k)} + r_i^{(k)}$
      **if** node $i$ is located at an anchor point **then**
        $b_i^{(k+1)} \leftarrow \{\bar{B}_i\}$
      **else**
        $b_i^{(k+1)} \leftarrow b_i^{(k)} - f_{i,i-1}^{(k)} \tau^{(k)} e_{i,i-1}$
      **end if**
    **end for**
  **end for**
  Update $\mathcal{P}$;
**end for**

Table 2.2: Parameter settings for performance evaluation of OWER-MDG.

| Parameter | Default value | Parameter | Default value |
|---|---|---|---|
| $\bar{B}_i$ | 2100mAh | $e_{i,j}$ | 0.3mJ/Kbit |
| $L$ | 4230m | $l_{i,j}$ | 250Kbit |
| $\tau^{(k)}$ | $60s$ | $\sigma$ | 0.4 |

Since the SenCar can easily obtain the inputs of the above algorithm, it is viable to let the SenCar execute the algorithm and send the allocated data rates to sensor nodes. Such a mechanism simplifies network operation and eliminates the overhead introduced by distributed algorithms that spend extra energy consumption on information exchanging among sensor nodes.

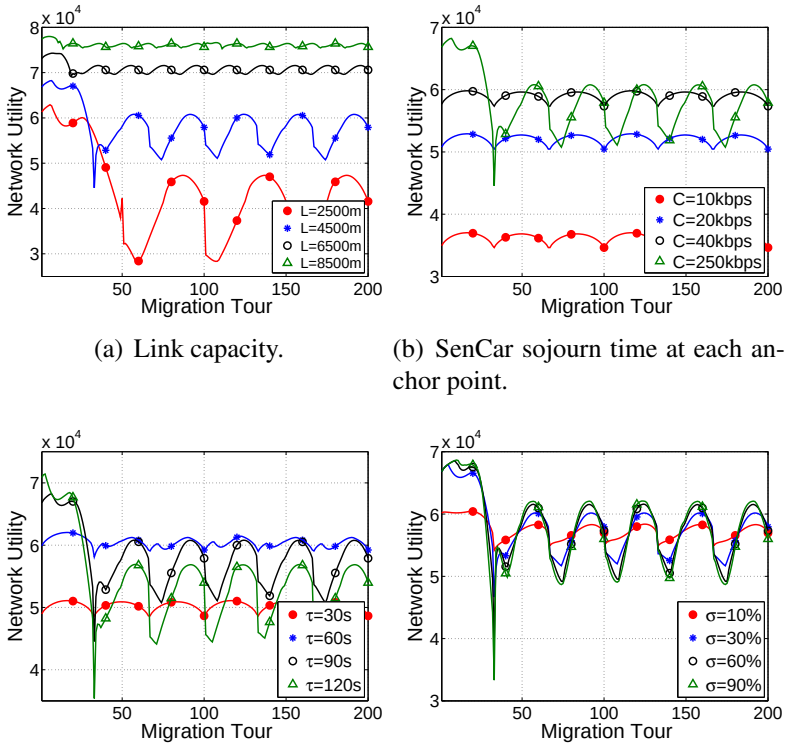
## 2.5 Numerical Results

Having described the utility maximization algorithm for OWER-MDG, in this section, we evaluate the impact of critical parameters in OWER-MDG through simulations, with the goal of providing a guidance on parameter selection when designing and deploying such WSNs.

In the simulation, we generate a network consisting of 10,000 wireless rechargeable sensor nodes, which are evenly distributed as a $100 \times 100$ grid over a $1000m \times 1000m$ area. The parameters we study include the capacity of the links between any two nodes $l_{i,j}$, the time SenCar sojourns at each anchor point $\tau^{(k)}$, the portion of available energy for data transmission in each migration tour $\sigma$, and the maximum length of migration tour $L$. The default values of the parameters used in the simulations are listed in Table. 2.2. The simulation is run for 200 migration tours for each parameter setting.

Note that the increase of $L$ may result in a smaller region width, which is a monotonically decreasing function of $L$. A smaller width of region means a shorter interval that sensor nodes get recharged and it takes fewer hops for the data to be uploaded, which reduces energy consumption on data relay and increases the overall network utility. Fig. 2.4(a) illustrates the impact of $L$, where larger network utility with less fluctuation is observed as $L$ increases from $2500m$ to $8500m$.

After the area is properly partitioned into regions, the data rate of a node is determined based on the node capacities of the its ancestors, which are functions




(a) Link capacity.

(b) SenCar sojourn time at each anchor point.

(c) Available energy for data transmission in each migration tour.

(d) Maximum migration tour length.

Figure 2.4: Impact of different parameters on network utility.

of the capacities of the outgoing links of ancestors, $l_{i,j}$, the SenCar sojourn time, $\tau$, and the portion of the battery energy for data transmission during this period, $\sigma$. In Fig. 2.4(b)-(d), we examine the impact of these parameters separately. Fig. 2.4(b) shows the variation of network utility due to the change of $l_{i,j}$. A low $l_{i,j}$ allows a small amount of data to be transmitted and results in low network utility. As $l_{i,j}$ increases, more data are gathered from the network for each migration tour. As the nodes closer to the SenCar consume more energy for data relay, the increase of $l_{i,j}$ also results in the increase of unbalanced energy consumption among sensors due to intense data transmission. When $l_{i,j}$ is high enough and the energy budget for data transmission of some sensor nodes during a sojourn time is relatively low, the node capacities of these nodes are limited by their energy status instead of $l_{i,j}$. The network utility fluctuates intensively when link capacity is very high and no longer affects the rates of nodes. Fig. 2.4(c) shows how SenCar sojourn time at each anchor point affects network utility. A very short sojourn time (30s in the Fig. 2.4(c)) results in less network utility because a small amount of data are collected from each node $(r_i^{(k)}\tau)$. As sojourn time increases, network utility increases as well. When sojourn time is larger than a certain value, network utility appears to have big fluctuation as much energy is consumed on data relay for the current migration tour and less energy is left for future migration tours. Fig. 2.4(d) illustrates the impact of the portion of available energy for data transmission in each migration tour. A larger portion of node energy set for data transmission leads to more energy consumption for each migration tour and less energy is left for future migration tours before the nodes get recharged. The network utility fluctuates violently as a result.

Evaluation results in Fig. 2.4(a)-(d) provide some insights for parameter selections. By carefully selecting the values of parameters, we can design WSNs with desired performance for specific applications. The evaluation results also demonstrate that OWER-MDG is very effective in achieving perpetual operation in wireless sensor networks while maintaining high network utility. Compared with solar harvesting sensor networks in which the data output from a sensor node drops to zero quickly after solar irradiance becomes unavailable, data utility in OWER-MDG is guaranteed to be above a threshold for each sensor in each migration tour, and only some fluctuation of network utility is observed.

## 2.6 Conclusions

It is challenging to achieve perpetual operation for wireless sensor networks while maintaining high network utility. In this chapter, we have proposed a novel wireless energy replenishment and mobile data gathering architecture (OWER-MDG) for wireless sensor networks that enables long term operation while maintaining high network utility, and presented a deterministic algorithm for optimal rate allocation in such networks. We have demonstrated that OWER-MDG is effective in guaranteeing network lifetime and network utility. We have also provided a guidance on parameter selection for designing and deploying such WSNs through numerical results.

# Chapter 3

# Realtime Energy Information Aggregation in Wireless Sensor Networks

In Chapter 2, we present a methodology for energy replenishment in WSNs with controlled topology. In such networks, the pattern of data transmission is predefined and the energy consumption on the sensors are predictable, thus the SenCar is able to recharge the sensors as needed. In some networks where sensors are randomly deployed, the routing scheme varies and it's very difficult, if not impossible, to calculate the residual energy on the sensors. In order for the SenCar to charge the network efficiently to achieve perpetual operation, it is necessary to obtain energy information of the network for the SenCar to make optimal decision for recharge. Each time the SenCar collects such information and decide which sensors to be charged. Conventional data gathering schemes don't qualify for this task as data from each sensor will be sent to the SenCar at its current location, which results in large amount of data transmissions and large amount of energy consumption.

In this chapter, we propose NETWRAP, an NDN based Real Time Wireless Recharging Protocol for dynamic recharging in wireless sensor networks. We leverage concepts and mechanisms from NDN (*Named Data Networking*) to design a set of protocols that continuously gather and deliver energy information to the mobile vehicle, including unpredictable emergencies, in a scalable and efficient manner.

## 3.1 Introduction

Wireless energy transmission techniques [56, 57] have great potential to prolong the lifetime of wireless sensor networks. With such techniques, the energy of wireless sensor nodes can be replenished over the air without any wire or plug, and more reliable energy sources can be provided than those from environmental energy harvesting techniques [73, 74]. A SenCar operating as the energy transporter can move around and recharge nodes conveniently. How the SenCar collect energy information from the network has not been well studied, and it has been assumed that the sensors send their residual energy information piggybacked with sensory data to the SenCar. Since the SenCar has to move to different anchor points for data collection, which usually take some time depending on the moving speed of the SenCar, the energy situation may have changed when the SenCar receives the last sensory data at the end of a migration tour.

In order to provide the SenCar with most up-to-date energy information, we apply *Named Data Networking (NDN)* [84] techniques to gather and deliver real-time energy information. To scale to large network sizes, we divide the network in a hierarchical fashion and the energy information is aggregated bottom-up through different levels. NDN uses names instead of locations to address data, which is a natural match for aggregated energy information that belongs to an area instead of any particular node. Thus the aggregated energy information can be addressed by the area's name. NDN also supports mobile receivers because the routing states in intermediate nodes are constantly updated to follow the movements of receivers. This is important for the SenCar to receive the energy information timely after it changes its location.

The protocols satisfy both normal and emergency recharging needs for a mobile vehicle. To the best of our knowledge, this is the first work capable of adapting to dynamic network conditions such as emergencies, and the first effort to apply NDN techniques to wireless sensor networks.

The rest of the chapter is organized as follows. Section 3.2 discusses the related work. Section 3.3 outlines the framework and assumptions made in the network model. Section 3.4 describes the operations and mechanisms of our protocol.

## 3.2 Related Work

Named Data Networking is a new network architecture proposed recently for the Internet [84]. In NDN, data are addressed by their names instead of hosting nodes' locations. The operation is based on two types of messages, *Interest* and *Data*, and the communication is initiated by the receiver. A receiver interested in certain data sends Interest messages carrying the name of the desired data. The Interest message propagates in the network following FIB (Forward Interest Base) states towards nodes hosting desired data. It also leaves a "trail" of PIT (Pending Interest Table) states in intermediate nodes. Once the Interest reaches a node hosting the desired data, Data messages can follow PIT states to traverse back to the receiver.

So far NDN research has largely focused on the Internet, with some efforts on mobile networking. Whether it can be used to satisfy the needs of wireless sensor networks is still unexplored. In this chapter, we use wireless recharging as a case study to investigate its applicability in wireless sensor networks, and we find that it does have attractive benefits in our scenario.

## 3.3 A Novel Framework for Wireless Rechargeable Sensor Networks

In this section, we describe the components, network model and assumptions for our NDN based wireless recharging framework. NDN has a few attractive benefits for our environment. First, by sending out new Interest packets, a mobile receiver can continuously update the routing states (i.e., PIT entries) in intermediate nodes. Data can follow the reverse paths traversed by the most recent Interest packets and reach the new location of the receiver. This solves the mobility issue of the SenCar and ensures that the latest energy information can reach the SenCar in a timely manner. Second, to scale to large network sizes, we divide the network in a hierarchical fashion and energy information is gathered in aggregated forms. Thus the data is bounded to an area rather than any particular node. This makes a natural fit for NDN: the data can be addressed using the area's name.

### 3.3.1 Network Components

The network consists of the following components for building and maintaining the name hierarchy, querying energy information, and recharging the sensor nodes.

*SenCar*: The SenCar queries the network for energy information and recharge nodes based on the energy information collected.

*Head nodes*: A head is a sensor node delegated to aggregate energy information from its subordinate area. When requested by the SenCar (for top level heads) or by the head of the upper level (for other heads), a head queries energy information from subordinate sub-areas at the lower level, aggregates such information and sends to the requester.

*Proxy*: A proxy node aggregates emergencies from sensor nodes and reports such information to the SenCar when queried. Only top level head nodes are proxies.

*Normal Node*: A sensor node not selected as a head is a normal node. It reports energy information to head nodes, or sends emergency directly to its proxy when its energy level drops below the emergency recharge threshold.

### 3.3.2 Name Assignments and Network Model

We assume sensors are scattered uniformly randomly. The network field is divided into several areas and each area is further divided recursively. The division of the area is based on geographical coordinates of the sensing field. Each division generates some new *sub-areas* and increases the number of *levels* in the network. This process repeats until the bottom level subarea becomes small enough such that the SenCar can recharge one such a subarea with no need to query normal energy levels in the middle of the recharging process. Fig. 3.1 gives an example of a 2-level network with 2 areas (in red solid line), each further split into 2 sub-areas (in blue dash line). Each sub-area on the second level contains about 10 sensor nodes.

Based on the results of area divisions, we assign NDN data names for different subareas in a hierarchical manner. For example, Fig. 3.1 shows all the name assignments for different subareas (e.g., the first level areas are “a” and “b”, and the second level has “a/a", “a/b", . . . ). Thus each subarea is identified by its unique hierarchical name. Each node has an ID including the name of the containing bottom level subarea plus an identifier. For example, “a/a/3” is node “3" in subarea “a/a”.

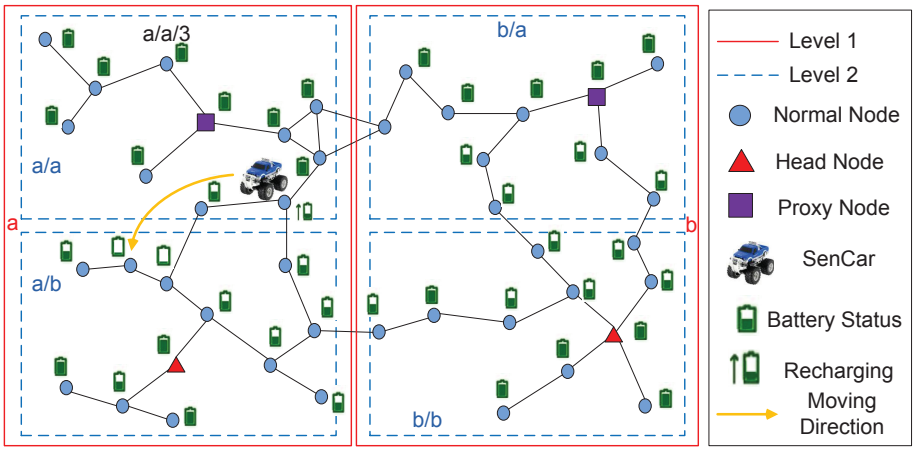



Figure 3.1: An illustration of area names and network components.

A message can carry the name of intended propagation subarea and nodes can use their names to ensure that the propagation does not go beyond that subarea.

In addition, we have the following assumptions: 1) Sensor nodes are stationary and each node knows its location. 2) Nodes have the same transmission range and messages are forwarded over multiple hops in large networks. 3) The SenCar has a positioning system and knows its own location. The IDs and locations of all sensor nodes, and the subarea names are known to the SenCar (e.g., through an one-time effort at the initial stage). 4) The field is barrier-free so the SenCar can move to any sensor node in straight movement lines. 5) Sensors might perform different tasks so the energy consumption is not uniform among nodes.

## 3.4 The NDN based Real Time Wireless Recharging Protocol

In this section, we present the detailed design of NDN based Real Time Wireless Recharging Protocol (NETWRAP). We first give an overview of NETWRAP in Section 3.4.1. Then we describe different operating phases of NETWRAP in Section 3.4.2.

### 3.4.1 Protocol Overview

In NETWRAP, the SenCar obtains the most up-to-date energy information from sensors and makes recharge decisions in real time. The energy information is aggregated on heads at different levels. To be robust, the head is usually selected as the node having the maximum energy level in its area. This selection process is done at the beginning of network startup through the propagation of *head selection* messages. The details will be discussed in the next subsection.

To start a new round of energy information collection, the SenCar sends out an *energy interest* message to poll the *heads* on the top level. Once the heads receive such messages, they send lower level *energy interest* messages to their child heads in respective subordinate areas. This process repeats down in the head node hierarchy, until finally the bottom level *energy interest* messages reach the nodes in the bottom level subareas.

Once a sensor node receives a bottom level *energy interest*, it responds by sending out an *energy* message containing its ID and battery level. When the heads on the bottom level receive such *energy* messages, they select sensor nodes below a normal recharge threshold, and send the names of these nodes and their energy information in an *energy* message to their parent head nodes. This is repeated up the head node hierarchy, until finally the top level head nodes have the aggregated energy information and send it to the SenCar.

Note that in NETWRAP, for the purpose of reducing transmission overhead, the head is delegated partial responsibilities to pre-select sensor nodes to be recharged. This is done as the heads select the nodes with low energy level. In upper levels, a head selects the subordinate area which can be recharged with the most amount of energy. Thus the SenCar can replenish the network with more energy in one movement.

Such normal energy aggregation is conducted at the request of the SenCar. For emergency nodes that have dangerously low battery levels below an emergency threshold, they send out *emergency* messages to the proxy that manages its area. The route to its proxy is built by *head selection* messages from the proxy.

The SenCar monitors whether there is any emergency by sending out *emergency interest* messages after finishing recharging any single node. These messages are directed to proxies where lists of emergency node are stored. A proxy responds

by sending back the emergent node names, estimated residual lifetimes and energy levels. The SenCar receives the message and follows the emergency recharge algorithm to recharge those nodes.

When a head node is low on energy, it can choose another node with high energy, and send out a *head notification* message to notify the latter to become the new head.

### 3.4.2 Protocol Design

We describe the detailed protocol assuming the head hierarchy has $l$ levels.

#### 3.4.2.1 Head Selection

After the areas and names have been configured, the network performs head selection from the bottom up starting at the $l$-th level. Since initially sensor nodes have about the same level of energy, any of them may become a head. Each sensor node $i$ generates a random probability $x$. If $x > K$, where $K$ is a pre-determined threshold, the node floods a *head selection* message in its $l$-th level subarea, containing the name of this subarea, $x_{max} = x$, and $ID_{max}$ set to its own ID. Otherwise the node waits for messages from other nodes in the area.

A node receiving such a *head selection* message compares the $x_{max}$ in the message with its local record $x_{max}$. If its local record is larger, the message is discarded. Otherwise, the sensor updates its local $x_{max}$ to that in the message, sets $ID_{max}$ to that in the message, and forwards the message to its neighbors except the one that sent it this message. Finally the node with the maximum $x$ wins the election and is recorded by all the nodes in this subarea as the head.

New heads at the $l$-th level then contend to become heads of the $(l-1)$-th level. They flood new *head selection* messages in the $(l-1)$-th level subarea, carrying the area's name, their respective $x$ values and IDs. Intermediate nodes perform similar comparisons. This will elect the heads at the $(l-1)$-th level. This process is repeated until head nodes of all levels are elected.

One difference for the *head selection* messages starting from the $(l-1)$-th level and up is that messages carrying smaller $x$ than the local copy are not discarded. Instead, they are propagated throughout the respective subarea. This builds routing states in intermediate nodes of the subarea: An intermediate node has one entry for

each child head, pointing to the neighbor from which the message from that head arrives first. Duplicate copies of the same message arriving later are discarded.

Such states are effectively FIB (Forward Interest Base) entries in NDN. Later a parent head at the $(k-1)$th-level can send *energy interest* messages to its child heads at the $k$-th level using such states. To build FIB entries for the 1st level head nodes, they each flood a top level *head selection* message throughout the whole network. Later the energy interest queries from the SenCar can use such states to reach them.

#### 3.4.2.2 Normal Energy Interest Propagation

After the head hierarchy is constructed, the SenCar sends *energy interest* messages to query for nodes needing recharge. The energy information is gathered on demand, and top down in the hierarchy. We will describe normal energy information collection first. Emergency information is collected similarly, but with only top level heads involved to reduce latency.

For normal energy information, an interest message is sent by the SenCar (e.g., with data name set to "/energy/normal/*", the energy information at all top level heads). Intermediate nodes use the FIB entries established by top level *head selection* messages to forward it to all top level head nodes. To guide the return of future data, an intermediate node also sets up a PIT (Pending Interest Table) entry pointing to the neighbor from which the interest message is received. Later energy information from a head can follow such directions to return to the SenCar.

Upon receiving an energy interest message, a first level head sends a new energy interest message to its child heads, with the data name set to all subareas of its children (e.g., from the head node of area /a, "/energy/normal/a/*"). Similarly, these messages reach all child heads following FIB entries. Intermediate nodes also set up PIT entries so later energy information from child heads can go back to their parent head. This process is repeated down the hierarchy, until finally heads at bottom level flood their respective subareas with interest messages.

#### 3.4.2.3 Normal Energy Report and Node Recharge

When a sensor node receives a $l$-th level *energy interest* message, it responds with an *energy* message including its ID and residual energy. With the help of PIT en-

tries, the message is returned to the head of the $l$-th level.

The head examines if the reported residual energy is less than the normal recharge threshold. If so, the ID of the node is added to a list, and the energy that can be recharged to this node is added to a summation counter. After the head has collected these messages, it sends an aggregation message, containing the list, the summation counter and its subarea name to its parent head. A parent head compares such messages from its child heads, selects the one with the largest summation counter (i.e., the bottom level subarea that can be recharged of the greatest amount of energy), and forwards to its parent head. This process is repeated upwards in the hierarchy. Finally, the SenCar receives one message from each top level head. The SenCar then moves to the bottom level subarea with the largest summation counter, and recharge those nodes in the ID list one by one. Only after recharging those nodes will the SenCar send another normal energy query.

The reason we delegate selection partially to head nodes is twofold. First, we expect much less variation in normal energy levels. Thus the SenCar can choose one bottom level subarea and finish recharging all listed nodes. Only after the whole subarea is recharged, we expect enough changes in normal energy distribution that warrants a new normal energy query from the SenCar. Thus it is not necessary to include the energy information from other sensors because that will be collected again. Second, this also keeps the return message sizes small and reduces overhead.

#### 3.4.2.4 Emergency Energy Report and Node Recharge

Emergency energy report is slightly different due to the urgency. Each node periodically examines its energy level. If the level is below the emergency recharge threshold, it immediately sends an *emergency* message containing its ID and energy level to its proxy (i.e., its top level head node). Because the head node floods a top level *head selection* message during head election, the same FIB entries can be used to forward emergency messages to the head.

Instead of waiting for recharging a whole bottom level subarea, the SenCar sends out an *emergency interest* message to each proxy (i.e., the top level head nodes) after finishing recharging any single normal or emergency node. The proxies return their lists of emergency node IDs and energy levels, if there exists any. The SenCar interrupts its normal recharge, switches to emergency operation mode and

recharge those nodes first. It switches back to normal operation mode when no more emergency is reported.

#### 3.4.2.5 Head Hierarchy Maintenance

A head can be short on energy, which can happen once in a while because the head usually engages in more activities than a normal node. When this happens, a new head is needed. Because only heads of bottom levels contend for higher level elections, a head at any level is always the head of its bottom level subarea. It receives the energy reports from normal nodes in its bottom level subarea upon the normal interest query from the SenCar. So it can choose a node with the highest energy, and floods a *head notification* message to notify all nodes in the bottom level subarea of the new head.

The new head then triggers a new head election process in its $(l-1)$-th level subarea. It propagates a new *head selection* message in its $(l-1)$th subarea, but carrying its energy level instead of the random number $x$. Other heads in this $(l-1)$-th level subarea do the same. Then a new $(l-l)$-th head with the maximum energy is elected. If this is the same head, the process stops. Otherwise, the new $(l-1)$-th level head triggers the same process in its upper level subarea, until finally a new top level head is elected.

### 3.4.3 Summary of Protocol Design

We now summarize how we use NDN to route different messages briefly. First, FIB entries are established during the head selection process so that the *interest* message can be sent from parent head nodes to child ones. Second, the propagation of *interest* messages from the SenCar, or from parent to child heads, establishes PIT entries for later return of *energy* messages from top level heads or child heads. Third, FIB entries to top level heads (i.e., proxies) allow emergency nodes to send reports to proxies without waiting for the emergency interest queries from the SenCar, which minimizes latency.

## 3.5 Conclusions

In this chapter, we propose a novel framework for real time wireless energy recharging for wireless sensor networks. We develop a comprehensive set of protocols using NDN concepts and mechanisms to enable effective recharging for the perpetual operation of the network. The protocols adapt to unpredictable network conditions and satisfy the needs for both normal and emergency recharging. We formally analyze the probability for the energy neutral condition required by perpetual operations. We also model the optimal recharging of multiple emergencies as an Orienteering problem and provide a Knapsack approximation that has high accuracy under typical network environments. The extensive simulation results demonstrate the efficiency and effectiveness of our framework, and the close match of energy neutral analysis with simulation results.

# Chapter 4

# Optimal Energy Replenishment in WSNs with Random Topology

With the realtime energy information collected from the networks, SenCar is able to perform energy replenishment to achieve *perpetual operation* if nodes can always be recharged before energy depletion. When to recharge which nodes, and in what order, critically impact the outcome. So far only a few work has studied this problem and relatively static recharging policies were proposed. However, dynamic changes such as unpredictable energy consumption variations in nodes, and practical issues like scalable and efficient gathering of energy information, are not yet addressed.

In this chapter, we derive analytic results on *energy neutral* conditions that give rise to perpetual operation. We discover that optimal recharging of multiple emergencies using one SenCar is an Orienteering problem with Knapsack approximation. We further address the scalability of the networks by using multiple SenCars. We define the problem of scheduling and coordinating the SenCars to recharge the sensors within their residual lifetimes while minimizing the cost of SenCars as the *Emergency Recharge Optimization with Multiple SenCars* (EROMS) problem. Our extensive simulations demonstrate the effectiveness and efficiency of the proposed framework and validate the theoretical analysis.

## 4.1 Introduction

In wireless rechargeable sensor networks, one or more SenCars move around and recharge sensors conveniently. The recharging policy - when to recharge which nodes and in what order - critically impacts the efficiency and thus the lifetime of the network. So far only a few work [82, 83] has studied the recharging policy problem. Basically, nodes report their energy levels periodically, and a centralized algorithm computes a specific order to recharge all nodes in the next cycle. Although commendable first steps, they do not fully consider important practical issues, which significantly limit their applicability in a real environment.

First, it takes nontrivial (e.g., 30-60 min) time to recharge a commercial off-the-shelf battery, such that finishing one round of recharging for a network of a few hundred nodes may take several days. During this time the energy levels of nodes may have changed significantly due to unpredictable external events that can trigger extensive activities and quickly drain the battery. The recharging policy computed at the beginning of the cycle is no longer optimal. This can cause energy depletion on some nodes, leading to network disconnection or application failures. Second, the timely, efficient and scalable gathering of energy information of nodes to a mobile vehicle is an important and challenging issue in itself. The previous work does not consider this issue and assumes such information is already available. Finally, they use centralized algorithms that have high complexity and may not scale to large network sizes. A distributed solution is more desirable in real network environments.

In this chapter, we propose a novel real time recharging framework that optimizes the recharging policy under dynamic network conditions. The SenCars collect real-time energy information using the NETWRAP protocol we introduce in Chapter 3 and makes recharging decisions based on the latest energy information. To deal with unpredictable emergencies where nodes may dramatically drain the battery in short time, the recharging of sensor nodes whose energy levels are below a critical threshold has high priority and takes precedence over those that can work for relatively long time with their residual energy.

We study the conditions for perpetual operations of the network under such a recharging framework. We identify the energy neutral requirement (i.e., the energy replenished should be more than or equal to the energy consumed) and formally

derive the probability for this condition to hold in a network. Given the network and SenCar parameters, an administrator can estimate the likelihood that the network can operate perpetually.

Due to the unpredictable nature of external events, the SenCars may need to deal with multiple concurrent *emergencies* occurring in different locations. We study the optimal recharging policy using one SenCar and find that it can be formulated as an *Orienteering problem* [75], which has been studied before but only with computationally intensive solutions. We show that by taking reasonable approximations, it can be converted into a Knapsack problem, which has more efficient heuristics. We extend this to multiple SenCars scenario, and find that schedule and coordinate multiple SenCars to recharge these sensors within their residual lifetimes while minimizing the cost of SenCars is defined as the Emergency Recharge Optimization with Multiple SenCars (EROMS) problem.

We make the following contributions in this chapter. We formally define the energy neutral condition and analyze the theoretical probability for perpetual operations. We discover that optimal recharging of multiple emergencies using one SenCar is an Orienteering problem, and further approximate it to a Knapsack problem with more efficient heuristics. We also discover that the emergency recharge optimization with multiple SenCars is a m-TSP with Deadlines problem, and further propose an efficient heuristic algorithm suitable to sensor recharging context. Finally, we conduct extensive simulations to demonstrate the effectiveness and efficiency of the framework and validate the correctness of the theoretical analysis.

The rest of the chapter is organized as follows. Section 4.2 discusses the related work. Section 4.3 addresses theoretical analysis and simulation of the problem of optimal recharge using one SenCar, including the definition of energy neutral problem, system modeling and performance evaluation. Section 4.4 describes the problem of optimal recharge with multiple SenCars. Finally, Section 4.5 concludes the chapter.

## 4.2 Related Work

Wireless rechargeable sensor networks have drawn interest from both academia and industry recently [80–83]. In [80], the impact of wireless charging technology on wireless sensor networks was investigated, and heuristic algorithms were developed

to solve the deployment and routing problem. In [81], the problems of point provisioning and path provisioning were studied in a wireless rechargeable sensor network built from industrial wireless sensing platform and commercial off-the-shelf RFID readers.

In [83], the scenario of periodically recharging each sensor node using a mobile charging vehicle was considered. A near-optimal solution was provided to calculate the optimal traveling path of the mobile car. It forms the shortest Hamilton cycle through all sensor nodes and maximizes the ratio of mobile car's charging to idle time. In [82], the problem of joint optimization of effective energy charging and high-performance data collections with bounded tour length and data gathering interval was studied. A two-step approach was proposed to recharge the nodes with the least residual energy and maximize network utility.

The above work makes pioneering steps in this new area of wireless rechargeable sensor networks. However, none of them consider how the energy information is aggregated and reported to the mobile car, nor do they handle changes in energy levels that occur inevitably and unpredictably during long recharging cycles.

## 4.3 Wireless Energy Replenishment Using a Single SenCar

We first study the scenario where a single SenCar is deployed for energy replenishment. We prove that a single SenCar is able to attend a network with a moderate number of sensors, which is applicable for many places. Although the schematic designed for multiple SenCars can also be applied to single SenCar deployment, the complexity of single SenCar charging scheduling is much lower, which will be demonstrated in the next section.

### 4.3.1 Theoretical Analysis

We investigate a couple important theoretical questions in this section. First, what is the condition for the network to operate perpetually? Second, when multiple emergencies occur, what is the optimal recharging policy such that no sensor node would exhaust energy and the total energy replenished into the network is maximized?

#### 4.3.1.1 Energy Neutral Condition

A rechargeable sensor network achieves perpetual operation when the *energy neutral* condition holds, i.e., for each sensor node the energy provided is no less than the energy consumed in any arbitrarily large time period.

We assume that the energy consumption in each time slot on a sensor is a random variable. A long time period $T_n$ is equally slotted into $n$ slots. Let $R_{T_n}$ and $E_{T_n}$ denote the energy replenished and consumed for a sensor node during time period $T_n$, respectively, and $E_0$ denote the node's initial energy. Thus the energy neutral condition is:

$$R_{T_n} + E_0 \geq E_{T_n} \tag{4.1}$$

We first estimate a loose upper bound for $R_{T_n}$. Intuitively, a SenCar reaches its maximum recharging capacity when it can "barely" keep up with the recharging needs. This is when it keeps recharging node after node without any idle time in between, and each node has almost zero energy before being recharged.

The SenCar can replenish at most the battery's full capacity in the full recharge time [1]. Thus the average recharge power $R$ (i.e., the rate energy is replenished) for the whole network is the full battery capacity divided by the full recharge time (e.g., 780 mAh divided by 80 min for a Panasonic Ni-MH AAA battery [86]). The average recharge power $r$ for each sensor node is $R$ further divided by $N$, the number of nodes. Thus, the upper bound of $R_{T_n} = rT_n$.

$E_{T_n}$ is a random variable and its probability distribution can be estimated using the central limit theorem. We have the following lemma.

**Lemma 1.** *The distribution of the energy consumption $E_{T_n}$ of a sensor node in a long time period $T_n$ asymptotically converges to a Normal Distribution.*

*Proof.* We model the energy consumption of a sensor node in a unit time slot as an independent and identically distributed random variables $X_1$ with mean $\mu(1)$ and standard deviation $\sigma(1)$. Using the Central Limit Theorem, the summation of many such i.i.d. random variables follow a Normal distribution of $\mu(n) = n\mu(1)$ and $\sigma^2(n) = n\sigma^2(1)$. Similar results have been derived in [87]. □

[1] We assume fully recharging batteries to avoid "memory effects" that can reduce the number of chargeable cycles.

As a result, when the time period $T_n$ is long enough, $E_{T_n} \sim \mathcal{N}(\mu(n), \sigma^2(n))$. Therefore, the energy neutral condition on a sensor node holds with the following probability:

$$p = Pr\{R_{T_n} + E_0 > E_{T_n}\} = \Phi\left(\frac{R_{T_n} + E_0 - \mu(n)}{\sqrt{\sigma^2(n)}}\right) \tag{4.2}$$

The probability to have $k$ out of $N$ sensor nodes working in a network follows a Binomial distribution (i.e., $k \sim \text{B}(N, p)$).

To illustrate how Eq. (4.2) can be used to estimate network longevity, we consider a concrete example comprised of 800 sensor nodes with the recharge function of Panasonic Ni-MH AAA batteries. For a time period $T_n$ of 6 months and each time slot is 1 second, the average total energy recharged $R_{T_n}$ for a sensor node is 76.18 KJ. Assume in an application the battery can last 32 days on average without recharge and the standard deviation is 1.5 times of the mean, which yields $\mu(1) = 5.9mJ$ and $\sigma^2(1) = 77mJ^2$. Using Eq. (4.2), we can calculate $p = 0.71$. The mean value of binomial distribution is $Np_i$ so on average we expect at most 568 nodes can work perpetually. Later in Section 4.3.3, we will see that this estimation is quite accurate, only about 10% higher than simulation results.

### 4.3.2 Emergency Recharge Optimization

We study the optimal recharge policy for multiple emergencies in this subsection. We consider the scenario where $m$ emergency nodes need to be recharged and define $T_{em}$ as the mean inter-arrival time of emergencies in a long run. The SenCar needs to recharge these nodes before they exhaust energy, and it should choose a recharging order to maximize the amount of energy replenished into the network before a new emergency occurs (i.e., within time $T_{em}$).

This problem can be formulated into the famous Orienteering Problem (OP). In an OP, a set of control points associated with scores are visited by competitors before a time expiration. The competitor collecting the highest score wins the game. In analogy to the OP, the SenCar visits sensor nodes in a directed graph $G = (V, E)$, where $v_1$ is the starting location of the SenCar and $v_i$, $i \in \{2, 3, \ldots, m\}$ represents the emergent sensor locations to be visited. $E$ is the set of routes among sensor nodes or the starting location of the SenCar. The profit $r_i$ at $v_i$ is the amount of

energy replenished when recharging the node at this location to full capacity. The cost of traveling along $e_{ij}$ is the traveling time from $v_i$ to $v_j$ (denoted as $t_{ij}$) plus the recharging time at $v_i$ (denoted as $t_i$). To be consistent with the modeling of OP, we make the SenCar return to $v_1$ virtually after recharging all the selected nodes by setting $t_{i1} = 0, i \in \{2, 3, \ldots, m\}$, $t_1 = 0, r_1 = 0$. The objective is to maximize the total amount of energy refilled given the time constraint $T_{em}$. We introduce decision variables $x_{ij}$ for edge $e_{ij}$. The decision variable is 1 if an edge is visited, otherwise it is 0. $u_i$ is the position of vertex $i$ in the path. The formulation of the problem is shown below.

$$\mathbf{P1}: \quad \max \sum_{i=2}^{m-1} \sum_{j=2}^{m} r_i x_{ij}, \tag{4.3}$$

**Subject to**

$$\sum_{j=2}^{m} x_{1j} = \sum_{i=1}^{m-1} x_{i1} = 1, \tag{4.4}$$

$$\sum_{i=1}^{m-1} x_{ik} = \sum_{j=2}^{m} x_{kj} \leq 1; \forall k = 2, 3, \ldots, m-1 \tag{4.5}$$

$$\sum_{i=1}^{m-1} \sum_{j=2}^{m} (t_{ij} + t_i) x_{ij} \leq T_{em}, \tag{4.6}$$

$$x_{ij} \in \{0, 1\}; \forall i, j = 1, 2, \ldots, m, \tag{4.7}$$

$$2 \leq u_i \leq m; \forall i = 2, 3, \ldots, m, \tag{4.8}$$

$$u_i - u_j + 1 \leq m(1 - x_{ij}); \forall i, j = 2, 3, \ldots, m \tag{4.9}$$

Constraint (4.4) guarantees that the recharge path starts at 1 and finishes at 1. Constraint (4.5) ensures the connectivity of the path and that every vertex is visited at most once. Constraint (4.6) sets the time threshold to be $T_{em}$. Constraint (4.7) imposes $x_{ij}$ to be 0-1 valued. Constraints (4.8) and (4.9) eliminate the subtour in the planned route. The subtour elimination constraints are formulated according to [90].

There have been quite a few heuristics to solve the OP [76–79] and a recent survey is available in [75]. Tsiligirides [76] has developed a stochastic Monte Carlo technique and a divide-and-conquer method. Chao, et. al [78] proposed a five-step heuristic. However, they are all quite complex and a more efficient solution is

desirable to find a solution in short time to avoid nodes exhausting energy. We have the following lemma.

**Lemma 2.** *When recharging sensor node $i$ requires much more time than traveling from node $i$ to node $j$ ($i.e.$ $t_j \gg t_{ij}$), the OP can be approximated as a Knapsack problem.*

*Proof.* Once the traveling time $t_{ij}$ is negligible, Constraint (4.6) in the original OP formulation can be rewritten as $\sum_{i=2}^{m} t_i y_i \leq T_{em}$ where $t_i$ is associated with the item weight in a Knapsack problem. The item value is the amount of energy replenished $r_i$. $y_i$ is a 0-1 valued decision variable and is set to 1 only if $v_i$ is selected for recharge. □

Thus we have a much simpler Knapsack formulation:

$$\mathbf{P2}: \quad \max \sum_{i=2}^{m} r_i y_i, \tag{4.10}$$

**Subject to**

$$\sum_{i=2}^{m} t_i y_i \leq T_{em}. \tag{4.11}$$

$$y_1 = 1, \tag{4.12}$$

The complexity of Knapsack heuristic for $m$ emergencies takes $\mathcal{O}(mT_{em})$ running time. It is much simpler and efficient to implement on the SenCar than the OP heuristics in [76–79].

To examine the accuracy of Knapsack approximation, we test several cases when the number of emergencies $m$ varies from 3 to 12, and compare the amount of energy recharged by Knapsack approximation against the exact solution. We assume a $200 \times 200m^2$ square field, the SenCar moving at a constant speed of 3 m/s, and the recharge function follows that of Panasonic Ni-MH AAA battery. The accuracy is defined as $1 - \left|\frac{R_k - R_e}{R_e}\right|$, where $R_k$ is the solution by Knapsack approximation and $R_e$ is the solution by exhaustive search. Table 4.1 shows that the accuracy is more than $99\%$ for different parameter settings.

Table 4.1: Accuracy of Knapsack approximations to exact solutions.

| # Emergencies $m$ | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|
| $T_{em} = 300$ min | 100% | 100% | 100% | 100% | 100% |
| $T_{em} = 400$ min | 100% | 100% | 100% | 100% | 100% |
| $T_{em} = 500$ min | 100% | 100% | 100% | 100% | 100% |
| # Emergencies $m$ | 8 | 9 | 10 | 11 | 12 |
| $T_{em} = 300$ min | 100% | 100% | 100% | 100% | 100% |
| $T_{em} = 400$ min | 99.7% | 99.6% | 99.9% | 99.8% | 99.7% |
| $T_{em} = 500$ min | 100% | 100% | 99.6% | 100% | 100% |

### 4.3.3 Performance Evaluations

In this section, we use simulation to evaluate the effectiveness and efficiency of our framework. We have developed a discrete event-driven simulator using POSIX Thread programming in C language. We use two network sizes of 500 and 800 sensor nodes, uniformly randomly distributed over a $200 \times 200m^2$ and $250 \times 250m^2$ square field respectively. The network has a 3-level hierarchy with $4^l$ number of areas on the $l$-th level. The energy consumption on each sensor is a Gaussian random variable for each time slot. The relationship between recharged energy and recharge duration follows that of Panasonic Ni-MH AAA battery.

To undestand the impact of different energy consumption rates and fluctuations, we adopt two node energy consumption rates $r_c^1 \sim (4.2mW, 11mW^2)$ and $r_c^2 \sim (5.9mW, 77mW^2)$. On average, a fully recharged battery can last for 45 and 32 days at $r_c^1$ and $r_c^2$ respectively. The standard deviation is chosen to be 0.8 and 1.5 times the corresponding mean rate to emulate small and large fluctuations in energy variation. All the parameter settings in the simulation are listed in Table 4.2.

#### 4.3.3.1 Protocol Performance Evaluations

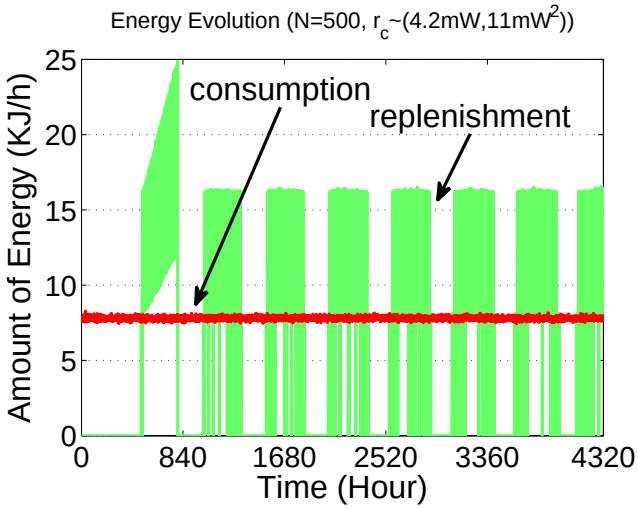
**4.3.3.1.1 Energy Evolution and Energy Distribution** First, we show the energy evolution in the network of 500 nodes at different consumption rates. In Fig. 4.1, the amount of energy consumed and replenished in every one-hour time slots is plotted as functions of simulation time. Fig. 4.1(a) shows that the recharged energy remains at zero until after about 500 hours, then it increases steadily and stops. This is because that none of the nodes need recharge until the normal recharge threshold (50% of capacity) is reached at about 500 hours.

Table 4.2: Parameter settings for performance evaluation of optimal recharge using one SenCar.

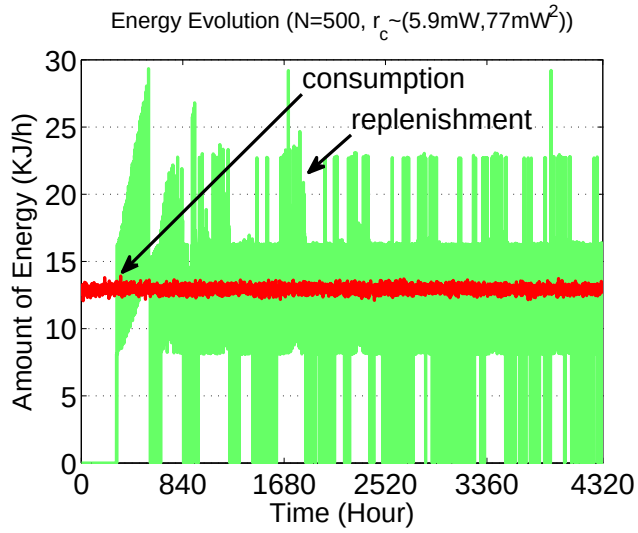
| Parameter | Value |
|---|---|
| Field Length | 200m, 250m |
| Number of Nodes $N$ | 500, 800 |
| Number of Levels | 3 |
| Areas on $l$-th level | $4^l$ |
| Battery Capacity | 780 mAh |
| Transmission Rang | 16 m |
| Consumption Rate $r_c$ | $r_c^1 \sim (4.2mW, 11mW^2)$, $r_c^2 \sim (5.9mW, 77mW^2)$ |
| SenCar Speed | 1 m/s |
| Recharge Time to Full Capacity | 73.4 mins |
| Normal Recharge Threshold | 50% |
| Emergency Recharge Threshold | 10% |
| Simulation Time | 6 months |

As sensor nodes reach that threshold, the SenCar starts normal recharge. A node recharged at a later time has consumed more energy, thereby the SenCar puts back more energy to it. This corresponds to the increase of recharged energy from about 500 to 840 hours. The recharge process pauses when the SenCar has responded to every request and resumes when a new request is received. After the network enters equilibrium, the SenCar operates in a pattern of alternating between idle and recharge as shown in Fig. 4.1(a). Similar behavior of the SenCar is observed in Fig. 4.1(b) except that the idle time is shorter because of more frequent recharge due to a higher energy consumption rate.

The energy distributions among nodes carries valuable information about the health of the network. A distribution at a higher average energy is more robust to unexpected surges in energy consumption. Fig. 4.2 shows the energy distribution of a 500-node network with $r_c^1$ and $r_c^2$ after the network enters energy equilibrium. In Fig. 4.2(a), all the nodes maintain more than 50% of the energy indicating that the SenCar has enough capacity to work only in normal operation mode and recharge all nodes. However, in Fig. 4.2(b), some nodes have energy under the normal recharge threshold, indicating the possibility of emergencies and the SenCar may need to switch to emergency recharge mode occasionally.

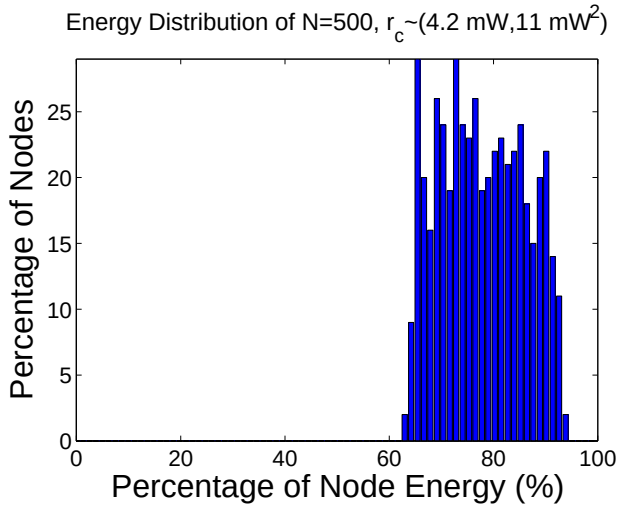



(a)

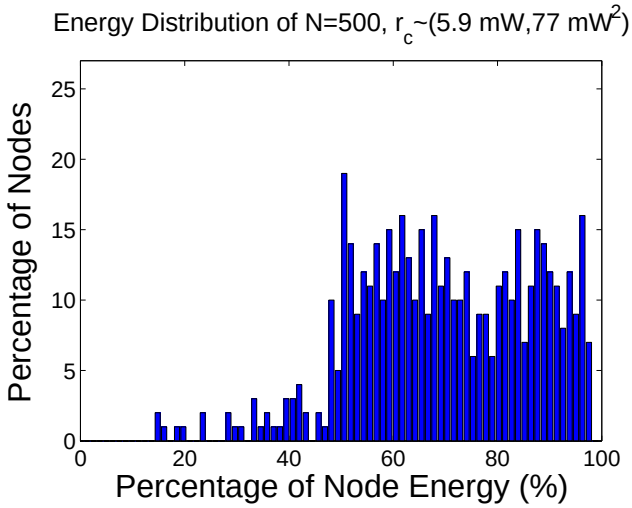



(b)

Figure 4.1: Evolution of energy consumption vs. energy replenishment in 6 months time. (a) Network Size $N$ = 500 and consumption rate $r_c \sim (4.2mW, 11mW^2)$; (b) Network Size $N$ = 500 and consumption rate $r_c \sim (5.9mW, 77mW^2)$.




(a)




(b)

Figure 4.2: Energy distribution at equilibrium. (a) Network Size $N$ = 500 and consumption rate $r_c \sim (4.2mW, 11mW^2)$; (b) Network Size $N$ = 500 and consumption rate $r_c \sim (5.9mW, 77mW^2)$.

**4.3.3.1.2 Number of Emergencies** Fig. 4.3 and Fig. 4.4 compare the percentage of nodes in emergency and nonfunctional (i.e., energy at zero) situations for networks of 500 and 800 nodes with different energy consumption rates. The 500 node network at consumption rate $r_c^1$ has no emergency since all the nodes are recharged timely after their energy drops below the normal recharge threshold. As the network size or energy consumption rate increases, the energy of some nodes falls below the emergency recharge threshold. When the increase goes beyond the capacity of the SenCar, it cannot recharge all of them in time and some nodes may become temporally nonfunctional.

For the 500 nodes with $r_c^2$ and 800 nodes with $r_c^1$, there are a number of nodes in emergency and they become temporally nonfunctional at the beginning. After the networks enter equilibrium, most of the nodes are recharged in time. Only a few nodes enter emergency state or become temporally nonfunctional, which happens only sporadically. When the 800 node network has consumption rate at $r_c^2$, the recharging capacity of the SenCar is exceeded. Fig. 4.4 shows that the percentage of nonfunctional nodes holds persistently around 38%.

Now we examine how much gap exists for the maximum number of nodes the SenCar can sustain between the simulation results and the theoretical analysis. We can calculate that the 800 node network at consumption rate of $r_c^2$ has $p = 0.71$ probability for the energy neutral condition (Eq. (4.2)) to hold. Thus the SenCar can sustain at most 568 (i.e., $Np$) nodes. This is about 9% higher than the actual number of 496 nodes indicated by simulations.

The gap is caused mainly by two reasons. 1) The SenCar does not start recharging until there are some nodes whose energy drops below the normal recharge threshold. This idle time corresponds to the time from 0 to about 500 in Fig 4.3, during which no energy is refilled into the network. 2) The SenCar does not recharge nodes when it is moving or collecting energy information. Table 4.3 shows that the theoretical probability matches very well against the percentage of nonfunctional nodes from simulation results. We believe that although the energy neutral analysis gives a loose upper bound, the gap is reasonably small so network administrators can make reasonable estimations.

**4.3.3.1.3 Response Time to Emergencies** [Cong] We also evaluate the response time to emergencies and compare with the static optimization approaches used in

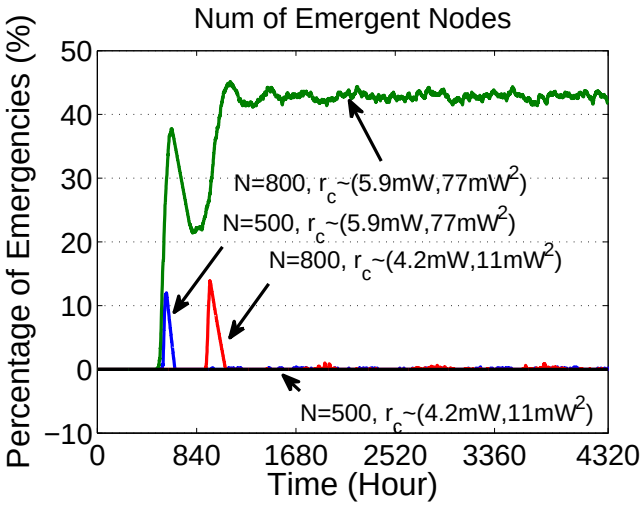



Figure 4.3: Number of emergent nodes.

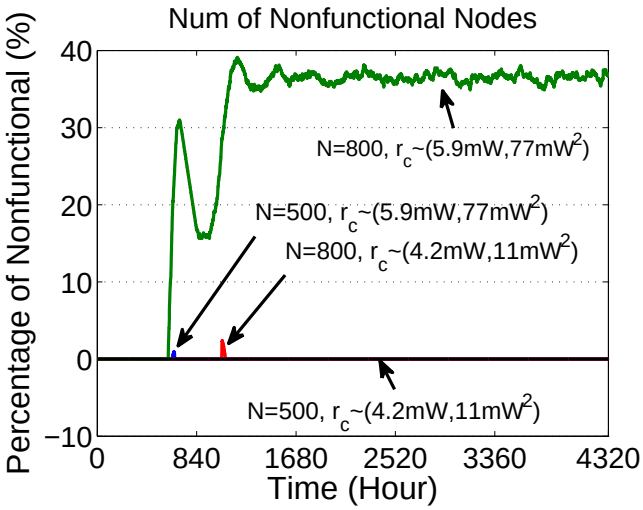



Figure 4.4: Number of nonfunctional nodes.

Table 4.3: Probability for the energy neutral condition to hold.

| $N$ and $r_c$ | $P_{op}$ | %(nonfunctional) |
|---|---|---|
| $N = 500, r_c^1$ | 1.0 | 0 |
| $N = 500, r_c^2$ | 1.0 | 0 |
| $N = 800, r_c^1$ | 0.999 | 0 |
| $N = 800, r_c^2$ | 0.71 | 38 |

[82, 83]. The response time to emergencies is measured from the time a node reports emergency until it is recharged by the SenCar. A shorter response time indicates that the SenCar can respond faster to emergencies. Fig 4.5 shows the SenCar's average response time to emergencies compared to the static approach when $N = 500, 800$ and energy consumption rate at $r_c^2$. In the static approach, the SenCar selects nodes with energy less than the normal recharge threshold, calculates the minimum traveling distance throughout these nodes and performs recharge one by one. We can see that when $N = 500$, the average response time in our approach is less than 3 hours whereas it requires as long as 20 hours in the static approach.

The situation becomes worse with the static approach when $N = 800$ because of the surging number of emergencies which ultimately results in as high as 56% nonfunctional nodes. The average response time increases to around 200 hours because in [82, 83] emergency nodes and normal nodes are not differentiated. A pre-computed route contains the combination of these nodes would result in extremely long waiting time for emergency nodes to get recharged. The approach becomes infeasible as the network size increases. A node in emergency would have been dead already when the SenCar arrives. In contrast, our real-time approach prioritizes nodes in emergency. The average response time is 14 hours even when $N = 800$ which is more than one order of magnitude faster than that of static approaches. It also incurs only 38% nonfunctional nodes (in Fig. 4.4), much less than the 56% in static approaches. Thus for extreme cases when the capacity of the SenCar is exceeded, nonfunctional situations are resolved much faster than the static approach. [Fan: The new text is clearer] [Cong]

#### 4.3.3.2 Protocol Overhead and Cost Evaluations

**4.3.3.2.1 Evaluation of Protocol Overhead** We evaluate the overhead introduced by our protocol, including all types of messages sent by sensor nodes or the SenCar to recharge nodes. Fig. 4.6 shows the average overhead per node in a 6 month period. The average overhead on each sensor node is from 4 to 7 bits per second, which is negligible compared to radio transmission rates in sensor nodes (e.g., 20 - 900 kbps). We also find that the overhead is slightly higher at the lower energy consumption rate, and much lower for the larger network size of 800 nodes. This can be explained by the difference between normal and emergency

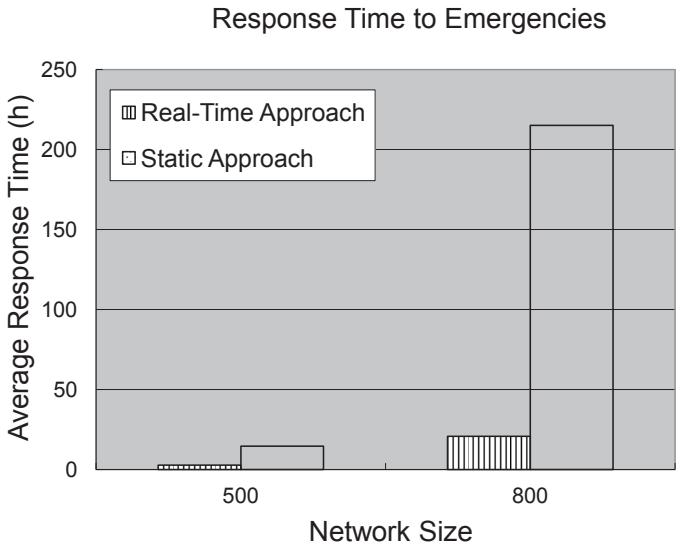



Figure 4.5: Average response time to emergencies.

energy information gathering. In the 500-node network, the SenCar has to collect energy information and performs normal recharge operation more frequently. More messages are produced during normal recharge operation in collecting energy information along the entire head hierarchy. However, in the 800-node network, there are more emergency situations and the emergency messages are directly reported to the proxies without propagation along the head hierarchy. Thus, the overhead in the 800-node network is less than the 500-node network.

#### 4.3.3.3 Evaluation of SenCar Maintenance Cost

We use the the mileages the SenCar travels to evaluate the cost (e.g., the energy consumed) for the SenCar to move around. Fig. 4.7 shows the accumulated mileages in 6 months. For the network with 500 nodes and consumption rates at $r_c^1$, there is no emergency, and it takes long time for the energy of a node to drop below the normal recharge threshold, thus the SenCar seldom moves and has the least mileages. When the energy consumption rate increases to $r_c^2$ for the 500-node network, the energy of nodes drops much faster. As a result, the SenCar performs normal recharge more frequently and the mileages increase significantly.

For the network with 800 nodes and the energy consumption rate set to $r_c^1$, emergencies occur more frequently than the network with 500 nodes at $r_c^1$, thus the

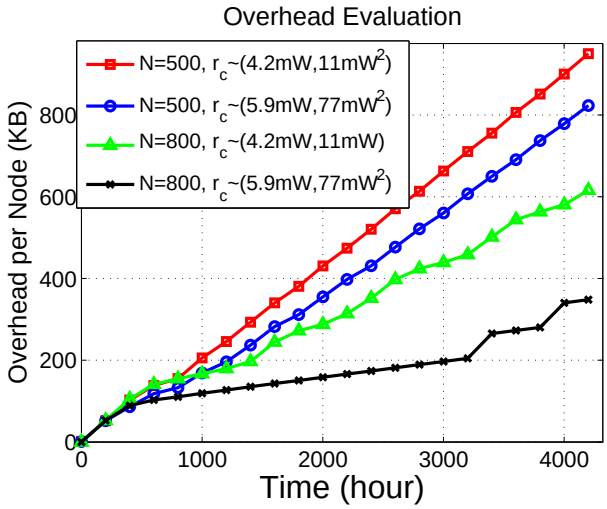



Figure 4.6: Average overhead for each sensor node per second.

SenCar performs emergency recharge more frequently. Since it takes longer time to recharge a node in emergency, the SenCar moves less frequently and the mileages are less than 500 nodes with consumption rate at $r_c^2$. When the energy consumption rate increases to $r_c^2$, emergencies exist persistently and the SenCar performs emergency recharge all the time, which is presented in Fig.4.7 as the mileages of 500-node network at $r_c^2$ exceed that of 800-node network at $r_c^2$ after 2520 hours, and the trend that the mileages of 800-node network at $r_c^1$ will also exceed that of 800-node network at $r_c^2$ shortly after 4320 hours.

### 4.3.4 Summary

The simulation results demonstrate that NETWRAP is very effective in achieving perpetual operation with high efficiency in wireless sensor networks. The SenCar has the capacity to maintain the network energy level and handle emergencies in large networks as no nodes deplete energy throughout the 6 month-simulation.

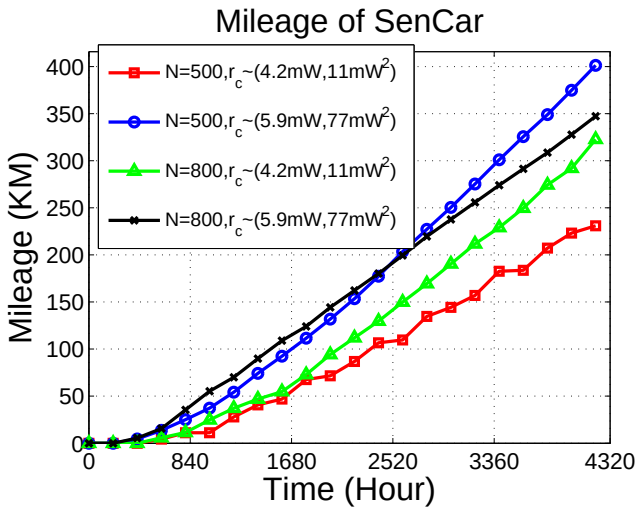



Figure 4.7: Mileages SenCar travels.

## 4.4 Wireless Energy Replenishment through Multiple SenCar Coordination

In this section, we extend the study to recharge optimization with multiple SenCars problem. We first derive the necessary condition for the network to operate perpetually and the minimum number of SenCars needed to satisfy this condition. Our design objective is to minimize the total traveling cost of the SenCars while guaranteeing recharge before sensors' battery depletion. We formalize this problem into a Multiple Traveling Salesmen Problem with Deadlines. We show our problem is NP-hard and propose a heuristic algorithm suitable for dynamic real-time recharging.

### 4.4.1 Theoretical Analysis

The energy neutral condition for one SenCar scenario we discussed in 4.3.1 also holds for multiple SenCar recharging application. Based on this condition, we have the following Proposition 1.

**Proposition 1.** *The minimum number of SenCars required to achieve perpetual*

*operation is*

$$S = \left\lceil \frac{t_r N(2.33\sqrt{np(1-p)} + np - E_0)}{Cn} \right\rceil$$

*Proof.* Since $\Phi^{-1}(1) \to \infty$, we consider the sensor network achieves perpetual operation when $P_{op} \geq 0.99$. From $\Phi^{-1}(0.99) \leq \frac{\frac{nCS}{t_r N} + E_0 - np}{\sqrt{np(1-p)}}$, we obtain the minimum number of SenCar $S$. □

The derivation from Proposition 1 can help network administrator plan the network. Once the experimental parameters and the application specifics from the sensors have been determined (e.g., network size $N$, recharge time $t_r$, initial energy $E_0$, working probability $p$, operation duration $n$ and battery capacity $C$), we can easily obtain the minimum number of SenCars needed. As will be seen later, we also validate the correctness of the derivation in simulations.

### 4.4.2 Problem Formulation

Given a set of SenCars $\mathcal{S}$ and a set of emergency nodes $\mathcal{M}$, we formalize the problem as follows. Consider a graph $G = (V, E)$, where $V_0^{(k)}$ is the starting position of SenCar $k$, and $V_i$ ($i \in \mathcal{M}$) is the location of emergency sensor $i$ to be visited. $E$ is the set of edges. Each edge $E_{ij}$ has a latency cost $c_{ij} = t_i + t_{ij}$, where $t_i$ is the time to recharge node $i$ from its current energy level to full capacity, and $t_{ij}$ is the traveling time from node $i$ to node $j$. For SenCar $k$, $c_{0j}^{(k)}$ represents its cost from its initial position 0 to node $j$. For each sensor node $i$, the residual lifetime is $L_i$. $A_i$ specifies the arrival time for a SenCar at sensor node $i$.

We introduce decision variables $x_{ij}$ for edge $E_{ij}$. The decision variable is 1 if an edge is visited, otherwise it is 0. Additionally, $x_{0j}^{(k)}$ is 1 if SenCar $k$ moves from its initial position to node $j$. $u_i$ is the position of vertex $i$ in the path. We virtually make the SenCars return to $V_0^{(k)}$ after recharging all the selected nodes by setting $c_{i0}^{(k)} = 0, i \in \mathcal{M}$, thus the EROMS problem can be formulated as the Multiple Start Traveling Salesman Problem with Deadlines in which multiple traveling salesmen start from different locations to visit a set of cities within their deadlines.

$$\mathbf{P1}: \quad \min \left\{ \sum_{i=1}^{M} \sum_{j=1}^{M} c_{ij} x_{ij} + \sum_{k=1}^{S} \sum_{j=1}^{M} c_{0j}^{(k)} x_{0j}^{(k)} \right\} \tag{4.13}$$

**Subject to**

$$\sum_{j=1}^{M} x_{0j}^{(k)} = \sum_{i=1}^{M} x_{i0}^{(k)} = 1, \forall k = 1, 2, \ldots, S, \tag{4.14}$$

$$\sum_{i=1}^{M} x_{ik} = \sum_{j=1}^{M} x_{kj} = 1; \forall k = 2, \ldots, M, \tag{4.15}$$

$$A_i \leq L_i; \forall i = 1, 2, \ldots, M, \tag{4.16}$$

$$x_{ij} \in \{0, 1\}; \forall i, j = 1, 2, \ldots, M, \tag{4.17}$$

$$2 \leq u_i \leq M; \forall i = 2, 3, \ldots, M, \tag{4.18}$$

$$u_i - u_j + (M - S)x_{ij} \leq M - S - 1; \quad \forall i, j = 2, 3, \ldots, M, i \neq j. \tag{4.19}$$

Constraint (4.14) guarantees that the recharge path starts at 0 and finishes at 0. Constraint (4.15) ensures the connectivity of the path and that every vertex is visited at most once. Constraint (4.16) guarantees the arrival time of the SenCar is within sensor's residual lifetime. Constraint (4.17) imposes $x_{ij}$ to be 0-1 valued. Constraints (4.18) and (4.19) eliminate the subtour in the planned route. The subtour elimination constraints are formulated according to [90, 91].

We now show that EROMS is NP-hard. If we remove Constraint (4.16) and set all the SenCars to start from one position, the problem becomes finding the shortest tour of visiting every sensor exactly once by multiple SenCars with sensors having infinite lifetime, which is known to be another NP-hard problem, Multiple Traveling Salesman Problem (m-TSP). Thus EROMS is NP-hard.

Condition to Cover All Emergencies We have known that the EROMS problem is NP-hard. In this subsection, we derive a condition for the residual battery lifetime $\mathcal{L}$ such that none of the sensor node would deplete energy for a given set of $\mathcal{M}$ concurrent emergencies. We know that running the Multiple Traveling Salesmen Problem on $\mathcal{M}$ should give the shortest distance to cover all the emergencies using $S$ SenCars. The result contains a number of $S$ node sequences and for a node in the $i$-th position of a sequence to survive, its residual lifetime should be larger than the sum of recharge time of all the previous $(i-1)$ nodes plus the traveling time from

the beginning to its position,

$$L_i \geq \sum_{j=1}^{i-1} r_j + \sum_{j=1}^{i} \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2}, i = 1, \ldots, M/S. \tag{4.20}$$

Actually, as pointed out in [92] that even checking the feasibility of EROMS is an NP-complete problem, checking the above condition requires solving the m-TSP problem that is too expensive during the emergency recharge. However, it is beneficial to find a cost-effective way to know if the number of emergencies has exceeded the capacity of the network during real-time operation. Hence, we have the following lemma,

**Lemma 3.** *If* $\max(\mathcal{L}) < \lceil \frac{M}{S} \rceil r_e$, $r_e$ *is the minimum recharge time for emergency nodes, the EROMS problem is infeasible.*

*Proof.* □

Based on this condition, we propose a heuristic algorithm that can schedule and coordinate SenCars in an efficient manner.

### 4.4.3 Minimum Weighted Sum Heuristic Algorithm and Complexity Analysis

In this subsection, we propose a heuristic algorithm for the EROMS problem that jointly considers the residual lifetime and traveling time. As mentioned above, when all the SenCars have the same initial position, the problem becomes the Multiple Traveling Salesmen Problem with Deadlines. In general, the m-TSP with Deadlines can be considered as a special case of the *Multiple Traveling Salesman Problem with Time Windows* (m-TSPTW)[2]. This problem is related to Vehicle Routing Problem with Time Window (VRPTW) which has been studied in the literature and a handful of optimal and approximation algorithms are available [92–96].

The approaches to VRPTW are usually divided into two phases. A construction of a feasible tour is sought in the first phase and the tour is interactively improved in the second phase. In [92], a local search algorithm was proposed to reduce the computation of checking feasibility constraint of TSPTW. In [93], the minimum

[2]m-TSP with Deadlines is m-TSPTW having all the release time at 0.

number of vehicles to meet the time window requirements was studied by utilizing precedence graphs. However, since checking the tour feasibility is as hard as the original problem [92], these approaches are still computationally expensive.

Several approximation algorithms have been proposed for the VRPTW problem in [94, 95]. However, these algorithms are not suitable for the recharging problem context. First, they assume the number of vehicles is unlimited but the number of SenCars is bounded. Second, existing algorithms deal with a static problem input. However, in EROMS, new emergencies may appear at any time, and residual node lifetimes also vary due to ongoing sensing activities. Maintaining an optimal schedule would become prohibitively expensive. Finally, existing algorithms may generate unbalanced workloads among SenCars, resulting in idling SenCars while emergencies still exist.

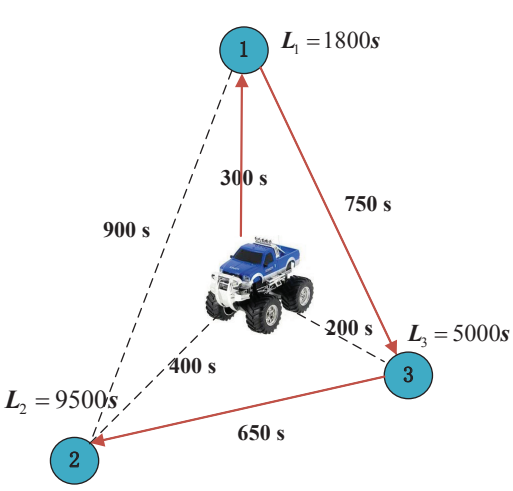
We present a heuristic algorithm that schedules recharge assignments among SenCars. Two important metrics impact the recharging order between node $i$ and node $j$: the traveling time between node $i$ to node $j$, and their residual lifetime $L_i$ and $L_j$. If node $j$ has a small $L_j$ such that it would be dead if a SenCar finishes recharging node $i$ first and then travels to node $j$, it should be visited first.

We use a weighted sum $w_{ij}$ of traveling time from the current node $i$ to next node $j$ and the residual lifetime of node $j$,

$$w_{ij} = \alpha t_{ij} + (1 - \alpha) L_j. \tag{4.21}$$

$w_{ij}$ is used to decide which node $j$ to recharge next. A sensor node with a smaller weighted value should be visited at a higher priority. When $\alpha = 1$, the algorithm reduces to nearest node selection that the SenCars always recharge the closest node first regardless of battery deadlines; when $\alpha = 0$, it picks the node with the earliest battery deadline first regardless of the traveling distance.

The value of $\alpha$ greatly affects the schedule. Fig. 4.8 gives an example of a SenCar and 3 sensor nodes. The residual lifetime and the traveling time on each edge have been shown in the figure, and $\alpha$ varies from 0, 0.5 to 1. We assume that recharging a sensor battery takes 1 hour to finish. At time 0 s, the SenCar calculates the weight to sensor nodes 1, 2 and 3. The minimum weights have been circled. When $\alpha = 1$, node 3 has the minimum weight; when $\alpha = 0.5, 1$, node 1 has the minimum weight. However, if node 3 is chosen to be the next node, node 1 would have been dead after finishing recharging node 3. Thus $\alpha = 1$ is infeasible in this




Current time: 0s

| Weight | $\alpha = 1$ | $\alpha = 0.5$ | $\alpha = 0$ |
|---|---|---|---|
| $w_{01}$ | 300 | 1050 | 1800 |
| $w_{02}$ | 400 | 2600 | 9500 |
| $w_{03}$ | 200 | 4950 | 5000 |

Current time: 3900s

| Weight | $\alpha = 0.5$ | $\alpha = 0$ |
|---|---|---|
| $w_{12}$ | 3250 | 5600 |
| $w_{13}$ | 925 | 1100 |

Figure 4.8: Example of minimum weighted sum algorithm.

example. $\alpha = 0.5$ and $\alpha = 0$ generate the same schedule $1 \rightarrow 3 \rightarrow 2$.

From this example, we can see that the value $\alpha$ affects the feasibility of the solution. We might expect that the total distance be inverse proportional to $\alpha$ and a binary search may locate the maximum feasible $\alpha$. However, some preliminary tests have shown that it is not always true. We decide to search through a list of $A$ distinct $\alpha$ values, e.g., $\alpha = 0.0, 0.1, \ldots, 1.0$ where $A = 11$. We find that this choice achieves a desirable tradeoff between optimality and complexity.

When a SenCar performs calculation, it communicates via a long range radio with other SenCars to know their positions for computing the weighted sum. To avoid conflicts where multiple SenCars choose the same node for recharge, we utilize the service station to store and update the availability of each node. The procedure is similar to memory access in operating systems [97]. The service station maintains a 0-1 valued node list $\Omega$. Once a sensor is chosen, its value is set to 1 (locked). Otherwise, it is 0. The value should be changed back from 1 to 0 when a SenCar finishes recharging that node. A SenCar can simply communicate with the service station, exclude nodes already selected by other SenCars, and notify the service station of the status of nodes it chooses. Table 4.4 shows the pseudo-code of the entire algorithm.

We now analyze the complexity of the heuristic algorithm. Note that the node selection operations are executed on each SenCar, which takes $\mathcal{O}(M + \log M)$ time.

Table 4.4: Minimum Weighted Sum Algorithm.

**Input**: weight parameter $\alpha \in [0, 1]$ in stepsize $1/(A-1)$, position of SenCar at node $k$, emergency set $\mathcal{M}$, traveling time from $i$ to $j$, $t_{ij}$, residual lifetime $L_i, \forall i, j \in \mathcal{M}$, node list $\Omega_i$ at service station, $i \in \mathcal{N}$.
**Output**: result weight parameter $\alpha$ and schedule sequence $Q$.
Initialize minDist $= \infty$
**For** $\alpha = 0, \ldots, 1$
**While** $\mathcal{M} \neq \emptyset$
Compute weight $w_{kj} \leftarrow \alpha t_{kj} + (1-\alpha)L_j$.
Communicate service station **If**
$\Omega_i = 1$, Set $w_{ki} = \infty$.
**End if**
Find $j \leftarrow \arg\min_j w_{kj}$.
$Q_t \leftarrow Q_t + j$, $M \leftarrow M - j$.
update $\forall i \in M$, $L_i \leftarrow L_i - t_{kj} - t_j$.
**If** $L_i \leq 0$
Declare infeasible and break (Inform service station).
**End if**
Move to position $j$, $k \leftarrow j$, recharge and update lifetime $L_j$
**End while**
**If** feasible
Compute total cost dist($Q_t$).
**If** dist($Q_t$) $<$ minDist,
minDist $\leftarrow$ dist($Q_t$), $Q \leftarrow Q_t$.
**End if**
**End if**
**End for**

For each SenCar, it performs $M/S$ rounds of node selections and the total number of tests on $\alpha$ is $A$. Thus, the total computational complexity of our heuristic algorithm is $\mathcal{O}(\frac{AM}{S}(M + \log M))$.

### 4.4.4 Performance Evaluations

In this section, we use simulation to evaluate the effectiveness and efficiency of our framework. We have developed a discrete event-driven simulator using POSIX thread programming in C language. We examine two network sizes of 500 and 1000 sensor nodes, uniformly randomly distributed over a $200 \times 200m^2$ and $282 \times 282m^2$ square field, respectively. The field size is chosen so that the two cases have the same node density. The network consists of 3-level hierarchy with $4^l$ number of subareas at the $l$-th level. The energy consumption on each sensor is a Bernoulli random variable with probability $p$ to consume unit energy (37.5 mJ). If a sensor node works continuously at this rate, the battery can last for 5 days. The relationship between recharged energy and recharge time follows that of Panasonic Ni-MH AAA battery [86]. To understand the impact of the number of SenCars on network performance, we show marginal cases where the number of SenCars is not sufficient while adding one more SenCar would guarantee perpetual operations. These cases are $S = 2, 3$ for $N = 500$ and $S = 4, 5$ for $N = 1000$. We will show these cases in the following and validate the correctness of Proposition 1. All the parameter settings in the simulation are listed in Table 4.5.

#### 4.4.4.1 Evaluation of Weighted-sum Algorithm

In this subsection, we evaluate the effectiveness of the weighted-sum algorithm in finding the shortest path and achieving no node failure. We examine cases when 4 SenCars are employed. We assume the locations of emergencies are randomly distributed in the field of $282 \times 282m^2$, and the residual energy uniformly distributed from zero to the emergency threshold. The corresponding residual lifetime is calculated by dividing the residual energy by $pr_c$, the expected energy consumption in unit time.

Table 4.6 shows the total distance of SenCars when the number of concurrent emergencies $M$ increases from 72 to 96 in a step of 8. Note that when the number reaches 96, the set of 4 SenCar is not sufficient to resolve all the emergencies with-

Table 4.5: Parameter settings for performance evaluation of optimal recharge using multiple SenCars.

| Parameter | Value |
|---|---|
| Field Length | $200 \times 200$, $282 \times 282m^2$ |
| Number of Nodes $N$ | 500, 1000 |
| Number of SenCars $S$ | 2, 3, 4, 5 |
| Number of Levels | 3 |
| Areas on $l$-th level | $4^l$ |
| Battery Capacity | 780 mAh |
| Transmission Range | 18 m |
| Unit Energy Consumption $r_c$ | 37.5 mJ |
| Energy Consumption Probability $p$ | 0.5 |
| SenCar Speed | 1 m/s |
| Maximum Recharge Time | 73.4 mins |
| Normal Recharge Threshold | 50% |
| Emergency Recharge Threshold | 10% |
| Simulation Time | 6 months |

out complete battery depletion. For $M = 88$, $\alpha = 0.8, 1$ are not feasible and for $M = 72, 88$, $\alpha = 1$ is not feasible either. We notice that in the case when $\alpha = 1$, some nodes that suffer from energy shortage may not get recharged in a higher priority thereby rendering the result infeasible to avoid battery depletion. As we can see from this example, the choice of $\alpha$ is critical, when $\alpha$ approaches 1, the total distance is decreased at the risk of becoming infeasible. Thus, we need to search for $\alpha$ in our algorithm. In real applications, the value of $\alpha$ is subject to change and determined by real-time statistical data and parameters.

Table 4.6: Total traveling distance of SenCars, $D$.

| $M$ | $D$ ($\alpha = 0$) | $D$ ($\alpha = 0.2$) | $D$ ($\alpha = 0.4$) |
|---|---|---|---|
| 72 | 7524.1 | 7473.3 | 7740.2 |
| 80 | 7652.4 | 7578.9 | 7706.6 |
| 88 | 8662.6 | 8128.3 | 7251.6 |
| 96 | Infeasible | Infeasible | Infeasible |
| $M$ | $D$ ($\alpha = 0.6$) | $D$ ($\alpha = 0.8$) | $D$ ($\alpha = 1$) |
| 72 | 6843.5 | **6390.6** | Infeasible |
| 80 | 7271.8 | **6941.0** | Infeasible |
| 88 | **6998.3** | Infeasible | Infeasible |
| 96 | Infeasible | Infeasible | Infeasible |

#### 4.4.4.2 Performance Evaluations

In this subsection, we evaluate the energy evolution of the network, the number of emergency and nonfunctional (i.e., energy depleted) nodes of the network, and the maintenance cost of the framework.

**4.4.4.2.1 Energy Evolution and Energy Distribution** First, we show the energy evolution in the network of 500, 1000 nodes served by different numbers of SenCars. In Fig. 4.9, the amount of energy consumed and replenished in every one-hour time slot is plotted as functions of simulation time. In Fig. 4.9(a) and (c), we can see that the consumed energy "steps down" to a lower level around 400 hours and then enters equilibrium. This is because that a portion of sensor nodes deplete their energy and do not get recharged. In these two scenarios the energy neutral condition has been violated, simply because the number of SenCars is not enough. Fig. 4.9(b) and (d) show the energy evolution when the numbers of SenCars is increased by 1, both of which satisfy the energy neutral condition at the equilibrium and there is no such "step-down" effect in energy consumption.

The energy distribution among nodes also carries valuable information about the health of the network. Higher average energy distribution is more robust to unexpected surges in energy consumption. Fig. 4.10 shows the energy distribution of $N = 500$, $S = 2, 3$ and $N = 800$, $S = 4, 5$. To see the benefits of more SenCars, compare Fig. 4.10(a) to Fig. 4.10(b). The latter has energy distribution that concentrates around a higher average value. In Fig. 4.10(d) for a network size of 1000 sensors, the number of nodes with energy below the emergency threshold is significantly lower than that in Fig. 4.10(c).

#### 4.4.4.3 Number of Emergencies

Fig. 4.11 and Fig. 4.12 compare the percentage of nodes in emergency and nonfunctional (i.e., energy at zero) status for networks of 500 and 1000 nodes with different numbers of SenCars. First, we can see that there are surges in the numbers of emergency and nonfunctional nodes during the first 200 hours. This is due to the fact that the SenCars only responds to requests when the node energy is below the normal recharge threshold. When such requests swarm into the job queues on the SenCars at the beginning of 200 hours, we can see that the SenCars' capacity has

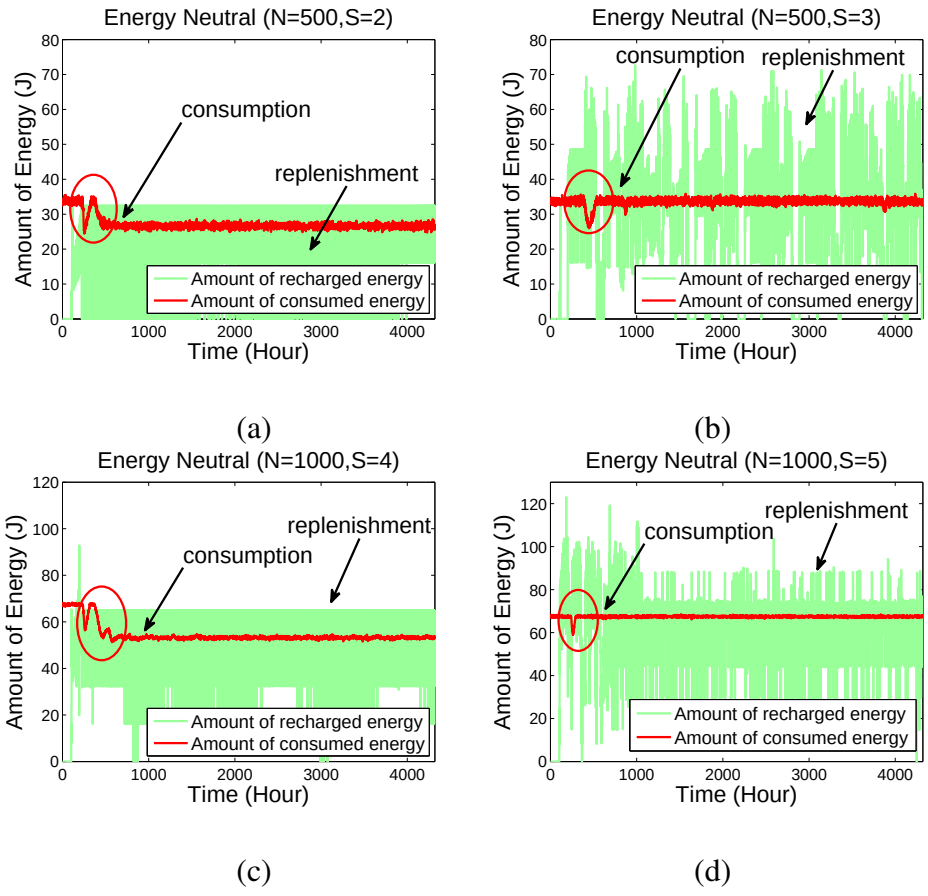



Figure 4.9: Evolution of energy consumption vs. energy replenishment in 6 months time. (a) Number of nodes $N = 500$, number of SenCars $S = 2$. (b) Number of nodes $N = 500$, number of SenCars $S = 3$. (c) Number of nodes $N = 1000$, number of SenCars $S = 4$. (d) Number of nodes $N = 1000$, number of SenCars $S = 5$.

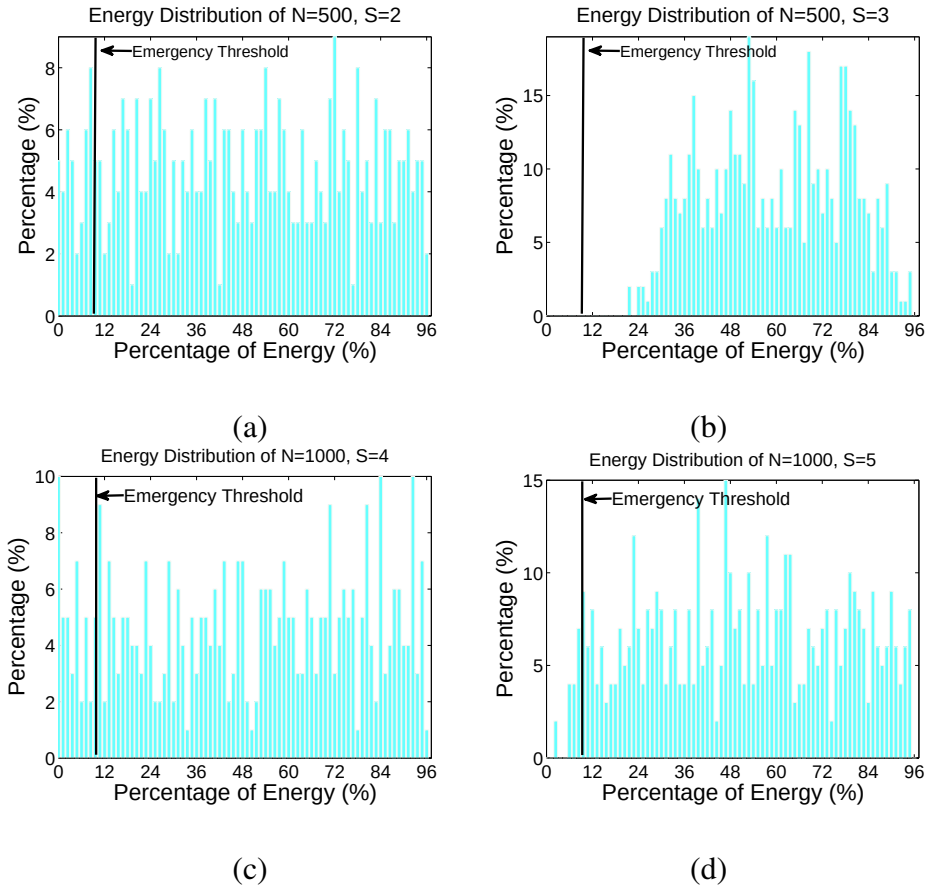



Figure 4.10: Energy distribution at equilibrium. (a) Number of nodes $N = 500$, number of SenCars $S = 2$. (b) Number of nodes $N = 500$, number of SenCars $S = 3$. (c) Number of nodes $N = 1000$, number of SenCars $S = 4$. (d) Number of nodes $N = 1000$, number of SenCars $S = 5$.

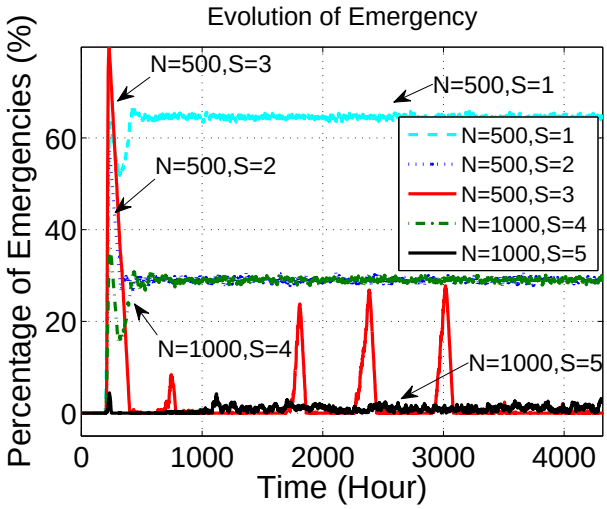



Figure 4.11: Number of emergency nodes.

been temporarily exceeded. As the energy of sensors is restored, the numbers of emergency and nonfunctional nodes decrease sharply. For cases $N = 500, S = 2$ and $N = 1000, S = 4$ when the number of SenCars is not sufficient for the sizes of the network, we can see that about 30% nodes are in constant emergency and 20% nodes are in nonfunctional status after the network achieves equilibrium. For $N = 500, S = 3$, there are occasional nonfunctional nodes but they were soon recharged by the SenCars. For a majority of the time, the number of nonfunctional nodes stays at zero. For $N = 1000, S = 5$, the number of nonfunctional nodes stays at zero at equilibrium with only a small number of emergencies.

Recall from Proposition 1 that the minimum number of SenCars for $N = 500$ and $N = 1000$ can be calculated as $S = \lceil 2.41 \rceil = 3$ and $S = \lceil 4.84 \rceil = 5$ for the given parameter settings in Table 4.5. These numbers match well with our simulation results that $S = 3, 5$ are the minimum number of SenCars to achieve perpetual operation at equilibrium, respectively. By utilizing our theoretical analysis, the network administrator can make reasonable estimations for the minimum number of SenCars needed when planning a network.

#### 4.4.4.4 Protocol Overhead and SenCar Workload

**4.4.4.4.1 Evaluation of Protocol Overhead** We evaluate the overhead introduced by our protocol, including all types of messages sent by sensor nodes or the

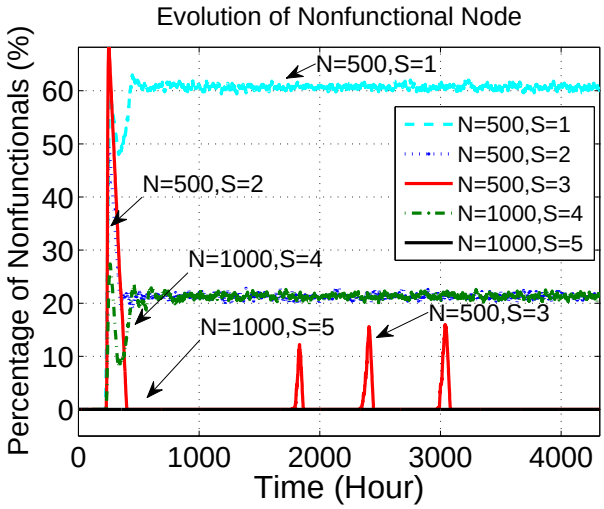



Figure 4.12: Number of nonfunctional nodes .

SenCar to recharge nodes. Fig. 4.13 shows the average overhead per node in a 6 month period. After the networks enter equilibrium, the overhead on each sensor node is from 8 to 48 bits per second, which is negligible compared to radio transmission rates in sensor nodes (e.g., 20 - 900 kbps).

From Fig. 4.13, we can observe that all the four scenarios have a large amount of message transmission when simulations start up. Such bursts are caused by simultaneous head selection processes in all the sub-areas, during which a lot of messages are broadcast. Energy information query also contributes to the bursts which also leads to message broadcast. As time elapses, however, the energy levels of the nodes drop and emergency occurs in the network, so that the top level heads, when receiving *energy interest* messages, response with emergency information instead of querying lower levels for energy information. Consequently the amount of messages transmitted in the network decreases.

For $N = 500, S = 2$ and $N = 1000, S = 4$, there are a large number of emergency nodes after the networks enter equilibrium (Fig. 4.11). Thus the SenCars always query for emergency information. In our protocol, *emergency interest* messages sent by the SenCars are relayed directly by proxies so the overhead of message forwarding is quite small.

As the number of SenCars increases, the number of emergencies decreases dramatically and a majority of the time the SenCars query for normal energy informa-

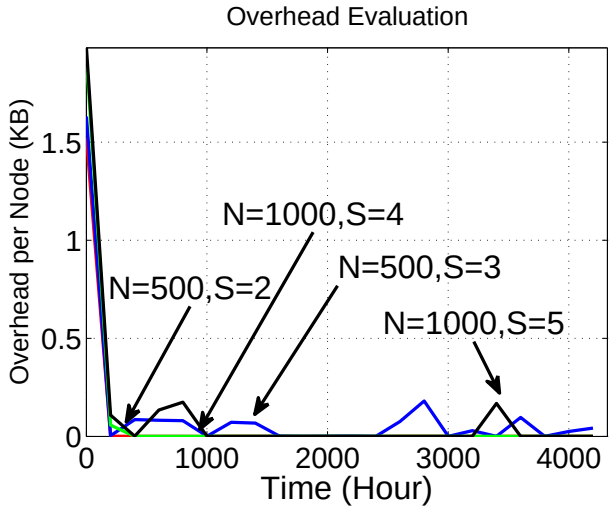



Figure 4.13: Average overhead for each sensor node per second.

tion and perform normal recharge. In response to *energy interest*, the heads poll energy information in a top-down method which finally results in the broadcast of *energy interest* message in subareas at the bottom level. Such broadcast, as well as the transmission of energy information sent by each node, causes the increase of the number of messages transmitted in the networks, which is observed as the spikes in the curves.

#### 4.4.4.5 Evaluation of Balance of Load on SenCars

We monitor the energy replenished by each SenCar and compare their workloads. Fig. 4.14 shows that the workloads are well balanced in all four scenarios due to the effective coordination in our framework. The SenCars shares the work evenly and no SenCar is overloaded.

#### 4.4.4.6 Evaluation of Mileage on SenCars

We use the the mileages the SenCars travel to evaluate the cost (e.g., the energy consumed) for the SenCar to move around. Fig. 4.15 shows the accumulated mileages in 6 months.

For both network sizes, the networks with fewer SenCars (500 nodes and 2 SenCars, 1000 nodes and 4 SenCars) have lower mileage compared with the same

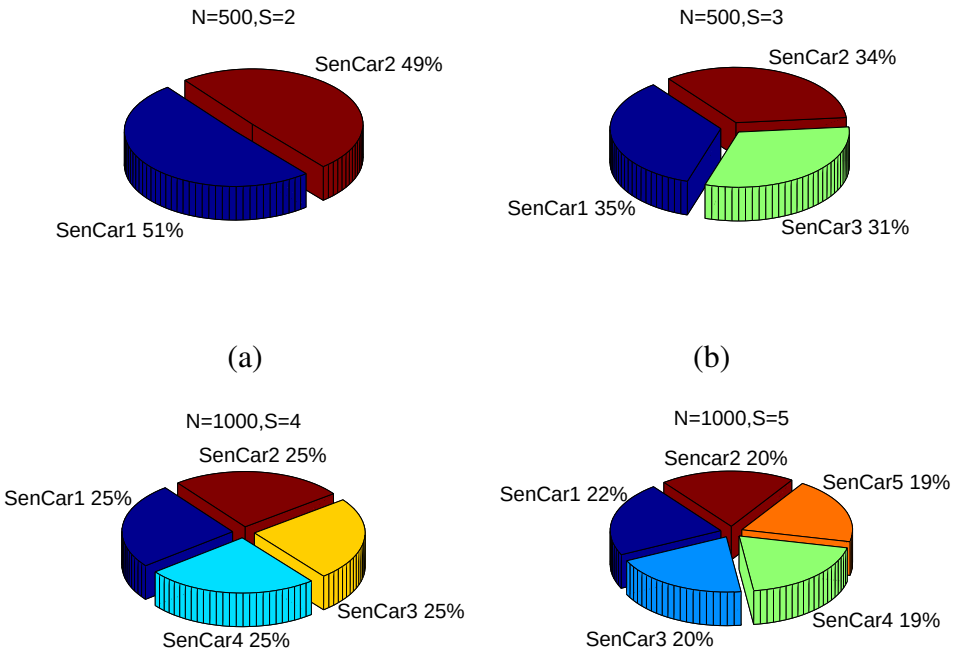



Figure 4.14: Balance of load on SenCars. (a) Number of nodes $N = 500$, number of SenCars $S = 2$. (b) Number of nodes $N = 500$, number of SenCars $S = 3$. (c) Number of nodes $N = 1000$, number of SenCars $S = 4$. (d) Number of nodes $N = 1000$, number of SenCars $S = 5$.

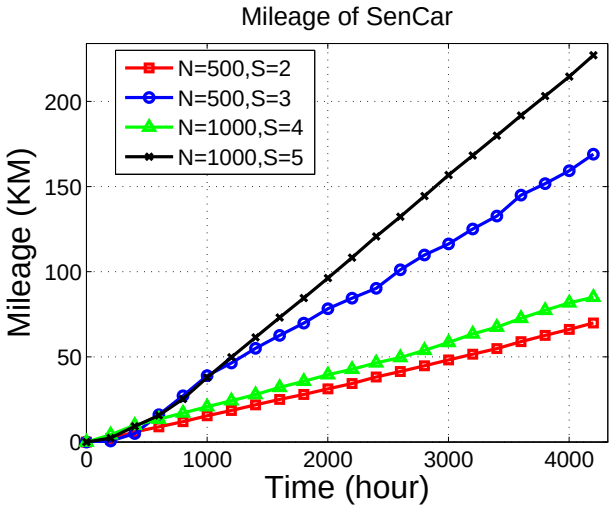



Figure 4.15: Mileages SenCars have traveled in 6 months time.

network with more SenCars (500 nodes and 3 SenCars, 1000 nodes and 5 SenCars), respectively. This is due to the presence of nonfunctional nodes. According to the calculation of weight for emergency selection (Eq. (4.21)), decision is made based on the residual lifetime of the nodes and the traveling time from the SenCars to the nodes. For the networks with fewer SenCars, there are always approximate 20% nonfunctional nodes after the networks enter equilibrium. The weights are dominated by the traveling time which is proportion to the distances from the SenCars to these nodes. Thus the SenCars always choose the nearest nodes for recharge. For the network with more SenCars, however, the traveling time is not the dominating factor, thus the SenCars may choose a farther node with shorter residual lifetime for recharge to avoid battery depletion. This causes the increase of SenCar mileage.

### 4.4.5 Discussions

The simulation results have demonstrated the effectiveness of our framework for handling both emergency and normal recharge requests. We also validate our theoretical derivations on the minimum number of SenCars for perpetual operation. The results also show that coordinating multiple SenCars to perform the recharge assignments improves the network scalability and immunity to burst of emergencies. The design of our framework distributes workload evenly among the SenCars and the communication overhead is negligible compared to the data rate in sensor

networks.

## 4.5 Conclusions

In this chapter, we propose a novel framework for real time wireless energy recharging for wireless sensor networks. We develop a comprehensive set of protocols using NDN concepts and mechanisms to enable effective recharging for the perpetual operation of the network. The protocols adapt to unpredictable network conditions and satisfy the needs for both normal and emergency recharging. We formally analyze the probability for the energy neutral condition required by perpetual operations. We also model the optimal recharging of multiple emergencies as an Orienteering problem and provide a Knapsack approximation that has high accuracy under typical network environments. The extensive simulation results demonstrate the efficiency and effectiveness of our framework, and the close match of energy neutral analysis with simulation results.

# Chapter 5

# Mobility Assisted Data Gathering with Solar Irradiance Awareness in Heterogeneous Energy Replenishable Wireless Sensor Networks

Wireless sensor networks adopting static data gathering may suffer from unbalanced energy consumption due to non-uniform packet relay. Although mobile data gathering provides a reasonable approach to solving this problem, it inevitably introduces longer data collection latency due to the use of mobile data collectors. In the meanwhile, energy harvesting has been considered as a promising solution to relieve energy limitation in wireless sensor networks. In this chapter, we consider a joint design of these two schemes and propose a novel two layer heterogeneous architecture for wireless sensor networks, which consists of two types of nodes: sensor nodes which are static and powered by solar panels, and cluster heads that have limited mobility and can be wirelessly recharged by power transporters. Based on this network architecture, we present a data gathering scheme, called Mobility Assisted Data Gathering with Solar Irradiance Awareness (MADG-SIA), where sensor nodes are clustered around cluster heads that adaptively change their positions according to solar irradiance, and the sensing data are forwarded to the data sink by these cluster heads working as data aggregation points. We evaluate the performance of the proposed scheme by extensive simulations and the results show

that MADG-SIA provides significant improvement in terms of balancing energy consumption and the amount data gathered compared to previous work.

## 5.1 Introduction

Wireless sensor networks suffer from energy constraint as the sensors usually have limited energy supply. Since the sensory data is usually collected in ad-hoc manner, which consumes a lot of energy on packet relay. thus how to gather data from sensor nodes is an important issue for energy consumption optimization in sensor networks. One of the most important problems in data gathering in WSNs is to balance the energy consumption among sensor nodes. Unbalanced energy consumption caused by packet relay leads to energy holes in the network, which may disable the packet forwarding towards the data collector, and eventually results in degraded network performance, such as short network lifetime and low data throughput. In a conventional homogeneous wireless sensor network, a statically deployed data collector, referred to as *data sink*, is used to gather data from the network. Since data packets converge towards the data sink, the nodes that are closer to the data sink have to relay much more data than the nodes that are farther away from the data sink, and they consume energy much faster than other nodes. When these nodes deplete their energy, the data sink becomes unreachable to the rest of the nodes, thus the entire network can no longer operate.

Hierarchical WSNs have been proposed to relieve the unbalanced energy consumption problem, in which sensor nodes are organized into clusters [25, 98–103]. Instead of sending all the data to the single data sink in a multi-hop manner, sensors upload data to the aggregation nodes of the cluster they belong to, which are referred to as *cluster heads*. These data are then relayed to the data sink by cluster heads which are built with stronger wireless communications capability and more energy supply. Such a hierarchical architecture can mitigate energy unbalance to some extent, however, since cluster heads are statically deployed, network lifetime is limited by the nodes around these cluster heads.

Mobility has been introduced into WSNs due to its benefits, such as guaranteeing network connectivity, reducing network cost, increasing reliability, and improving energy efficiency [104]. However, a challenge of mobile data gathering is that such a scheme inevitably introduces long data collection latency because the

mobile data collector has to visit all the selected positions before it can upload data to the data sink. The positions of data aggregation points also need to be carefully selected depending on multiple factors, such as network topology, energy levels of all the sensor nodes, and the amount of sensing data generated in each sensor node. The computation inevitably increases overhead and operational complexity of the network.

Energy harvesting techniques have been recently employed as a solution to prolong network operating time from another aspect. Such a scheme captures energy from the ambient environment, e.g., mechanical, thermal, photovoltaic or electromagnetic energy, to charge sensor nodes. However, the main drawback of energy harvesting techniques is the low efficiency of recharging, since the power output of energy harvesting devices is relatively low compared to the power consumption of the node for sensing and communications [106], especially for the sensor nodes around the data aggregation points.

In order to overcome the aforementioned problems, it is desirable to find a novel approach to balancing energy consumption to improve network performance in WSNs. By taking advantage of mobility and renewable energy while shortening data collection latency, we propose a **Mobility Assisted Data Gathering with Solar Irradiance Awareness** scheme, abbreviated as **MADG-SIA**, to achieve balanced energy consumption in WSNs and prolong network lifetime. We construct a MADG-SIA enabled network with three types of devices: the static sensor nodes that are powered by solar panels, the mobile cluster heads that can be wirelessly recharged, and the power transporters (referred to as *SenCars*). Different from sensor nodes, the discharge rates of cluster heads are much higher than the charge rate provided by solar panels due to the large amount of data forwarding and movement. We use SenCars to charge cluster heads more efficiently when their energy levels are low. The network has a hierarchical architecture where sensor nodes are clustered and send data to their corresponding cluster heads. The cluster heads sojourn at different positions, referred to as *anchor points*, in each data gathering period, to collect data from surrounding sensor nodes in a one-hop or multi-hop manner. By carefully moving cluster heads to some positions for data gathering, the energy-consuming data forwarding tasks are shared among all the sensor nodes to balance their energy consumption. On the other hand, as the moving distance of cluster heads is limited, and they communicate with the data sink directly once

settled down, the data collection latency in the proposed network is much shorter than that in mobile data gathering. Based on this network architecture, we will find optimal positions for anchor points and moving paths for cluster heads, and develop a clustering approach and determine the routes for sensor nodes to upload data to cluster heads.

The rest of this chapter is organized as follows. The related work is discussed in Section 5.2. Section 5.3 describes the framework of the proposed scheme. In Section 5.4, the system model and the proposed algorithm are presented for WSNs with regular and random topologies. Section 5.5 evaluates the impact of various parameters on network performance. Finally, the chapter is concluded in Section 5.6.

## 5.2 Related Work

In this section, we briefly review some related work in the literature, which includes the work on data gathering with clustering and energy replenishment in WSNs.

### 5.2.1 Data Gathering in Clustered WSNs

An energy-efficient framework for clustering-based data collection in wireless sensor networks was proposed in [98]. By adaptively enabling/disabling prediction operations and updating clustering as well as accommodating in-network aggregation and the sleep/awake scheduling, the framework achieves energy efficiency when sensor data is spatially and temporally correlated. Clustering sensor nodes was also considered in [25]. By periodically selecting cluster heads according to the combination of residual energy and a secondary parameter, such as node proximity to its neighbors or node degree, this clustering method outperforms weighted clustering protocols in terms of several cluster characteristics. Heterogeneous ad hoc sensor networks were studied in [99] which focused on energy and link heterogeneity. The impact of the number and placement of heterogeneous resources on the performance of networks of different sizes and densities was evaluated. It was shown that it requires only a modest number of reliable, long-range backhaul links and line-powered nodes to have a significant impact. Employing mobile cluster heads in hybrid sensor networks was explored in [100]. A heuristic algorithm for positioning cluster heads and balancing traffic load in the network was proposed

and shown to be able to increase network lifetime after only a few rounds of adjustments. An energy-balanced dominating set based clustering scheme (EBDSC) was proposed in [102], where normal nodes broadcast the number of cluster head candidates around it. Each candidate calculates the median of such numbers received from its neighbors, and becomes a final cluster head with a probability inversely proportional to the median. Employment of distributed load balanced clustering and MIMO uploading techniques in wireless sensor networks was studied in [103], in which the sensors are organized into clusters by executing a distributed load balanced clustering (LBC) algorithm, which also generates multiple cluster heads in each cluster to balance the work load and facilitate MIMO data uploading.

The above schemes can greatly save energy by utilizing clustering compared to conventional relay routing in networks. However, since they are based on conventional sensor nodes and do not consider energy harvesting, the network lifetime is limited by the battery life. Alternatively, our work in this chapter jointly considers energy replenishment and cluster head movement with the goal to extend operation time of WSNs.

### 5.2.2 Energy Harvesting in WSNs

A general target coverage problem for a solar-powered active sensor network with a controllable sensing range was investigated in [107], where a near-optimal approximate solution was provided with a 60X improvement in speed at the cost of 8% reduction in the quality of coverage. A joint study of energy management and resource allocation problem for energy-harvesting sensors was presented in [65], in which the optimal sampling rate was explored based on the average energy replenishment rate, and a local algorithm was presented for each sensor to adapt the sampling rate according to short term fluctuations in recharging, with the objective of maintaining the battery at a target level. A dynamic energy-oriented scheduling method (DEOS) was proposed in [108] for multiple tasks allocation with a time-varying and limited energy constraint in energy harvesting wireless sensor networks. Simulation results indicated that DEOS is extremely lightweight and it effectively schedules tasks to utilize the dynamically available energy. RF radiation based energy replenishment for sensor networks was studied in [68], where a wireless charging system was developed and implemented so that sensor nodes get

charged by a mobile charger. However, experiment reveals that energy replenishment with this technique is of low efficiency and difficult to be used in large scale WSNs with perpetual operation.

The above works provide promising methodologies to relieve the energy constraint from the perspective of power supply. [65, 107, 108] focus on prolonging the lifetime of an energy harvesting sensor network, but do not consider mobility. As a result, the network lifetime is still fairly short due to unbalanced energy consumption in the network. In [68, 109–113], all the sensors in the networks are charged by the mobile vehicles. Since the capability for a vehicle to deliver energy to the sensor is limited, large scale networks that consist of thousands of sensors may demand a large numbers of vehicles to maintain perpetual operation, which ultimately leads to high cost for network operation. In contrast, our work takes advantage of controlled mobility, data rate allocation and heterogeneity of energy replenishment, to provide perpetual network operation and maximize data throughput.

## 5.3 Framework Overview

In this section, we provide an overview of MADG-SIA. The entire network consists of three types of devices: sensors, cluster heads and SenCars. All the sensors are equipped with solar panels that harvest energy from solar irradiance, while the cluster heads can be wirelessly recharged by SenCars. The application of wireless recharge in sensor networks is constrained by the slow charging problem for conventional batteries, such as Nickel-Metal Hydride (NiMH) and Lithium-ion (Li-ion) batteries which are widely used as energy storage devices for energy replenishable sensor nodes. Both NiMH and Li-ion batteries have such a problem as the optimal charging rate for Li-ion batteries that achieves the best performance is less than 1C, while for NiMH batteries, the optimal charging rate is even lower (C is determined by the nominal capacity of the battery). Under such restriction, it will take long time to fully charge a battery, which is unrealistic for large scale WSNs, thus we choose to charge the cluster heads only.

A wireless sensor network with MADG-SIA works as follows. The *area* where the sensor nodes are deployed is divided into several *regions* (the method of division is presented in Section 5.4), and a mobile cluster head is employed in each region for data gathering. To be more specific, the cluster head visits all the anchor points

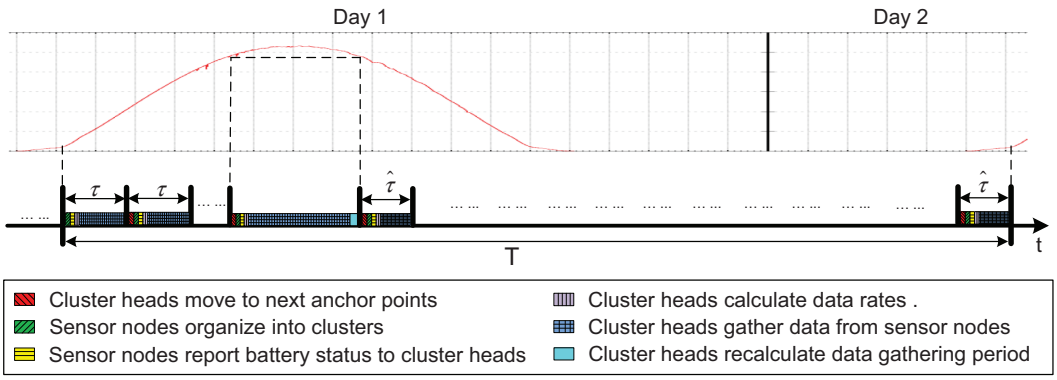



Figure 5.1: Operating timing diagram of a MADG-SIA enabled wireless sensor network. Recharging of cluster heads depends on their energy status and can be performed whenever necessary.

in the region once in a given time duration, which is denoted as a *data gathering cycle*. It sojourns at each anchor point for a period, which is denoted as a *data gathering period*, to collect data from the sensor nodes near this anchor point. An example of the operating timing diagram of a wireless sensor network with MADG-SIA in a data gathering cycle is shown in Fig. 5.1. The red dotted curve plots solar irradiance from real measurement. At the beginning of each data gathering period except for the first one of each data gathering cycle, all cluster heads move to next anchor points, and initiate clustering and routing operations. With the positions of the cluster heads, all the sensor nodes organize into clusters and find the routes to send data to the corresponding cluster heads. After that, data from sensor nodes are aggregated by cluster heads and then forwarded to the data sink via paths consisting of one or more cluster heads. It is possible that the energy levels of the nodes in a cluster increase due to the high output from solar panels. In such a scenario, the cluster head stays at the current anchor point until the batteries of the nodes begin to discharge. Then the cluster head recalculate the data gathering period for the rest anchor points, and move to these anchor points to collect sensory data. When a cluster head detects that its battery level is below a threshold, it reports its energy status to the base station and request energy replenishment. A SenCar is dispatched to recharge the cluster head upon the request.

## 5.4 Mobility Assisted Data Gathering with Solar Irradiance Awareness (MADG-SIA): Routing and Data Rate Control

Having outlined how the network works for energy replenishment and data gathering, in this section, we present an efficient implementation of MADG-SIA in WSNs in a distributed manner. The objective is to balance energy consumption during data gathering determine the positions of anchor points for cluster heads where they sojourn in each data gathering period to collect sensing data from their surrounding sensor nodes and forward these data to the data sink, the moving pattern of cluster heads among the anchor points, and the clustering and routing schemes of sensor nodes so as to guarantee perpetual network operation. This can be achieved in two steps: first, cluster heads find the anchor points they should move to for data collection; then sensor nodes calculate the clustering and routing according to the positions of anchor points. The notations and their semantics used for the following discussion are listed in Table5.1.

### 5.4.1 Anchor Point Selection

We first select the anchor points for cluster heads to visit in each data gathering cycle. Consider a wireless sensor network consisting of $n$ nodes randomly deployed in a $L_1 \times L_2$ area $\mathcal{S}$. Without loss of generality, we consider the case that sensor nodes follow uniform distribution, and each node is connected with at least one neighbor, which means that the distance between the two nodes is less than transmission range $R$.

An ideal solution to balance energy consumption for data transmission among sensor nodes in a given period is that each node transmits the same amount of data. In a large sensor network with random distribution, this can be approximately achieved by moving cluster heads to the anchor points where each node has an equal probability to be an $i$-hop neighbor of a cluster head ($i = 1, 2, \ldots,$). Since a sensor node can only communicate with a cluster head when the distance between them is no more than $R$, to achieve this objective with high efficiency, we cover the area with the fewest number of disks whose radius is $R$. It was proved in [114] that the

Table 5.1: Notations used in formulation of MADG-SIA.

| Notation | Definition |
|---|---|
| $\mathcal{S}$ | Rectangular area the wireless sensor network is deployed. |
| $m$ | Number of cluster heads. |
| $n$ | Number of sensor nodes. |
| $L_1, L_2$ | Side lengths of $\mathcal{S}$. |
| $\mathcal{D}_i$ | Set of descendant nodes of sensor $i$, i.e., the nodes for whom node $i$ relays data to the cluster head. |
| $R$ | Transmission range of a sensor node. |
| $e_r$ | Energy consumed for a sensor node to receive a unit flow. |
| $e_t$ | Energy consumed for a sensor node to transmit a unit flow. |
| $b_i$ | Battery level of sensor node $i$. |
| $b_i^t$ | Battery level of sensor node $i$ at time $t$. |
| $\check{b}_i^t$ | Estimated battery level of sensor node $i$ at time $t$. |
| $w_{i,j}$ | Weight of link $(i, j)$, which represents the capability of link $(i, j)$ to transmit data. |
| $w_{i-j-CH}$ | Accumulated weight of the route with least weight from sensor node $i$ to the cluster head through sensor node $j$. |
| $w_{i-CH}$ | Accumulated weight of the route with least weight from sensor node $i$ to the cluster head. |
| $r_i$ | Data rate of sensor node $i$. |
| $I$ | Solar irradiance arriving at the solar panel. |
| $\eta p$ | Efficiency of the solar panel to convert solar irradiance to electrical power. |
| $\rho_e$ | Electrical regulating and charging efficiency. |
| $A$ | Size of solar panel. |
| $\pi_i^d$ | Discharge rate of sensor node $i$. |
| $\pi^r$ | Recharge rate of sensor node. |

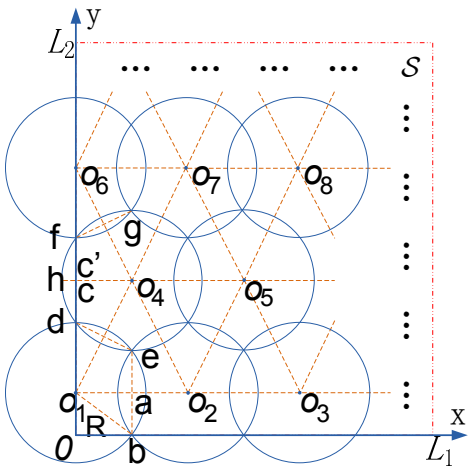



Figure 5.2: Placement of disks. The left- and bottom-boundaries are fully covered.

solution to the problem of covering an area with the minimum number of disks is to place disks so that the centers of disks are located at the vertices of equilateral triangles with side length equal to $\sqrt{3}R$.

It is easy to see that as long as the four boundaries of $\mathcal{S}$ are covered by disks placed in this way, it is guaranteed that $\mathcal{S}$ can be covered entirely. Based on this observation, we can derive a method of placing disks to cover $\mathcal{S}$ by considering the coverage of its boundaries.

In Fig. 5.2, $\mathcal{S}$ is defined by boundaries $x = 0$, $x = L_1$, $y = 0$ and $y = L_2$ in a two dimensional Cartesian coordinate system. We first find the $y$-coordinates for the centers of disks. Assume that there are $M$ rows of disks we intend to place. The centers of disks in the first row which is the closest to $y = 0$ are placed on straight line $y = \frac{R}{2}$. This ensures that the bottom boundary $y = 0$ is fully covered as $ab \perp o_1o_2$ and $ab = \frac{R}{2}$. The centers of disks in the successive rows are placed $\frac{3R}{2}$ away, i.e., the $y$-coordinates of the centers of disks in the $i^{th}$ row is $(i-1)\frac{3R}{2} + \frac{R}{2}$, where $i = 1, 2, \ldots, M$. Let the centers of disks in the $m^{th}$ row be the closest to $y = L_2$ from beneath. In order to fully cover top boundary $y = L_2$, the number of rows $M$ should satisfy that

$$M = \begin{cases} m, & L_2 - ((m-1)\frac{3R}{2} + \frac{R}{2}) \leq \frac{R}{2} \\ m+1, & \text{otherwise} \end{cases} \tag{5.1}$$

Then we calculate the $x$-coordinates for the centers of disks. The distance between the centers of two neighbor disks in the same row is $\sqrt{3}R$. Assume that there are $N_{odd}$ disks in odd rows and $N_{even}$ disks in even rows. The $x$-coordinate of the center of the $j^{th}$ disk in odd rows is $\sqrt{3}(j-1)R$, where $j = 1, 2, \ldots, N_{odd}$, and the $x$-coordinate of the center of the $k^{th}$ disk in even rows is $\sqrt{3}(k-1)R + \frac{\sqrt{3}R}{2}$, where $k = 1, 2, \ldots, N_{even}$. Such placement ensures full coverage of left boundary $x = 0$, which can be proved as follows.

*Proof.* Denote the circle whose center locates at point A as *circle of A*. Let $d$ and $e$ be the two intersection points of circle of $o_1$ and circle of $o_4$. Extend $o_1d$ so that it intersects with $o_4h$ at $c$, where $o_4h$ is the radius of circle of $o_4$ and $o_4h \perp o_1c$. It is easy to see that $\angle do_1o_4 = \frac{\pi}{6}$ and $\angle o_1de = \frac{\pi}{3}$. Since $de \perp o_1o_4$, $\angle o_1o_4h = \frac{\pi}{3}$. We can see that $\angle ho_4o_5 = \pi$. Let $f$ and $g$ be the two intersection points of circle of $o_4$ and circle of $o_6$. We have $\angle o_4o_6f = \frac{\pi}{6}$ and $\angle o_6fg = \angle ho_4o_6 = \frac{\pi}{3}$. Extend $o_6f$ so that it intersects with $o_4h$ at $c'$. Since $o_4o_6 \perp fg$, $o_6c' \perp o_4h$. As $o_1o_4 = o_4o_6$ and $\angle o_1o_4h = ho_4o_6 = \frac{\pi}{3}$, $co_4 = c'o_4 = \frac{\sqrt{3}R}{2}$. Thus $c$ and $c'$ overlap with each other. We can see that $o_1$, $d$, $f$ and $o_6$ are on the same straight line. Since $o_1$ and $o_6$ are placed on $x = 0$, $d$ and $f$ are also on $x = 0$. This means that $o_1o_6$ which is a segment of left boundary $x = 0$ is covered by circle of $o_1$, circle of $o_4$ and circle of $o_6$. When $M$ is sufficiently large, the entire left boundary $x = 0$ can be fully covered. □

We now determine the numbers of circles in each row. Let the $N_1^{th}$ disk be the right-most disk in row $i$, and denote the $x$-coordinate of the center of this disk as $x_{N_1}$. Let the $N_2^{th}$ disk be the right-most disk in an adjacent row, and denote the $x$-coordinate of the center of this disk as $x_{N_2}$. In order to fully cover the right boundary of $\mathcal{S}$, $x_{N_1}$ and $x_{N_2}$ should satisfy that

$$|x_{N_1} - x_{N_2}| = \sqrt{3}R/2$$
$$\max\{x_{N_1}, x_{N_2}\} \geq L_1, \min\{x_{N_1}, x_{N_2}\} < L_1$$

Next, using the centers of disks as anchor points, we can determine the movement of the cluster heads. We first divide $\mathcal{S}$ into regions according to the number of cluster heads, so that each of the regions contains the same number of anchor points. To be more specific, each region has the same number of rows of anchor points, and each row has the same number of anchor points. In Fig. 5.3, e.g., $\mathcal{S}$ is covered by

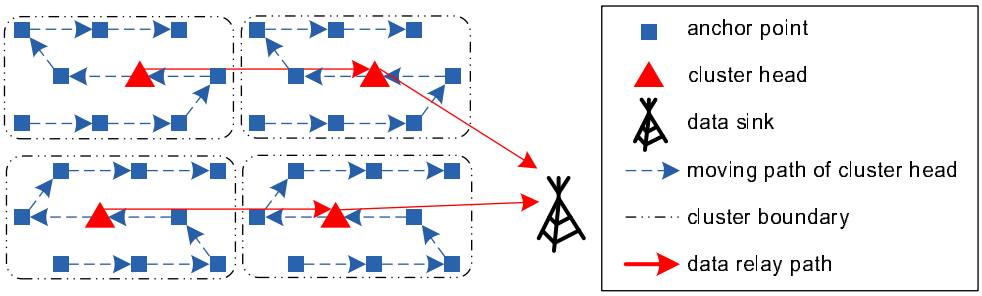



Figure 5.3: Division of regions and the moving paths of 4 cluster heads in $\mathcal{S}$. $\mathcal{S}$ is divided into 4 regions with 9 anchor points in each region, which consists of 3 rows and each row contains 3 anchor points. Each cluster head moves along the planned path and sojourns at each anchor point on the path for a period to gather data from its surrounding sensor nodes. The collected data are transmitted among the cluster heads to the data sink.

36 anchor points which are organized into 6 rows, and each row contains 6 anchor points. Since we have 4 cluster heads, we divide $\mathcal{S}$ into 4 regions, each of which consists of 3 rows and each row contains 3 anchor points.

The moving paths for the cluster heads are planned with the objective of maximizing the data collected from each sensor node in a data gathering period, which is calculated as

$$sum\_of\_data = (\tau - \frac{d}{v} - t_{rc})r \tag{5.2}$$

where $\tau$ is the duration of a data gathering period, $d$ is the moving distance of the cluster head in the data gathering period, i.e., the distance between the two anchor points selected in two successive data gathering periods, $v$ is the moving velocity of the cluster head, $t_{rc}$ is the time for routing and clustering of sensor nodes after the cluster heads arrive at the selected anchor points, and $r$ is the data rate of the sensor node. Clearly, the total amount of collected data increases when the moving distance is shorter, thus we plan zig-zag moving paths for cluster heads so that between two successive data gathering periods, each cluster head moves to an unvisited anchor point in the region which is $\sqrt{3}R$ away. After reaching the end of the path, the cluster head turns back and moves along the path in the reverse direction at the beginning of the next data gathering cycle.

In order to ensure that every sensor node can be a one-hop neighbor of a cluster head during a data gathering cycle, the cluster head should move fast enough to visit

all the anchor points in the network. The number of anchor points in a large network may be too large for cluster heads to visit all of them in such a period. There are two solutions for solving this problem. The first solution is to employ more cluster heads and to divide the area into more regions, so that the number of anchor points in each region is reduced. For the network shown in Fig. 5.3, each region contains 9 anchor points. In Fig. 5.4(a), the number of cluster heads increases to 9, so the network is divided into 9 regions, thus the number of anchor points in each region is reduced from 9 to 4. The second solution is that, instead of calculating the positions of the anchor points by using the method of no-gap covering, the anchor points are picked based on geographical symmetry. A region is divided into several *sub-regions*, and the center of the sub-region is selected as the anchor points of the cluster head in the region. Fig. 5.4(b) shows an example of such anchor point selection method. For the network in Fig. 5.3 which is divided into 4 regions, in Fig. 5.4(b), each region is further divided into 4 sub-regions, and the centers of the 4 sub-regions are chosen as the anchor points of the cluster head for this region. This method may lead to less effective energy balancing as the task of data forwarding is no longer shared by all the sensor nodes, and more energy consumption on the sensor nodes around anchor points due to that a large amount of data forwarding may cause energy holes in these areas. However, the advantage of this solution is that it has lower overhead for routing and clustering which is performed every time after cluster heads change their positions. Since the number of anchor points in each region is much smaller than that in the first solution above, routing and clustering will be calculated less frequently, thus the overhead is reduced. In Fig. 5.4(b), the number of anchor points in each region is 4, so the frequency of routing and clustering calculation is $\frac{4}{9}$ of that in Fig. 5.3 where each anchor point changes its position 9 times in a data gathering cycle.

### 5.4.2 Solar Irradiance Aware Mobility Control

In 5.4.1, we propose that a cluster head traverses all the anchor points along a Zig-Zag path, so that the total amount of collected data can be potentially maximized. Given such a moving path, another parameter that impact the moving pattern of a cluster head is the data gathering period, which is the time a cluster head sojourns at an anchor point for data collection. For fairness, the data gathering period for

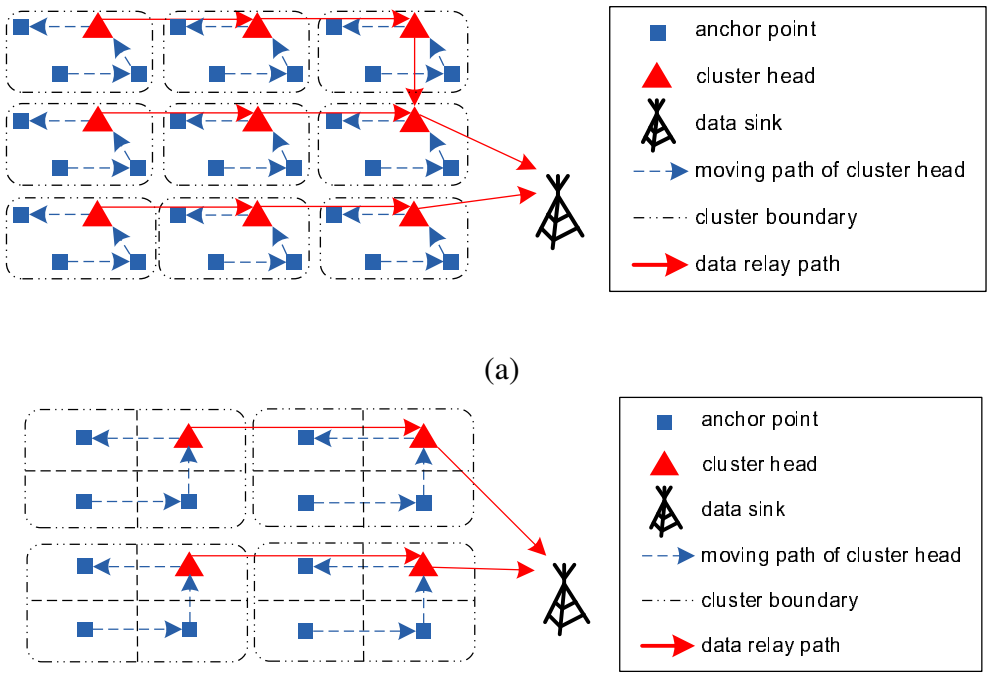



Figure 5.4: Reducing excessive movement for each cluster head caused by a large amount of anchor points in the region. (a) Divide the area into 9 regions and cluster head movement is reduced to 4 in a data gathering cycle. (b) Divide each regions into 4 sub-regions and select the centers of the sub-regions as the anchor point. Cluster head movement is reduced to 4 in a data gathering cycle.

each anchor point should be potentially the same, i.e., $\tau = \frac{T}{number\_of\_cluster\_heads}$. It is noticeable that when solar irradiance is high enough, the output of a solar panel could be larger than the rate of energy consumption caused by wireless communication, thus the energy levels of the nodes increase and the batteries could get fully recharged. In such a scenario, changing the position of of a cluster head may not benefit the balancing of energy consumption among the nodes, which is one of the major concern of MADG-SIA. Thus we propose a method to adaptively adjust data gathering period according to solar irradiance.

#### 5.4.2.1 Energy Model

In order to control the data gathering period at each anchor point to adapt to solar irradiance, it is necessary to accurately model the energy changes in sensor nodes. In general, the energy status of a node is determined by the energy consumption rate and energy replenishing rate of the node. For node $i$, the receiving rate equals the sum of the data rates of its descendants, while the transmission rate equals the receiving rate plus its own data rate, thus the energy consumption rate can be calculated as

$$\pi_i^d = e_r \sum_{j \in \mathcal{D}_i} r_j + e_t(r_i + \sum_{j \in \mathcal{D}_i} r_j) \tag{5.3}$$

where $\pi_i^d$ is the rate of battery discharging, $e$ is the energy consumed for transmitting a unit data, and $\mathcal{D}_i$ is the set of active descendants of node $i$. The recharging profile of solar harvesting sensor nodes is modeled as

$$\pi^r = I \cdot \eta_p \cdot \rho_e \cdot A \tag{5.4}$$

where $\pi^r$ is the rate of battery charging, $I$ represents the solar irradiance arrived at the solar panel, $\eta_p$ is the efficiency of the solar panel to convert solar irradiance to electrical power, $\rho_e$ is the electrical regulating and charging efficiency, and $A$ is the size of solar panel. The energy status of node $i$ after $\Delta t$ time can be updated as follows

$$b_i^{t+\Delta t} = \begin{cases} B & \check{b}_i^{t+\Delta t} > B \\ 0 & \check{b}_i^{t+\Delta t} < 0 \\ \check{b}_i^{t+\Delta t} & \text{otherwise} \end{cases} \tag{5.5}$$

$$\check{b}_i^{t+\Delta t} = b_i^t + \Delta t \cdot I \cdot \eta_p \cdot \rho_e \cdot A - \Delta t(e_r \sum_{j \in \mathcal{D}_i} r_j + e_t(r_i + \sum_{j \in \mathcal{D}_i} r_j))$$

where $b_i^{t+\Delta t}$ is the energy level of node $i$ at time $t + \Delta t$, $B$ is the capacity of the battery, and $\check{b}_i^{t+\Delta t}$ is the estimated energy level of node $i$ at time $t + \Delta t$. This energy model reveals that a battery can neither store energy more than its capacity nor provide energy more than the sum of its remaining energy and the energy harvested by the solar panel.

#### 5.4.2.2 Adaptive Mobility Control

According to the energy model established in 5.4.2.1, the battery of a node is recharged when the energy provided by the solar panel is higher than consumed for wireless communications, i.e., $\pi_i^d < \pi^r$. Due to the limitation of battery capacity, the energy of nodes whose batteries have been fully charged do not increase any more, while the energy of the nodes whose batteries are not full keeps increasing. As a result, the differences of energy levels among the nodes declines, and energy distribution tends to be uniform. Based on such fact, changing the position of cluster head when the nodes are being charged does not benefit balancing the energy distribution among the nodes. Furthermore, changing the position of cluster heads during such time reduces the number of nodes whose energy consumption can be balanced through the movement of cluster heads when solar irradiance is low ($\pi_i^d > \pi^r$). To address the impact of change of solar irradiance, we propose a mobility control method which is illustrated as follows.

1. A data gathering cycle starts when solar irradiance is available. A cluster head sojourns at each anchor point for $\tau = \frac{T}{number\_of\_cluster\_heads}$ before moving to the next anchor point[1]. This ensures only a few anchor points are visited before the solar irradiance becomes high enough to charge the nodes, thus the effort for balancing energy consumption by moving cluster heads in the regions can be improved.

2. Each node monitors its energy status change when uploading data to the cluster head. When the residual increases, the node on the edge of the cluster send

[1]It is noticeable that $T$ changes year round as sunrise and sunset times change. However, the change is slight between two successive days, thus $T$ can be easily adjusted to address the change.

a "Recharging" indicator to its next-hop neighbor. Each intermediate node, being aware of all the one-hop neighbor nodes for whom it relays data during cluster and routing operation, records the "RECHARGING" indicators from these neighbors. If all these neighbors send "RECHARGING" indicator, and its own battery is being recharged, the node sends a "RECHARGING" indicator to its next-hop neighbor. If a cluster head receives "RECHARGING" indicators from all its one-hop indicator, it enters "QUIESCENT" state, suspends future movement and collects data at current anchor point until it quits "QUIESCENT" state.

3. If a node detects decrease of residual energy and it has previously sent a "RECHARGING" indicator, it sends a "DISCHARGING" indicator to its next-hop neighbor. Upon receiving a "DISCHARGING" indicator, the node remove the "RECHARGING" indicator of the corresponding neighbor from its record, and send a "DISCHARGING" indicator to its next-hop neighbor. When the cluster head in "QUIESCENT" state receives such an indicator, it quits "QUIESCENT" state immediately. If the time that the cluster head has sojourned at current anchor point, denoted as $t_s$, is less than $\tau$, it waits at the current anchor point until $t_s = \tau$; otherwise it recalculate data gathering period for the rest of the tour, which is $\hat{\tau} = \frac{T - elasped\_time\_in\_current\_data\_gathering\_cycle}{number\_of\_unvisited\_cluster\_heads}$
.

### 5.4.3 Weighted Routing and Clustering Algorithm

In MADG-SIA, after the cluster heads have moved to anchor points, the next step is to organize the sensor nodes around cluster heads and find the routes for each node to send data to the cluster head. We present a distributed algorithm, named Weighted Clustering and Routing (WCR), for MADG-SIA sensor networks. The basic idea of WCR is to evaluate the cost of data transmission between a transmitter and a receiver by their energy levels. Data should always be sent along the route that has the lowest cost, and a sensor node should only send data to the cluster head to which the cost of the route is minimum. Next we describe WCR in detail.

With the energy levels obtained for all sensor nodes, we can draw the connections among sensor nodes and cluster heads. We represent these connections by a directed graph $G(V, E)$, where $V$ is the set of vertices denoting sensors and cluster

heads, and $E$ is the set of arcs denoting the connections between the ordered pairs of vertices. The weight of arc $(i, j)$ $w_{i,j}$, which represents the cost of data transmission over link $(i, j)$, is defined as the reciprocal of the minimum of the energy level of node $i$ and node $j$ if they are inside the transmission range of each other, i.e., $w_{i,j} = w_{j,i} = \frac{1}{\min\{b_i, b_j\}}$. The energy levels of cluster heads are considered as $+\infty$. If node $i$ and node $j$ are outside the transmission range of each other, or if one or both of them have no energy left, $w_{i,j}$ and $w_{j,i}$ are set to $+\infty$.

The weighted routing and clustering algorithm works as follows. At the beginning of the process, each cluster head initiates a ROUTE-REQUEST (RREQ) message, which includes the following domains: the address of the cluster head, route weight $w$ that is set to zero and a sequence number (SN) that is increased by one each time the cluster head moves to a new anchor point. In order to avoid flooding RREQ from each cluster head in the entire area $\mathcal{S}$, a domain containing time-to-live information (TTL) is also defined in RREQ.

Each node keeps a record of the maximal accumulated weight of the route to cluster head $k$. When node $i$ receives a RREQ message originated from cluster head $k$ from its neighbor node $j$, it calculates the accumulated route weight $w_{i-j-CH_k}$ by adding $w_{i,j}$ to $w$ in RREQ, and compares it with its record. If the new $w_{i-j-CH_k}$ is smaller than the record $w_{i-CH_k}$, node $i$ saves $w_{i-j-CH_k}$ as $w_{i-CH_k}$. When a route with smaller accumulated weight is found, the node also decreases $TTL$ in RREQ by one and broadcasts the message if $TTL$ is greater than one, otherwise, the message is discarded.

It is noticeable that such a procedure may cause a lot of message passing since multiple routes may exist between a node and the cluster head. A timed-updating mechanism is adopted to avoid this problem, in which the update message for a new route is not sent immediately, but after a period which is proportional to the accumulated route weight. This way, a route with larger weight could be discovered by a node before it updates its neighbors with smaller route weight. For example, when node $i$ finds route $a$ with efficiency being 50 which is the highest among the existing routes, it waits for 10s before updating its neighbors. Route $b$ with efficiency being 100 is found before the timer expires, then sending RREQ message which updates route $a$ information is avoided.

By flooding RREQ messages in the network, routing information is received by all sensor nodes. Clustering is also accomplished as each sensor node selects

the cluster head to which the route has the largest accumulated route weight. For example, when node $i$ receives RREQ message from two different cluster heads, $CH_a$ and $CH_b$, it compares the accumulated weights of the routes to the two cluster heads. If $w_{i-CH_a} < w_{i-CH_b}$, node $i$ considers itself to be in the cluster associated with $CH_a$. The WCR algorithm on node $i$ is described in Algorithm 2.

**Algorithm 2**: Weighted Routing and Clustering Algorithm for Node $i$. There are $K$ cluster heads deployed in $\mathcal{S}$.

**Input:** Battery status of node $i$.
**Output:** Next-hop neighbor $p$ on the path to the chosen cluster head.

Set $Timer_{proc}$;
**loop**
  Node $i$ waits for RREQ message or timer expiration;
  **if** RREQ is received from neighbor $j$ **then**
    $w_{i-j-CH_k} \leftarrow w_{j-CH_k} + w_{i,j}$;
    **if** $w_{i-j-CH_k} > w_{i-CH_k}$ **then**
      continue;
    **end if**
    Update local record $w_{i-CH_k}$ with $w_{i-j-CH_k}$;
    Save node $j$ as next-hop neighbor to cluster head $k$;
    **if** $TTL_{CH_k}$–==0 **then**
      continue;
    **end if**
    Update RREQ by setting domain $w_{i-CH_k}$ to $w_{i-j-CH_k}$;
    Set $Timer_k$;
    continue;
  **end if**
  **if** $Timer_{proc}$ expires **then**
    Find $\min\{w_{i-CH_k}, k = 1, 2, \ldots, K\}$;
    Choose the corresponding neighbor $j$ as the next-hop neighbor;
    break;
  **end if**
  **if** $Timer_k$ expires **then**
    Broadcast RREQ originated at cluster head $k$;
  **end if**
**end loop**

## 5.5 Performance Evaluation

In this section, we evaluate the effectiveness of MADG-SIA with different parameter setups through extensive simulations in MATLAB, and compare it with two static data gathering schemes of solar harvesting sensor network. One of them has have all the sensor nodes send data to the data sink directly, which is denoted as DSDG, while the other one has hierarchical structure with fixed cluster heads deployed and is denoted as CHDG. In the simulation, we generate a network consisting of 500 wireless sensor nodes randomly distributed in a $200m \times 180m$ area for demonstration purpose.

The cluster heads are placed at the bottom-left anchor point of the each region at the beginning of the simulation, and they move to other anchor points along the designed path at the beginning of each data gathering period. We assume that all the sensor nodes wait until a route to the cluster head is found before they start data transmission. In a data gathering period, each node stops sending data when the route selected for uploading is broken due to depletion of energy in some of their ancestors. We use real solar irradiation measurements collected by the Baseline Measurement System (Global 40-South PSP) at National Renewable Energy Laboratory in 2011. The default values of the parameters used in the simulations are listed in Table. 5.2. The simulation is run for 24 consecutive hours.

### 5.5.1 Performance of MADG-SIA

There are several critical parameters in MADG-SIA that affect the performance, including the number of cluster heads $N$ and the data rate of the sensor nodes $r$. Fig. 5.5 demonstrates the residual battery energy of sensor nodes after a data gathering cycle (24 hours, starting when solar irradiance is available) when the numbers

Table 5.2: Parameter Settings for performance evaluation of MADG-SIA.

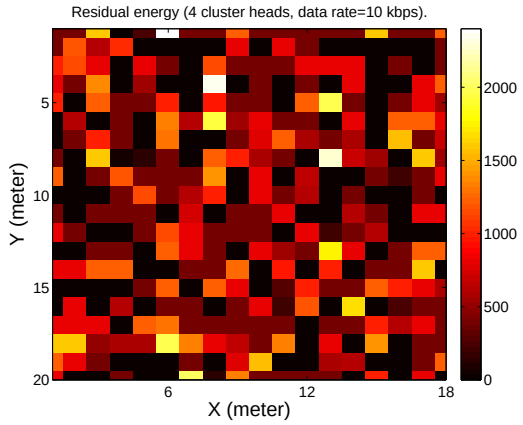
| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $e_r$ | 0.3mJ/Kbps | $e_t$ | 0.4mJ/Kbit |
| $N$ | 500 | $L_1 \times L_2$ | $200m \times 180m$ |
| $b$ | 1mAh | $r$ | 10Kbps |
| $\eta_p \cdot \rho_e$ | 0.06 | $A$ | $10cm \times 5cm$ |

of cluster heads are 4, 9 and 16, respectively. From the figures, we can see that the scenario with less cluster heads result in more balanced residual battery energy. This is reasonable since with fewer available cluster heads, there are more anchor points in each region, thus each cluster head has to move more frequently to go through all these anchor points in a data gathering cycle. As a result, the cluster heads sojourn for shorter time at each anchor point and less data is forwarded by each node which consumes less energy. Fig. 5.6 exhibits the amount of collected data in each hour when different number of cluster heads are. During the daytime when solar irradiance is strong, the output of the solar panels is sufficient to support transmission of data from most of the sensor nodes, thus all the three scenarios have similar data output. When solar irradiance becomes unavailable, the sensor nodes in the scenarios with more cluster heads have more aggregation points, but shorter transmission time due to limited energy supply from batteries. Thus all the three scenarios collect similar amount of data from the network.

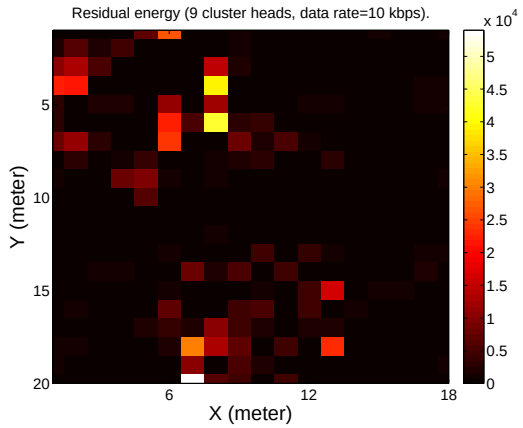
Fig. 5.7 presents residual battery energy of the sensor nodes after a data gathering cycle in the scenarios where the sensor nodes have different data rates. A higher data rate results in heavier traffic going through sensor nodes, which consumes energy much faster on the sensor nodes around the cluster heads than the sensor nodes in the scenario with lower data rate. After energy holes are formed around the cluster heads, all the related nodes disconnected from the cluster heads and stop consuming energy. Fig. 5.8 compares the amount of data collected from the network in a data gathering cycle when the data rate of sensor nodes increases. It can be observed that when solar irradiance is strong enough during the daytime, the network with higher data rate yields more data output. However, when solar irradiance is no longer available, the data output of such network drops rapidly and less data are collected.

### 5.5.2 Comparison with Static Solar Harvesting Networks

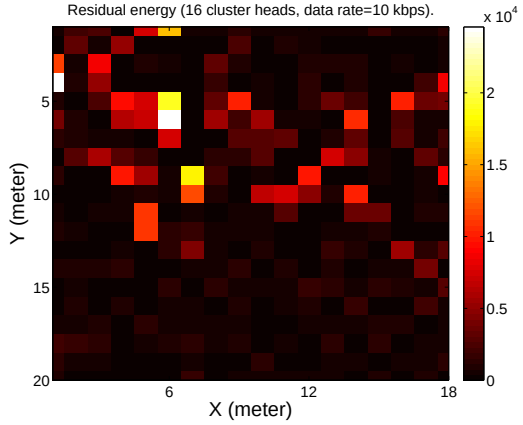
In this subsection, we compare MADG-SIA with DSDG and CHDG in solar harvesting sensor system. Using the Cartesian coordinate system shown in Fig. 5.2, DSDG has a data sink placed at $L_1, \frac{L_2}{2})$, and CHDG has four cluster heads placed at $(\frac{L_1}{4}, \frac{L_2}{4})$, $(\frac{3L_1}{4}, \frac{L_2}{4})$, $(\frac{L_1}{4}, \frac{3L_2}{4})$ and $(\frac{3L_1}{4}, \frac{3L_2}{4})$ separately. Fig. 5.9(a) and (b) show the residual battery energy at the end of a data gathering cycle (residual battery en-




(a)




(b)




(c)

Figure 5.5: Comparison of residual battery energy at the end of a data gathering cycle with different number of cluster heads. Fewer cluster heads result in more balanced energy distribution. (a) Standard deviation of residual energy $\sigma$=$0.1714J$; (b) $\sigma$=$2.1041J$; (c) $\sigma$=$2.7910J$.

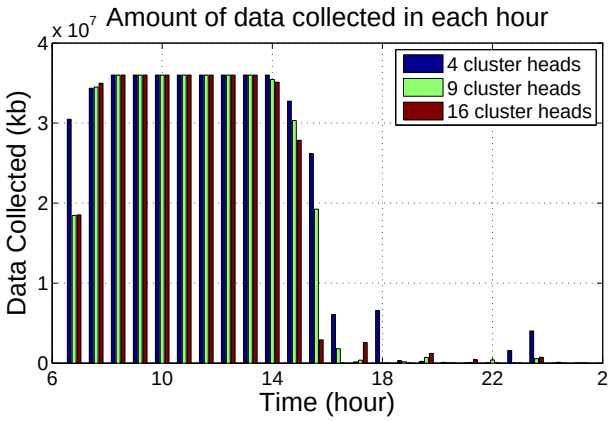
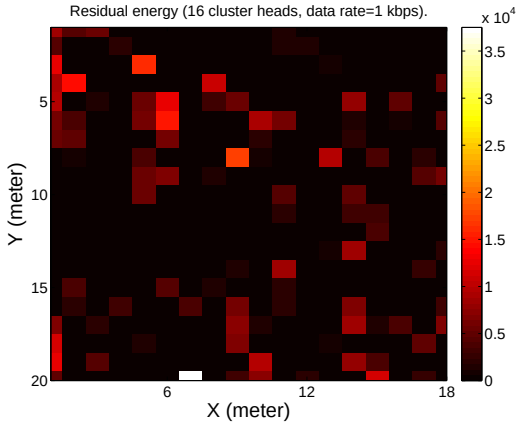


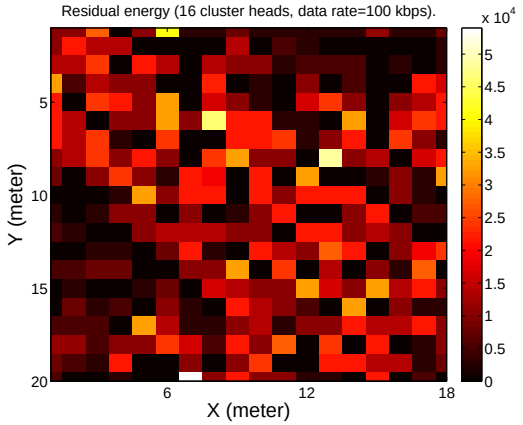

Figure 5.6: Comparison of accumulated collected data with different number of cluster heads.

ergy of MADG-SIA is shown in Fig. 5.5). Energy hole is observed in DSDG as the sensor nodes around the data sink have no energy left, while the energy levels of the sensor nodes in other positions are relatively high. The same thing happens to CHDG as the sensor nodes around the cluster heads have depleted their energy while others still alive. In comparison, with the assistance of mobile cluster heads, energy consumption in MADG-SIA is more balanced in the entire network. Fig. 5.10 compares MADG-SIA with DSDG and CHDG-fixed in consecutive 24 hours in terms of the amount of gathered data. Although they yield similar amount of data output when there is strong solar irradiance during the daytime, the output of DSDG and CHDG drops more rapidly than MADG-SIA when solar irradiance gets weakened. Once the data output drops to zero, DSDG and CHDG no longer have data output since the sensor nodes around the data sink or the cluster heads have depleted their energy, while MADG-SIA is still able to collect data from some nodes that have energy left as the cluster heads move to different anchor points.

The simulation results in Fig. 5.5-5.8 provide some insights for system parameter selections. For the input settings used in the simulation (Tab. 5.2), a network with 4 cluster heads outperforms the networks with 9 or 16 cluster heads in terms of residual energy distribution and the amount of collected data, and a network in which sensors send data at $10Kbps$ outperforms a network with the data rate being $100Kbps$ with respect to these two metrics. When designing such a network for a specific application, however, these inputs may change (e.g., the size of the




(a)




(b)

Figure 5.7: Comparison of energy distribution at the end of a data gathering cycle with different data rate. Lower data rate results in more balanced energy distribution. (a) $\sigma$=2.2034$J$; (b) $\sigma$=3.9691$J$.

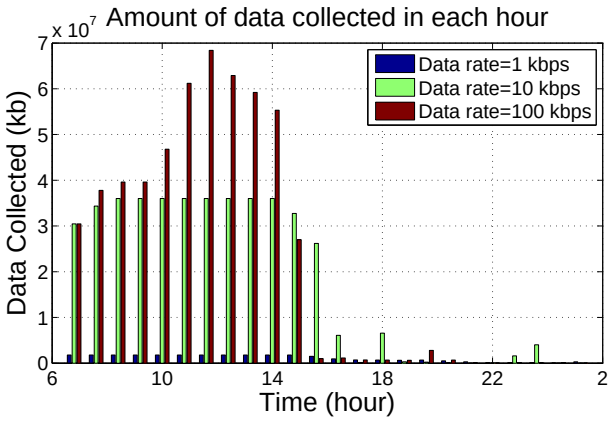
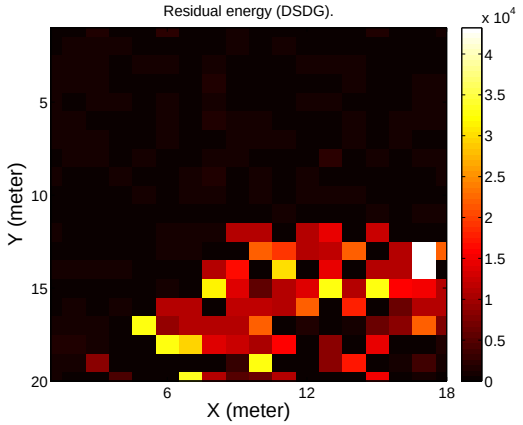



Figure 5.8: Comparison of accumulated collected data with different data rate.

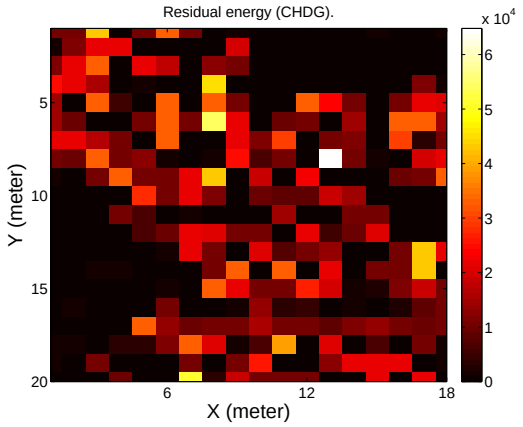
network, number of sensors and transmission range), thus the optimal value of the number of cluster heads and data rate may vary. Evaluation with inputs precisely modeling all the related physical factors should be performed to derive the optimal values of the parameters. The simulation results in Fig. 5.9-5.10 demonstrate that MADG-SIA is less sensitive to solar irradiance variance in solar harvesting WSNs. Compared with data gathering schemes with a single data sink or fixed cluster heads in which energy holes are observed around the data sink or cluster heads when solar irradiance turns to be low, MADG-SIA has more balanced energy distribution over the whole network, which consequently improves data output and extends network operating time.

## 5.6 Conclusions

Balancing energy consumption is a challenging task in designing a wireless sensor network as data collection is usually restricted by the sensor nodes that have depleted their energy, even for sensor nodes with energy replenishment. In this chapter, we have proposed a novel mobility assisted data gathering mechanism (MADG-SIA) which balances the energy consumption for data transmission among the sensor nodes to extend the operating time of a heterogeneous energy replenishable wireless sensor network. The area is divided into several regions, and a cluster head moves around in each region, sojourns at different anchor points to gather data from surrounding sensor nodes in a pattern with respect to solar irradiance.




(a)




(b)

Figure 5.9: Comparison of energy distribution at the end of a data gathering cycle with different data gathering schemes. (a) Data gathering with four static cluster heads (DSDG). $\sigma$=3.9527$J$. (b) Data gathering with a static data sink (CHDG). $\sigma$=4.7754$J$.

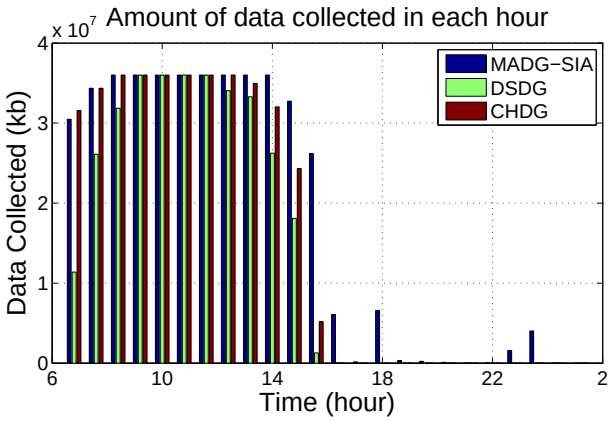



Figure 5.10: Comparison of accumulated collected data with different data gathering schemes.

By employing such mobile cluster heads, the energy distribution in the solar harvesting wireless sensor network is balanced, thus data output is increased and the network operating time is prolonged. We present a low-complexity method to find moving patterns for each cluster head. We also present a weighted algorithm for node clustering and routing. We demonstrate that MADG-SIA is efficient in prolonging network operating time and enhancing data output. We provide guidance on parameter selection for designing and deploying such WSNs by evaluating the performance of MADG-SIA with different parameter setups, and we also demonstrate the effectiveness of MADG-SIA in extending network operating time and increasing data output by comparing with the scenarios of data gathering with single data sink (DSDG) and data gathering with statically deployed cluster heads (CHDG-Fixed) in energy harvesting WSNs.

# Chapter 6

# A Versatile Platform for Mobile Data Gathering Experiments in Wireless Sensor Networks

In recent years, mobile data gathering in wireless sensor networks has attracted much interests in the research community. However, despite extensive efforts, many of previous work in this area lies only in theory and evaluates network performance with computer simulations, which leaves a large gap from reality. In this chapter, we present the design and implementation of a general purpose, flexible platform for mobile data gathering in wireless sensor networks to evaluate network performance and algorithms in a practical setting. Instead of relying on hand-crafted theoretical models, our platform integrates both mobile data collector and sensor nodes to provide realistic performance evaluations. In addition, the platform adopts a modular design in mobile data collector and sensor nodes, and equips the mobile data collector with advanced computing capability, which makes it versatile for evaluating the performance of a wide-range of applications. We expect that this platform can become a very powerful general tool for more accurate network simulations and facilitate performance optimization in wireless sensor networks.

## 6.1 Introduction

Recent advance in technology has fueled a renewed interest of intelligent designs in low-power mobile devices such as smart phones, sensors, wearable devices and Internet-of-Things. As a bridge between the physical world and cyber space, sensors play an irreplaceable role to detect and classify real world objects. For years, wireless sensor networks (WSNs) have been widely used for automation in many fields, such as object detection for battlefield surveillance, micro-climate monitoring for indoor/outdoor control, and soil moisture sensing for automatic irrigation.

Despite of the wide deployment of wireless sensor networks, energy remains a major concern in network design and operation. Energy holes, formed due to the unbalanced energy consumption for data transmission, leads to network disruption when nodes around the sink deplete their energy. In addition to the energy constraint, in sparse networks (e.g., airborne sensors), end-to-end connections cannot be guaranteed so sensed data from these disconnected regions may be lost.

To tackle these problems, mobile data gathering has been proposed, for example, [34, 39, 52, 119, 120, 122–125]. In mobile data gathering, a number of mobile data collectors are employed to collect data from sensors, which can move sufficiently close to sensors and greatly shorten data relay paths, thereby significantly alleviate the energy hole problem. Mobile data collectors also enjoy the freedom to move into disconnected regions for data collection.

Upon discovery of the benefits of adopting mobile data gathering, many algorithms and network protocols have been developed. In principle, most of these algorithms and protocols share the similarity that they introduce sophisticated designs on mobile data collectors, so we will discuss some example works [119, 120, 126] next. Ammari et al. [119] developed an energy-aware protocol for disseminating data to the mobile sink in WSNs using an information theoretic approach. Zhao et al. [120] applied Space-Division Multiple-Access (SDMA) technique by launching multiple antennas on an MDC such that distinct compatible sensors can make concurrent data uploading to reduce latency. Xu et al. [126] addressed the event collection problem by leveraging sink mobility and the spatial-temporal correlation of events. Note that all the above works were only proposed theoretically and evaluated by simulations. They have not been validated by real implementations.

Although mathematical tools and software simulators are straightforward, they

are generally "inaccurate" given the vast majority of simplifications in mathematical models and imperfection to characterize real world dynamics in software simulators. They often overlook the impact from multiple practical factors so the results may deviate significantly in reality. To this end, several testbeds have been built for more accurate evaluations [131, 132, 137]. Nevertheless, many of such testbeds were designed for a particular project in an application specific domain, and it may be infeasible to use the testbeds for other applications.

Realizing these problems, in this chapter, we present a versatile platform that enables performance validation for various MDG algorithms and network protocols in WSNs. We focus on architectural support for high performance computing and customizability on MDC and use high-performance Field Programmable Gate Array (FPGA) for MDC to handle mobile computing tasks. Our design also considers wireless communication, memory management, localization, time synchronization and mobility control. As another part of our platform, we enable low-power designs on sensors and enhance their computing capability, memory cache and resource expansions. We provide a complete experimental platform where customization can be achieved in different layers (application, network, MAC, or physical layer), and different MDG algorithms can be executed efficiently.

To validate the platform design, we also present a case study of wildlife detection on our campus. We implement the entire network on our platform. It includes sensor nodes with ultrasonic sensing for object detection and low-power wireless communications, and MDC with autonomous navigation/steering, data transmission and high-performance computing capabilities. During the operation, sensors are organized into clusters and MDC traverses through the cluster head (called *anchor*) locations for data gathering. Our experimental results reveal that beyond the theoretical analysis and software simulation, many unnoticeable, real factors also have great impacts on network performance. Based on such findings, we then propose solutions to account for these real factors to further improve the algorithms. Since experimental results are much closer to real applications, we believe that the platform will be a very useful tool for researchers/administrators to develop, deploy and manage WSNs utilizing MDG solutions.

The contribution of this chapter can be summarized as follows. First, we design a versatile mobile platform that can be used for various MDG tasks in WSNs. To the best of our knowledge, this is the first general mobile platform that can be

customized for different applications. Second, we adopt modular designs of components to allow future developments and expansions. Third, we validate the designs and implementations of the platform by an application of wildlife detection. Our results reveal possible estimation errors between the theoretical analysis and real implementations (e.g., 15% error in data latency). We further provide analysis and solutions to minimize these inaccuracies.

The rest of the chapter is organized as follows. Section 6.2 discusses previous theoretical works for MDG in WSNs. Section 6.3 gives the detailed description of the platform architecture and the implementation.

## 6.2 Related Work

Software simulation is the dominant method for performance evaluation of WSNs. NS-2 [127] was developed as a general network simulator for both wired and wireless networks, and has been adopted for WSNs as well. OMNET++ [128] is a discrete simulation library and framework primarily for building network simulators. Domain-specific functionalities such as support for sensor networks, wireless ad-hoc networks, Internet protocols, performance modeling, phonic networks, etc., are provided by frameworks. TOSSIM is a bit-level discrete event simulator developed to study the behavior of TinyOS/MICA nodes by modeling both software and hardware, including radio modules. SENS [129] was developed specifically for WSNs. The modular and layered architecture enables customization of applications, network communication and physical environment. ATEMU [130] enables instruction-level emulation for AVR processors which have been used in several wireless sensor prototypes. It provides libraries of heterogeneous hardware for simulating the entire hardware platform. In general, network simulators, such as NS-2, focus on communications whereas neglect the important aspects of node operations. On the contrary, node level simulator/emulators, such as ATEMU, can provide high-fidelity abstraction of node behavior but require high processing ability and usually have poor scalability. Moreover, such software simulators/emulators do not offer native support for mobile data collectors.

There are also testbeds designed for wireless networks with mobility support. Mobile Emulab [131] employs robots that carry motes and single-board computers traversing through a fixed indoor area for MDG. It uses marks and cameras

installed on the ceiling to position the robot and guide its movement in the network. SensLAB [132] tracks animals by attaching sensors to them, receiving signals by static anchors, and forwarding data to the sink. MiNT [134], a miniaturized 802.11b-based, multi-hop wireless network testbed, serves as a platform for evaluating mobile wireless network protocols and their implementations. It reduces the physical space requirement for a wireless testbed while providing the fidelity of experimenting on a large-scale-testbed. The ORBIT testbed [135] consists of an indoor radio grid emulator for controlled experimentation and an outdoor field trial network for end-user evaluations in real-world settings. Kansei [136] is a sensing testbed where nodes are placed in a $15 \times 14$ rectangular grid, and robotic mobile nodes operate on top of a transparent Plexiglas plane mounted over the stationary nodes. In [137], an Unmanned Aerial Vehicle (UAV) is employed to collect data from ground nodes. The UAV transmits plane telemetry and receives orders for the autopilot or commands for flight control.

Although these testbeds can provide higher fidelity in performance evaluation in WSNs compared to software simulation, they are usually built for a specific application instead of more general, multi-functional tasks. For example, the robotic mobile nodes in [136] may not be suitable for outdoor experiments due to the limited off-road capability and availability of guidance. The UAV in [137] may be unable to receive data from nodes on the ground when wireless communication ranges last for few meters. Such characteristics may discourage the deployment of these testbeds in applications facing different requirements and restrictions, which motivates us to design and implement a versatile platform for MDG in WSNs.

## 6.3 Platform Architecture and Implementation

In this section, we provide the design and implementation of our platform as depicted in Fig. 6.1. In the platform, the sensor nodes collect data from the physical environment; the MDC calculates the trajectory and travels in the network to collect data from the surrounding sensors. Both sensors and MDC have computing, storage and communication capabilities (at different levels), and the modular structures enable easy extension to facilitate various application demands. The implementation of two devices are described in the next two subsections, respectively.

Figure 6.1: A photo of the platform showing 4 wireless sensor nodes and an MDC built with a high performance computing subsystem mounted on a vehicle.

### 6.3.1 Mobile Data Collector

The objective of MDC design is to provide a high performance processing unit with wireless communication capability and mobility support, so that it is able to move under control or autonomously in the network and communicate with sensor nodes for data collection. MDC consists of functional blocks to perform computation, wireless communication, and autonomous/controlled movement. The computing unit is designed based on a heterogeneous architecture and consists of a PowerQUICC III processor (Freescale MPC8548E) and a Virtex-6 FPGA (Xilinx XC6VSX475T). The interchangeable wireless communication unit provides interconnection between MDC and sensor nodes. The mobility control unit calculates the trajectory and guides MDC moving in the network for data collection.

#### 6.3.1.1 High Performance Computing Subsystem (HPCS)

The mobile data gathering task consists of several computing functionality, such as baseband signal processing for wireless communication with SDMA technology, high level protocol processing and optimal trajectory planning. Furthermore,

it is beneficial to execute computing tasks on MDCs whenever possible since sensor nodes usually have low computing capability and constrained energy supply due to cost/size limit. As an example, optimal routing and data flow control can be achieved using distributed algorithms by exchanging information among sensors, however, this may require several rounds of message exchange in the network, which could consume a lot of energy. By migrating the computation to MDC, sensor nodes can reduce energy cost thus the network lifetime can be prolonged. Such migration is important since sensing tasks usually require real-time processing and high computing capability in many applications.

The processing unit of MDC is based on a heterogeneous computing architecture. It takes advantages of 1) high speed, parallel processing of the hardware, 2) flexibility of the software and 3) reconfigurability of the digital logic of the hardware system. Traditional high performance computation relies on CPUs, in which algorithms and network protocols are executed with the support of underlying operating systems. The implementations on CPUs are quite flexible. For example, tasks can be described at high-level using platform-independent programming languages, such as C/C++. Such high level description is then translated into low level instructions and executed in serial by CPUs. The computation is done by the Arithmetic Logic Unit (ALU) and the computation speed is determined by CPU clock frequencies (from 50 KHz in ENIAC to GHz). In fact, for MDC, multiple computation-heavy tasks might need concurrent CPU executions so software implementation may not be fast enough to meet the requirements of many real-time processing tasks.

A solution to this problem is to allow parallel computation. By Amdahl's law [138], the potential speedup is a monotonically increasing function of the degree of parallelism. The invention of superscalar processors for instruction-level parallelism and multi-core processors for task parallelism enables parallel processing to some extent. However, since CPUs have a limited number and types of ALUs, parallelism is constrained. In contrast, FPGAs have a large number of logic units which can be combined for more computing units to work simultaneously so parallelism can be improved significantly. Furthermore, unlike CPUs which rely only on ALUs for all computations, FPGAs can utilize logic elements, such as LookUp Tables (LUTs), flip-flops and RAMs, to make computation more efficient. These logic elements can be customized to build functional blocks, such as matrix mul-

tiplication and digital filters for signal processing. Due to these benefits, FPGAs have found their applications in base stations for wireless telecommunications, unmanned combat air vehicles, space probes, and other areas.

We now describe the designs of our FPGA-based HPCS. The functional blocks of our system are shown in Fig. 6.2, which consists of a Virtex-6 FPGA (Xilinx XC6VSX475T), a PowerQUICC III processor (Freescale MPC8548E), and peripheral devices. XC6VSX475T is optimized for applications that require ultra high-performance DSP and serial connectivity with low-power GTX 6.5 Gbps serial transceivers (even capable of handling aerospace and defense applications). The FPGA subsystem acts as the main processing module where most computation-heavy tasks are executed. MPC8548E was designed to deliver high-speed communications processing performance and high-speed connectivity required by enterprise networking, fast telecom switching, storage and high-end imaging markets. It is based on a scalable e500 system-on-chip platform and Power Architecture processor core (3065 MIPS at 1333 MHz), designated for high level protocol processing in MDC. For brevity, we refer to "MPC8548E" as "PowerPC" and "XC6VSX475T" as "FPGA" in the following.

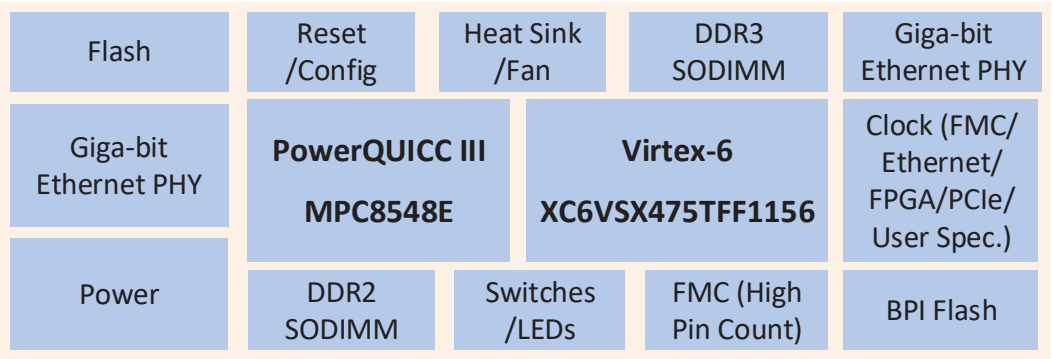



Figure 6.2: Functional blocks of HPCS. It consists of an FPGA, a PowerPC and peripheral devices.

The functionality and computing capability of HPCS can be easily extended. HPCS provides two industrial standard high-pin count FPGA Mezzanine Card (FMC) for function extensibility. Each FMC interface provides 10 Giga-bit serial transceiver pairs, 160 single ended, or 80 differential user-defined signals, and clocks. These two powerful interfaces can be attached to HPCS to implement various function-

alities. For example, using FMC-to-FMC cables, multiple HPCS boards can be connected through the high-speed interfaces to extend computing capability.

#### 6.3.1.2 Wireless Communication

Wireless communication functionality can be added to MDC by connecting R-F transceivers through FMC interface. The functionality depends on the selected transceivers related to applications. For experiments that do not involve lower layer processing, the transceiver can be a system-on-chip. It integrates both MAC and physical layer implementations, such as TI CC2530 which combines RF transceiver and 8051 MCU for 802.15.4/RF4CE/Zigbee applications. If advanced wireless communication techniques such as SDMA need to be implemented, an RF front end is preferred. It performs RF signal conditioning and data conversion (Maxim MAX2829), and provides flexibility to support the processing from all layers including baseband signal processing in physical layer. We currently use CC2400-based transceiver as the RF interface, and implement CC2400 controller, MAC protocol and routing algorithms. The CC2400 controller is designed as a Finite State Machine (FSM) on the FPGA, which controls status change by writing configuration data into the registers of CC2400, data transmission and reception through an SPI interface.

The structure of the wireless communication module is illustrated in Fig. 6.3. The protocol handler abstracts data from the incoming packet or encapsulates user data to be sent. It exchanges packets with the SPI feeder via a receiving buffer and a transmitting buffer. The status of the transceiver is controlled by the RF controller, which sets CC2400 in a proper state for idling, transmitting or receiving through an FSM. It reads data from the configuration ROM and writes them to CC2400 registers. The SPI feeder acts as the interface to the SPI driver which sends a byte for each transaction. It is designed to read/write data with arbitrary length so that the interfaces on other functional blocks towards the transceiver can be simplified.

#### 6.3.1.3 Mobility Control

MDC may move autonomously, or under control by receiving instructions from the command center through long range wireless communication. Since controlled mobility via instructions is relatively simple, here we focus on autonomous movement

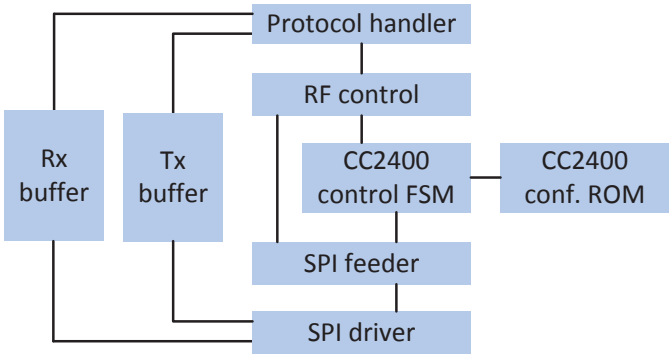



Figure 6.3: Structure of wireless communication module.

supported by the mobility control module. Mobility control module calculates the trajectory of MDC and directs its movements in the sensing field. It has three functional blocks: trajectory calculation, positioning, and vehicle control. When the locations of *anchors* are available, MDC calculates the trajectory based on certain criteria, such as finding the route of the shortest length. Such calculation is performed on HPCS and does not require additional hardware support. We will give an example for this implementation in Section 7.

For MDC to travel along the planned route, it is necessary to obtain its real-time position first (usually provided by a navigation system). In general, a navigation system can be a GPS receiver, an Inertial Navigation System (INS), or a combination of both. In our system, a GPS receiver Antenova M10214-A1 and an Inertial Measurement Unit (IMU) InvenSense MPU-9250 are incorporated in the mobility control unit and connected to HPCS via FMC connection. The GPS receiver provides real-time location information with an accuracy of 2.5 meters which is encapsulated using NMEA protocol and available through a UART interface. The IMU provides information of altitude estimation measured by the integrated 3-axis gyroscope, accelerometer and digital compass. The IMU also integrates a Digital Motion Processor for data fusion and outputs the attitude directly. However, the precision of such output may not be enough for some demanding applications as the output is updated every 5 ms. It will be necessary for such applications to read the raw IMU data and implement the data fusion processor in the FPGA or the PowerPC.

Given the locations and altitude of the starting point and destination, the mo-

bility control unit outputs commands to the mobile platform. The mobile platform used for our experiment is modified from a remote control vehicle (as shown in Fig. 6.1). The control signals of the vehicle consist of steering and motor control, which can be done by generating Pulse Width Modulation (PWM) signals in HPCS.

### 6.3.2 Wireless Sensor Node

The goal of the sensor node design is to provide a highly customizable platform for experimental purpose. The heterogeneous architecture described in Section 6.3.1 is also adopted in the sensor node. The node is constructed based on a microcontroller (MSP430F5438) that conducts control, computation and communication. A lightweight event-driven software platform is developed for the MCU to execute scheduling and data exchange among tasks. To reduce energy consumption, most of the chips in our platform (MCU, FPGA and RF transceiver) can be put into sleep mode. In the following, we use the term "MCU" interchangeably with "MSP430F5438" and "FPGA" interchangeably with "AGL125V2."

#### 6.3.2.1 Hardware Architecture

The sensor node consists of six major modules: main controller and processor, co-processor, wireless transceiver, positioning system, external memory, and I/O expansion. The hardware structure of the sensor node is illustrated in Fig. 6.4.

The main controller and processor, MSP430F5438, is a 16-bit ultra-low-power microcontroller with 256KB Flash and 16KB RAM. It can operate at low frequency (e.g., 4 MHz) to reduce power consumption, or high frequency (e.g., 32 MHz) to enhance computing performance. A Flash-based FPGA, Microsemi AGL125, is connected to the microcontroller through a customized parallel interface. The FPGA works as a co-processor to offload computation-heavy tasks from the MCU, such as sensory data encryption using AES [139]. By exploring parallelism between algorithms, the execution speed can be increased. The enhanced computing capability provided by the microcontroller and the FPGA implements more complex and powerful algorithms for better performance. Extra memory space is available on sensor node with the presence of an 8-Mbit SRAM. This is necessary for some applications when a large amount of data needs to be stored, e.g., for data caching in content-centric networks. The wireless transceiver module consists of a TI CC2400

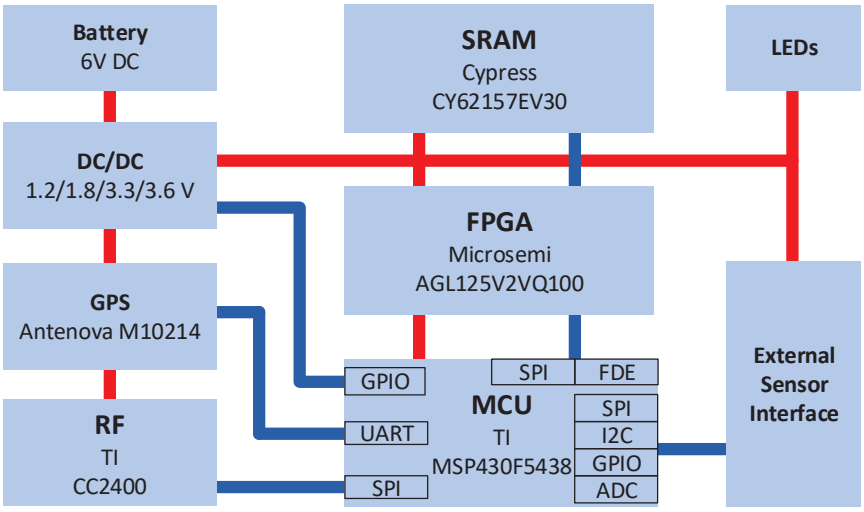



Figure 6.4: Hardware structure of a wireless sensor.

which is able to transmit data at different rates (10 Kbps/250 Kbps/1 Mbps) to adapt both low rate applications (e.g., indoor temperature monitoring in warehouses) and high rate applications (e.g., image/audio sensing). It is connected to the microcontroller through a connector, it can be easily replaced with other transceivers to satisfy the requirements of particular applications. Some sensor nodes may also contain a GPS module for positioning purpose. These nodes are called *anchors*. They can disseminate reference information to other sensors for local positioning by measuring the signal strength from surrounding anchors. Connections to different sensors (e.g., humidity, temperature) are enabled through an expansion connector. It offers both digital connections via SPI, UART, $I^2C$, or GPIOs, and analog connections which are connected to its 12-bit analog-to-digital converters (ADCs) to facilitate various interface standards used by sensors. All these components work with low supply voltage and small current when they are active. By supporting one or more stand-by modes, extra energy saving can be achieved. A well-designed power supply cycling policy makes components alternate among different power modes, thus energy consumption can be reduced significantly to extend battery lifetime.

#### 6.3.2.2 FDE: Customized Interface between MCU and FPGA

Since the FPGA is used as a co-processor and SRAM manager, it is necessary to exchange data between the MCU and the FPGA. Next, we introduce the definition of the FPGA Data Express interface (FDE), which is designed for communication between the MCU and the FPGA with low communication overhead. All FDE transactions are performed on its 4 signals: an MCU-to-FPGA indicator (MFI), an FPGA-to-MCU indicator (FMI), a bus grant (BUS_GNT), and a bidirectional data bus. To avoid the conflict of simultaneous writing to the data bus, the MCU acts as the bus arbiter that authorizes bus access by asserting BUS_GNT. BUS_GNT is low when the bus is idling, giving default bus access to the MCU. It is set to high if the FPGA is authorized to write to the data bus.

The data bus is high-'Z' when idling. When the MCU is writing to the data bus, it sets MFI to '1', keeps BUS_GNT at '0', and puts CMD (operation to be performed) on the data bus. For each successive data bytes, the MCU toggles MFI. The FPGA acknowledges with a falling edge on FMI, which notifies the MCU through an interrupt. The transaction ends after all the data has been received by the FPGA. Then the MCU releases the data bus.

When the FPGA wants to send data on FDE, it first requests for bus access by generating a falling edge on FMI. The MCU acknowledges by setting BUS_GNT to '1'. The FPGA writes CMD and other data to the data bus, and generates a falling edge on FMI for each byte. The MCU acknowledges the reception of the data by toggling MFI. After all the data has been received by the MCU, the FPGA releases the data bus.

#### 6.3.2.3 Software Platform

The software platform running on the MCU is implemented to ease the access to hardware and peripherals and support applications. It is programmed using C to ensure better compatibility and easiness of code migration from other platforms. The structure of the software system is illustrated in Fig. 6.5. The Hardware Abstraction Layer (HSL) implements the drivers of the hardware resources on the MCU, such as sending a byte over the SPI interface on a specific Universal Serial Communication Interface (USCI), setting a timer, and reading a block of data from the FDE interface. The light-weight Operating System Abstraction Layer (OSAL) is designed

to facilitate event-driven operation of sensors. OSAL implements handlers of peripherals (e.g., sending a packet to RF transceiver), interrupt service (e.g., interrupt from a timer), message queues and task scheduler. User applications create *tasks* to perform different functionalities. Message queues are used for data exchange between the tasks and the software platform.

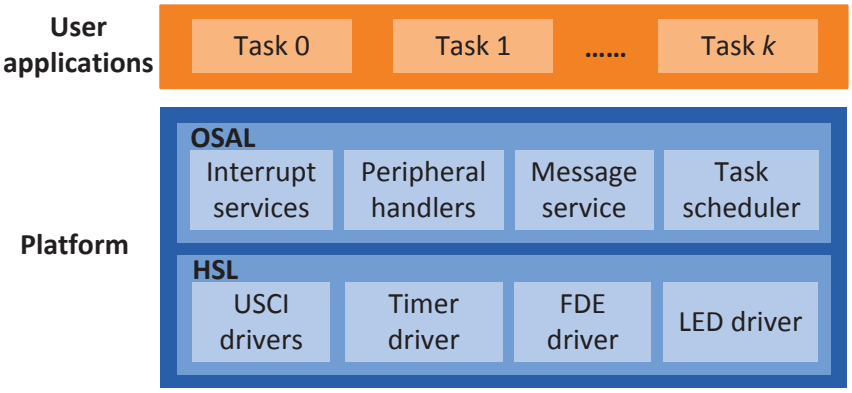



Figure 6.5: Software system structure of a wireless sensor node.

**6.3.2.3.1 Inter-Task Communications** Tasks and interrupt service routines (ISRs) exchange data via message queues. When a task is initialized, a message queue with customized size and depth is created by calling *creat_msg_queue()*. If task A or an ISR is trying to send data to task B, it writes the data to the message queue of task B. When the message queue of a task is not empty, the task can be activated for execution. Sending a message to a task and reading a message from a message queue can be implemented by calling function *send_msg()* and *read_msg()*, respectively.

**6.3.2.3.2 Event-Driven Operation and Task Scheduling** The operation of sensors is driven by events (e.g., arriving of an RF packet, timer expiration for monitoring the environment), thus the software system should also be modeled by event-driven designs. The events are represented in the MCU by interrupts through interrupt service routines (ISRs). The implementation of event processing depends on the evolved complexity. For light weight computation, the ISRs process the event directly and pass the result to the corresponding task by sending a message to it; for

heavy computations, the ISRs send a message to the task and leave the processing to it.

The task scheduler executes tasks in a priority-based, non-preemptive fashion. After the execution of the current task is accomplished, the task scheduler checks the lengths of all message queues and activates the task with the highest priority that has an unread message in its queue.

**6.3.2.3.3 Timing Service** A sensor may have several simultaneous timing requirements during operation. For example, for a timeout-resend mechanism of data transmission, a node resends a packet if no acknowledge is received in a time window; in a duty-cycled WSN, sensors wake up periodically to monitor the ambient environment. Such functionalities utilize the timers in MCUs. Given the fact that all MCUs have a limited number of hardware timers (e.g., MSP430F5438 has 3 16-bit timers), they may be insufficient to fulfill all the requests together.

In order to facilitate the multiple timing requirements, we design a timing mechanism that can potentially provide an unlimited number of timers. The system maintains a system time which is updated when a timer is read or expired. The system time does not begin to elapse after system start-up until the first timing request is being serviced. It is updated each time when the timer expires.

A task can request timing service anytime regardless of the availability of hardware timers. When a new request is received, the expiration time is first converted according to the system time and stored in the record of all unserved requests. If this expiration time, say, $t^*_{exp}$, is earlier than all the unserved requests, the timer is re-set to expire at $t^*_{exp}$. Otherwise, no changes will be made to the timer.

### 6.3.3 Summary of Platform Design

In this section, we have described the main design and implementation of our platform including MDC and sensors. To evaluate the performance of related algorithms and protocols for MDG in WSNs, users can implement the platform using digital design methodologies to construct complex computing unit on the FPGA, or C/C++ programs for high level abstraction and control on the MCU/PowerPC.

# Chapter 7

# Experiments of Mobile Data Gathering in Wireless Sensor Networks: A Case Study

It has been acknowledged that when compared to software simulation, the deployment of protocols and algorithms in real applications may exhibit worse result. Many factors contribute to such performance deterioration. One major reason is the inaccurate modeling of the applications in the design phase. The carefully designed protocols or algorithms may Another important impact comes from oversimplified simulation environment, which inappropriately exaggerates the performance. In this chapter, as a case study, we implement a wildlife monitoring system on our platform. Our experimental results demonstrate that real implementations can evaluate many practical performance factors which would have a great impact on the sensing results and are very difficult to fully capture by theoretical models and simulations.

The rest of the chapter is organized as follows. Section 7.1 introduces the background of our experimental work. Section 7.2 presents the design and implementation of a wildlife monitoring system on our platform introduced in Chapter 6. Section 7.3 provides and analyzes the experimental results of the experiment. Finally, Section 7.4 concludes the chapter.

## 7.1 Introduction

We present a experimental study of mobile data gathering in a WSN for wildlife detection to demonstrate such deficiency of software simulation. We focus on the problem that there is abundant presence of Canada geese on our campus. Due to protection of wildlife, the number of Canada geese grows dramatically over years in North America. Though natural migrants, spoiled Canada geese cause potential health problems, as they may carry parasites, bacteria, fungus or even bird flu dangerous to other domestic species (Fig. 7.1). Its ubiquitous presence may also become a nuisance to people when a large flock linger on airports, grass fields, playgrounds, etc. In order to minimize the adverse effects, there are companies providing control services to keep them away from important areas (e.g., airports).

A common approach is to dispatching patrols to possible areas. Once geese are found, they repel the flock with human force or herding dogs (e.g., Border Collies). The disadvantage of this approach is obvious since the patrol cannot cover all the locations in continuous time due to high labor cost. For geese control on our campus, as an example, the patrol vehicle arrives every few hours. In addition, a gaggle cannot be found until the patrol arrives, perhaps the gaggle have stayed for a long time and already caused troubles. Thus, the patrol needs to be sent more frequently at even higher cost. For example, in 2014, Day County in South Dakota spent $179,200 on its geese depredation program.

A more economic alternative is to use sensors for monitoring and reporting as soon as geese are detected. In this way, patrols can be dispatched purposely after receiving a report. The gathered data can also be analyzed to predict patterns of time and location for their future arrivals. Then patrols would be dispatched beforehand for quick response. The sensing data can be gathered by MDC and uploaded to the data sink for further analysis.

## 7.2 Experiment Implementation

We have implemented this application on our platform. In our experiment, sensors are deployed in the field, and they detect the presence of geese using ultrasonic transducers. Sensors are organized into clusters and a cluster head is selected for each cluster. MDC first traverses the network to collect location information of

(a) Goose

(b) Sanitation issues

(c) Repel with dog decoy

Figure 7.1: Potential health problems caused by Canada Geese.

cluster heads, and calculates the trajectory for data gathering. After that, every 10 minutes MDC starts from the origin, moves autonomously along the trajectory, collects sensory data from cluster heads, and stores them with other data in the memory. On our platform, we need to implement a sensing mechanism and data transmission protocols on sensors, and data transmission protocols and mobility control on MDC. In particular, mobility control includes trajectory calculation, positioning, vehicle control, and data logging. The functional blocks of MDC and sensor node are summarized in Fig. 7.2.

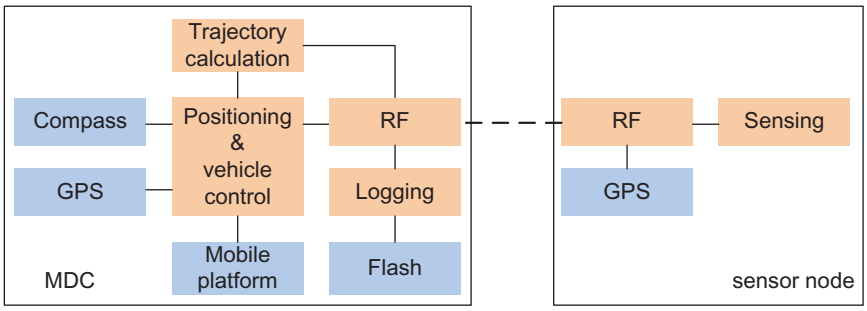



Figure 7.2: Functional blocks of MDC and sensor node for geese control.

### 7.2.1 Sensing Mechanism on Sensor Nodes

x

In this experiment, sensors work in a duty-cycle mode to monitor the presence of geese. In every second, the node wakes up, transmits ultra-sonic pulses and waits for echo. If echo is received in the given time, it considers the object as present; otherwise the object is considered as absent. A node only records the time of status change (either from presence to absence or vice versa). False alarm, which is a common problem in WSN design regardless of data gathering methods used, may occur in our case when uninterested objects are detected. To keep the design simple, in our experiment, we consider ultra-sonic sensors and leave object recognition using computer vision to future works. We focus on diminishing the false alarms caused by the movement of MDC, which occurs when MDC enters the sensing range of a node during a migration tour. By differentiating the mobility patterns of geese and MDC, we found that geese always appear in a large number

and wander slowly in a confined neighborhood (takes more than a benchmark of 10 seconds). In contrast, MDC passes by a node quickly even it sojourns for data gathering (data uploading usually is done within 10 s). When the detected existence of an object lasts for a short time and the presence occurs around the time MDC is performing data gathering, the object is considered as MDC and the detection result is discarded.

To track the activity of geese, time stamps need to be added upon sensory data so nodes are synchronized to Coordinated Universal Time (UTC). The GPS mounted on cluster heads can provide timing information up to one-second precision. For power consumption concerns, however, the GPS module is turned off after it obtains its position. Hence, another method is needed for time synchronization. A straightforward solution is to maintain a Real-Time Clock (RTC) on the node and keep it running regardless of power status (ON or OFF). This may induce extra hardware cost and increase energy consumption due to the real-time clock hardware. Alternatively, several algorithms/protocols for time synchronization can be applied in WSNs [140, 141]. We design a simple synchronization mechanism which enables synchronization among sensors with the assistance from MDC. MDC maintains a real-time clock which is set to the local time when it is powered on and dispatched for data collection. During movement, it disseminates timing information together with requesting messages. After receiving such messages, each sensor adjusts its RTC accordingly and updates the time stamp of its records. The precision of the synchronization mechanism has one-minute accuracy, which is sufficient for our application.

### 7.2.2 Data Transmission among MDC and Sensor Nodes

Based on the wireless communication structure presented in Fig. 6.3, we implemented the proposed data transmission mechanism on MDC and sensor nodes. A simplified data gathering protocol is employed to collect data from sensors efficiently. The nodes are organized into clusters and the cluster heads have GPS modules. They initiate the clustering process by sending *clustering* messages. Each node chooses the one-hop neighbor, from which it receives the *clustering* message with the smallest recorded number of hops, as the next-hop neighbor for data uploading. MDC sends out *data-request* message when it arrives at an anchor (location

of the cluster head). The *data-request* message is broadcast in the cluster, and each node uploads its sensory data with DATA messages after receiving the *data-request* message.

Reliable message transmission is guaranteed with the underlying MAC protocol, a Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) with back-off mechanism is implemented on both nodes and MDC to ensure reliable transmissions. For overhead concerns, RTS/CTS is not implemented.

### 7.2.3 Mobility Control on MDC

For autonomous movement, MDC first acquires the location information from sensors, especially the location of the cluster heads, calculates trajectory and moves along the route via the mobile platform.

#### 7.2.3.1 Moving trajectory planning

When sensor nodes are deployed and powered on, the GPS modules start up and wait for satellite signals (it takes less than $35$s for the GPS to fix from cold start up). Then MDC is powered on after sensor nodes. It ensures that the nodes obtain their location information before MDC is ready. When the location information is available on MDC, it first calculate the trajectory it should follow for data gathering.

Since the SenCar visits the locations of the cluster heads (anchor points),the trajectory of the SenCar, which connects all the anchor points, is usually calculated by solving the Traveling Salesman Problem (TSP). Since TSP is NP-hard and it is necessary to provide quick solution for mobile data gathering, heuristic algorithms are applied. In our experiment, for the purpose of demonstration, we adopt nearest neighbor algorithm to find a short tour.

#### 7.2.3.2 Motion control

after the moving trajectory has been calculated, the SenCar starts to move in the network and broadcasts *location-request* messages. It traverses along S-shaped lines, the distances in between are less than $30$ m. Since CC2400 is able to communicate reliably within $20$ m, it guarantees that each node can receive the message,

and report its location to MDC. MDC returns to the original location after receiving location information from all cluster heads.

The GPS location information contains latitude and longitude, which can be represented by 4 bytes: one byte for the integer part and 3 bytes for the fractional part. Such representation increases the complexity in computations, such as additions. In order to simplify calculations, the locations are transformed into coordinates in a Cartesian coordinate system with the origin at the initial location of MDC (the x-axis pointing eastward, and the y-axis pointing northward). The coordinates are now represented by two 2-byte numbers in the precision of centimeter.

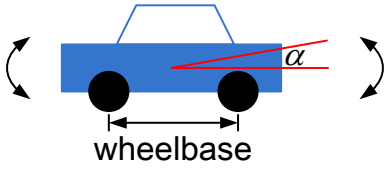
Given the locations of cluster heads and the initial position of MDC, the next step is to determine its trajectory. This problem is the well-known Traveling Salesmen Problem and a variety of efficient algorithms exist. In our experiment, we solve it using the nearest neighbor algorithm [142] on the FPGA. It sorts the anchors according to the Euclidean distances to MDC as the visiting sequence. To navigate MDC in the correct direction, the current orientation is needed. This can be calculated using the measurement from the digital compass in MPU9250.

$$Azimuth = \arctan Y_r / X_r \tag{7.1}$$

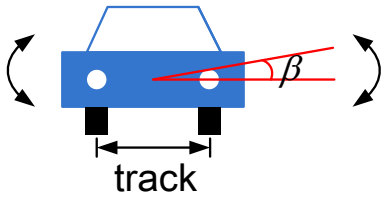
where $X_r$ and $Y_r$ are the adjusted measurement of the compass according to the attitude of MDC,

$$\begin{cases} X_r = X\cos\alpha + Y\sin\alpha\sin\beta - Z\cos\beta sin\alpha \\ Y_r = Y\cos\beta + Z\sin\beta \end{cases} \tag{7.2}$$

where $X$, $Y$ and $Z$ are the measurement output from the digital compass, $\alpha$ is the *pitch angle* in Fig. 7.3(a), and $\beta$ is the *roll angle* in Fig. 7.3(b). Precise values of pitch and roll angles can be calculated with the output of the gyrometer and accelerometer. To minimize bumpy conditions while moving, the vehicle is customized to have long wheelbase (26 cm) and track (30 cm) so they can help it keep balance during movement ($\alpha \simeq 0$ and $\beta \simeq 0$, and $X_r \simeq X$ and $Y_r \simeq Y$). We use the refined measurement directly to calculate MDC's direction, which a trailed moving average of 8 most recent measurements. Finally, MDC's orientation can be calculated by adjusting the azimuth with the magnetic declination of the location of MDC, which is the angle on the horizontal plane between magnetic north and true north, i.e., $Orientation = azimuth + magnetic\_declination$.




(a) Pitch: head up or down about the vehicle's axis.




(b) Roll: rotate about an axis running from the vehicle's front to its rear.

Figure 7.3: Illustration of attitude of MDC.

Vehicle control includes both steering and motor control. The mechanism of steering control is illustrated in Fig. 7.4. The black arrow points to the north and the green arrow points to the current direction of MDC. Direction angle $\theta_1$ can be calculated as explained above, and direction angle $\theta_2$ can be calculated using the location of MDC and the anchor. MDC steers to the right if $\theta_1 + \theta_2 \leq 180°$; otherwise, it steers to the left. MDC begins to transmit *data-request* message when it is within 20 m from the anchor. If an ACK message is received, it stops moving and starts to receive data from the cluster head. After a preset sojourn time, MDC moves to the next anchor.

**Remarks:** Due to the habit of Canada geese, large concentrations are usually found on flat, open grass fields. To this end, in our experiment, sensors are deployed with no obstacle in between. In fact, autonomous driving/control is a very popular area in computer vision community. In this chapter, we focus on system integration of different components to build a mobile computing and wireless platform. With these basics, we could implement more complex algorithms for object recognition (deep neural networks) on MDC in future.

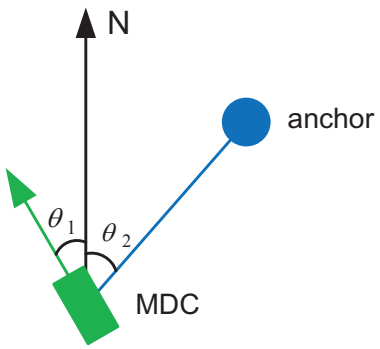



Figure 7.4: Principle of steering control of MDC.

### 7.2.4 Experiment Data Logging

For performance analysis after the experiment, data needs to be saved in non-volatile memory. On MDC, we use the BPI NOR Flash on the HPCS to store sensing data, locations of anchors, real-time locations of MDC during data gathering, and the time consumed for each migration tour.

## 7.3 Experimental Results

Having described the implementation of the geese control application on our platform, in this section, we present the experimental results and analyze the impacts on the overall performance in a real environment. We execute the experiment on an open grass field on our campus. A number of 18 sensors are deployed in the field. Among them, 6 sensors have GPS modules installed (assigned as cluster heads). To verify the capability of collecting data from disconnected nodes, nodes are deployed in such a way that 6 nodes are disconnected to the rest of 12 nodes.

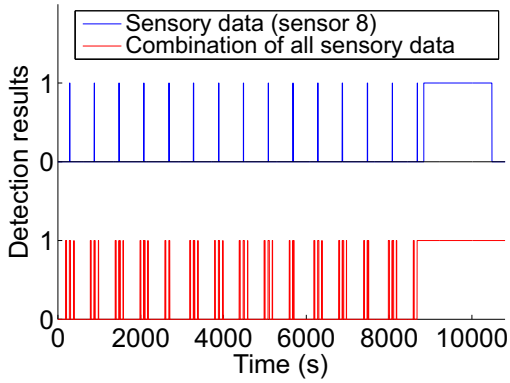
Fig. 7.5 shows the topology of the experimental network. The blue star indicates the original location of MDC, the yellow stars indicate the locations of anchors (cluster heads), and the red star at the lower left corner of the figure indicates the location of the origin of coordinate system as discussed in Section 7.2.3. MDC is dispatched to collect sensory data every 10 minutes for a total of 18 migration tours as the experiment runs for 3 hours.

Figure 7.5: Network topology. The blue star refers to the original location of MDC; the yellow stars refer to the location of anchors; the red star refers to the locations of the origin of coordinate system.

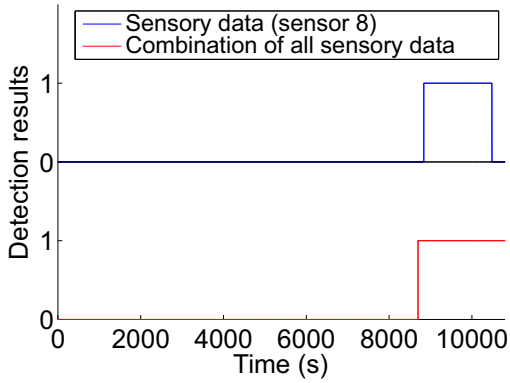
#### 7.3.0.1 Impact of MDG on Sensory Data

We first present the sensory data collected by MDC to verify their correctness in the application. Although data analysis is an important step to produce reliable sensing information, it is not the main focus of this chapter. As discussed, the sensors monitor the area using ultrasonic transducers every second, and MDC collects sensing data every 10 minutes. The sensory data is presented in Fig. 7.6, in which '0' indicates no object detected, while '1' indicates objects detected. The blue line indicates the sensory data obtained from node 8, and the red line indicates the convergence of sensory data from all nodes.

It is observed in Fig. 7.6(a) that there are spikes in the curves of individual sensor detection results when no flock is around, which are considered as "false alarms." Compare the time of those false alarms and trajectory of MDC, we can see that they are caused by MDC when it approaches sensors. MDC stays around a sensor for at most a few seconds, which is much shorter than the presence of geese as they usually stay in the same neighborhood for a longer time. Hence, we further refine the data by eliminating the spikes appearing at the same time when




(a) Sensory data of sensor 8 and converged sensory data before refinery. False alarms are presented as spikes.




(b) Sensory data of sensor 8 and converged sensory data after refinery. False alarms have been removed by considering the presence of MDC.

Figure 7.6: Sensory data: before and after data refinery considering MDC's impact on the correctness of sensing results.

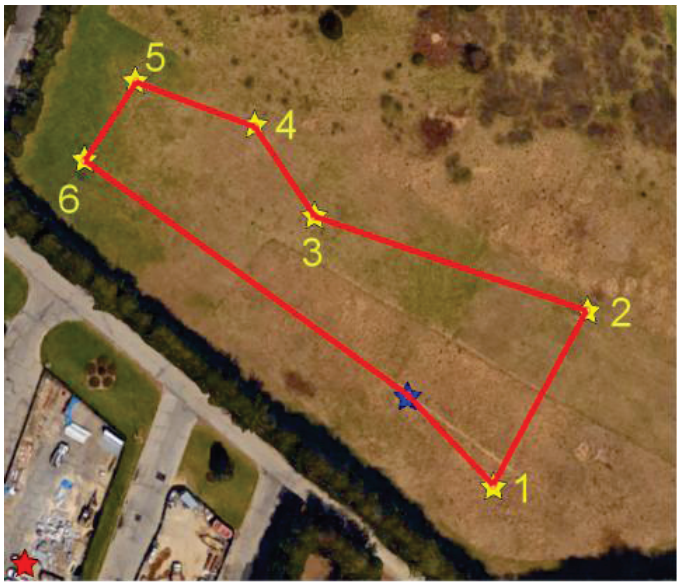
MDC performs data collection. Fig. 7.6(b) shows the sensing data after removing the false alarms caused by MDC.

#### 7.3.0.2 Calculated Trajectory vs. Actual Course: Cause of Differences

Fig. 7.7 exhibits the difference between the calculated trajectory and the actual course of MDC. The red lines in Fig. 7.7(a) indicates the planned trajectory, which is constructed by connecting anchors by the nearest neighbor algorithm. The yellow tracks indicate the movement of MDC. The two curves differ in shapes which implies that MDC does not move exactly along the trajectory.

Such inconsistency is due to several practical factors. The first one comes from the strategy of mobility control. In the experiment, MDC stops for data gathering when moves into the communication range of an anchor. Such impact is illustrated in Fig. 7.8(a) where the black dashed lines indicate the calculated trajectory, and the green solid curve indicates the actual course of MDC. MDC does not need to physically contact anchor A for communication. Instead, it stops within its transmission range before it actually visits the exact location of A. MDC collects data, and steers towards the next anchor B. In this approach, more optimizations can be done to yield shorter trajectories. In general, a larger transmission range results in shorter trajectories. It suggests that a more refined theoretical model than TSP should be found for further optimizations (e.g., the Traveling Salesmen Problem with Neighborhood (TSPN)). In TSPN, the salesmen only need to visit the neighborhood around cities (within the transmission range of sensors).

The second factor is the positioning error of both anchors and MDC due to the limitation of GPS accuracy. In our platform, the accuracy is 2.5 meters. The third factor is the error in orientation estimation due to measurement and calculation. Deviation caused by GPS error and orientation estimation error are analyzed in Fig. 7.8(b). $A$ and $B$ denote the real positions of MDC and the anchor respectively, and $\theta_1$ denotes the correct direction from $A$ to $B$. Due to imperfect GPS positioning, $A'$ and $B'$ are the incorrect locations obtained from their GPS modules, and $\hat{\theta}_1$ denotes the correct orientation of MDC. Meanwhile, due to the error in digital compass measurement, the orientation of MDC is indicated by $\hat{\theta}_2$ instead of $\theta_2$. As discussed in Section 7.2.3, MDC may steer towards an incorrect direction, thus increase the length of the trajectory. The bigger the error and velocity of the

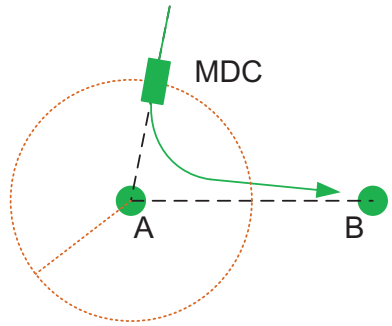



(a) Calculated trajectory.

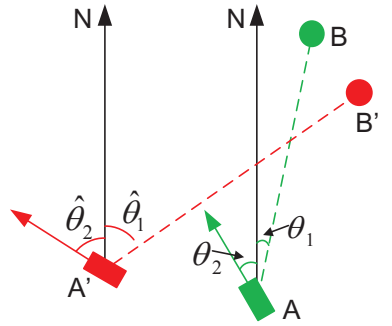
(b) Real course of MDC.

Figure 7.7: MDC trajectories. The red lines in the left figure present the calculated trajectory and the yellow tracks in the right figure present the actual course of MDC.

vehicle, the longer the trajectory. It is observed from Fig. 7.8 that the first factor (impact from node's transmission range) dominates the length of the trajectory in our experiment.




(a) Smooth turning as a result of non-contact communication.




(b) Contribution of GPS and orientation errors to steering control.

Figure 7.8: Analysis of causes of trajectory deviation.

#### 7.3.0.3 Traveling Time of MDC

The traveling time of MDC is directly associated with data latency for MDG, which is defined as the time elapsed between the generation of sensing data and its successful delivery to the base station for processing. After sensing data is generated at a sensor, it is buffered until picked up by MDC. The final delivery has to wait for MDC to complete the entire migration route, which is a major contributor to the data latency.

In this experiment, we examine the relations between physical environment and traveling time. Fig. 7.9 presents the traveling time of MDC in 18 migration tours.

The traveling time for each migration tour differs in a small amount, and is longer than the theoretical calculation ( $t_{migration} = \frac{L_{trajectory}}{v_{MDC}}$). The difference is analyzed below. First, different trajectories described in Section 7.3.0.2 result in different lengths of migration tour. Second, ground undulations and slopes extend the length of the trajectory whereas the theoretical Euclidean distance is calculated based on longitude and latitude information. For example, when a vehicle moves on a slope with grade $58\%$, for every $100$ m it moves horizontally, the actual moving distance on the slope is approximately $115$ m. The extra distance increases as the grade increases. Since our experiment is conducted in a nearly flat area, except that the trajectory between anchors 2 and 3 spans over a steep slope. MDC moves from the bottom of the slope to the top, which increases the length of the tour and decreases the velocity of MDC, thus the traveling time increases. Third, acceleration and deceleration of MDC result in different velocities at real-time. However, theoretical models usually consider the velocity as a constant which is set to the maximum speed measured on flat ground. In our experiment, grass and small stones in the field often causes MDC to decelerate so its actual speed is barely a constant. Therefore, the real traveling time could be very different from theoretical calculations.

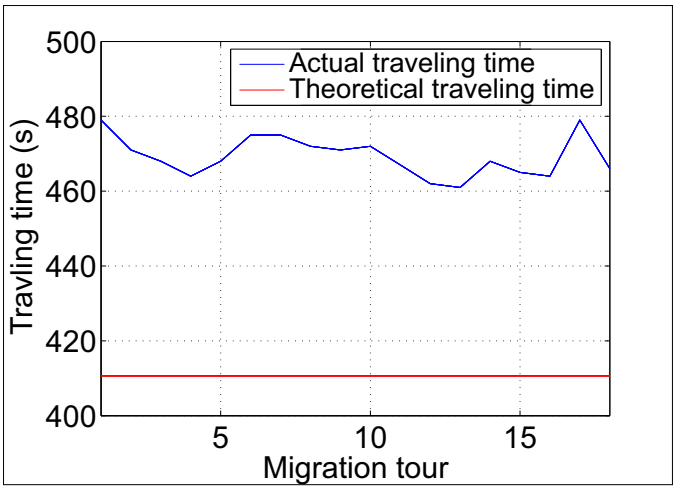



Figure 7.9: Traveling time for 18 successive migration tours. It takes more time in the experiment to perform each data gathering task than theoretical calculation.

Since the terrain affects the traveling time of MDC, it should be considered in the system, e.g., determining dispatching frequency for MDC to satisfy data latency.

As mentioned above, simple calculations based on Euclidean distance tends to be inaccurate when the sensors are actually deployed on undulating terrain or slopes. It is helpful to use 3-D coordination if steep slope is considered. However, it is still not easy to obtain accurate altitude (z-axis) provided by the GPS so any theoretical calculations would be unfortunately inaccurate. It is important to note that these factors are necessary to be considered in design and network planning for large-scale networks. A solution is to use the traveling time from several runs as training sets and adopt regression or more advanced machine learning algorithms to predict traveling time. That is, for the initial tours, we record the trajectory with its traveling time and obtain more accurate estimations based on the previous data.

### 7.3.1 Summary of Experiment

From our case study experiment, we can draw the following conclusions.

- Real-world deployment has a great impact on the overall performance of MDG in WSNs.
- Mobility of MDC may affect sensing results and lead to incorrectness, which can be addressed by analyzing sensed data.
- Tour planning of MDC can be refined by considering the benefits of wireless communication to shorten the length of trajectory.
- Terrain needs to be considered during trajectory calculation, and the uncertainty can be reduced by collecting experimental results from the movement of MDC.
- The accuracy of location and altitude measurement of MDC also greatly impact the results. More accurate mobility control can be achieved through advanced navigation techniques.

## 7.4 Conclusions

In this chapter, we have developed an efficient, flexible and versatile platform for performance evaluation in MDG in WSNs. In the platform, MDC provides high

computing capability to satisfy real-time signal processing and task handling. The expandability through modular designs can meet the requirements from different applications. The sensor nodes also provide enhanced computing capability, extended memory for caching and connection to other modules for extension. By running a case study of wildlife detection, we have demonstrated the feasibility of implementing the entire system on our platform. We have analyzed the impacts from real applications and proposed solutions to minimize such impacts on sensing results. We believe our platform is a very powerful tool for studying and developing network algorithms and protocols for various MDG applications in WSNs.

# Chapter 8

# Conclusions

In this dissertation, we focus on the application of mobile data gathering, wireless recharge and energy harvesting technologies in wireless sensor networks, and present a set of schemes, including algorithm synthesis and protocol design, to address the benefit of such solutions. We first propose an optimal wireless energy replenishment and mobile data gathering mechanism in networks with controlled topologies. We study the application of this mechanism in wireless sensor networks and provide an efficient algorithm which maximizes network utility. In order to enable on-demand energy replenishment in wireless sensor networks where energy status of the sensors change dynamically, we design an NDN-based wireless recharging protocol for dynamic recharging in wireless sensor networks. We leverage concepts and mechanisms from NDN to design a set of protocols that continuously gather and deliver energy information, including unpredictable emergencies, to the mobile vehicle in a scalable and efficient manner. We derive analytic results on energy neutral conditions that give rise to perpetual operation. We also discover that optimal recharging of multiple emergencies using one SenCar is an Orienteering problem with Knapsack approximation. We further extend the problem to recharging through multiple SenCar coordination, and formulate it into a MTSP problem with deadline. To accommodate the dynamic nature of sensor energy condition and reduce computational overhead, we present a heuristic algorithm that selects the sensor with the minimum weighted sum of traveling time and residual lifetime, which not only improves network scalability, but also guarantees the perpetual operation of the network. We also present an efficient, flexible mobile data

gathering mechanism to balance the energy consumption in heterogeneous wireless sensor networks. By adaptively moving cluster heads to some positions for data gathering, the energy-consuming data forwarding tasks are shared among all the sensor nodes to balance their energy consumption. Since the moving distance of cluster heads is limited, and they communicate with the base station directly once settled down, the data collection latency in the proposed network is much shorter than that in conventional mobile data gathering. Based on this network architecture, we find optimal positions for anchor points and moving paths for cluster heads, and develop a clustering approach and determine the routes for sensor nodes to upload data to cluster heads. We design and implement a general purpose, flexible platform for mobile data gathering in wireless sensor networks. Our platform integrates both mobile data collector and sensors with high flexibility and enhanced computing capability to facilitate requirements of implementing different applications. In addition, we implement mobile data gathering scheme for wildlife detection using wireless sensor networks. The experimental results demonstrate that our platform is a powerful tool to capture many practical factors which would have a great impact on the sensing results and are very difficult to fully capture by theoretical models and simulations, thus is very helpful for accurate system modeling and performance evaluation.

To summarize, in this dissertation, we conduct comprehensive studies on mobile data gathering and energy replenishment in wireless sensor networks. We explore different network structures and propose efficient solutions to take advantages of different techniques and improve performance of data gathering and energy replenishment from both systematic and theoretical points of view, which combine algorithm design, optimization, analysis and simulation techniques. We also provide a versatile platform for mobile data gathering experiments in wireless sensor networks. The outcome of this research can be applicable to research and real application in many categories, including smart home, industry automation, precision agriculture, environmental monitoring, wildlife study, battlefield surveillance and other scopes. Therefore, our research would have a significant impact on fundamental design principles and infrastructures for the development of future sensor networks.

# Bibliography


[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “A survey on sensor networks,” *IEEE Commun. Magazine*, pp. 102-114, 2002.

[2] E.Biagioni and K. Bridges,"The application of remote sensor technology to assist the recovery of rare and endangered species," Special Issue on Distributed Sensor Networks, *International Journal of High Performance Computing Applications,* vol.16, no. 3, Aug. 2002.

[3] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J.Zhao. "Habitat monitoring: Application drivers for wireless communications technology," *Proc. of the 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, April 2001.

[4] A. Mainwaring, J.Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA ’02),* Atlanta, GA, Sept. 2002.

[5] S. Chessa and P. Santi, "Crash faults identification in wireless sensor networks", *Computer Communications,* vol. 25, no. 14, pp. 1273-1282, Sept. 2002.

[6] L. Schwiebert, S.K.S. Gupta, and J. Weinmann, "Research challeneges in wireless networks of biomedical sensors," *ACM/IEEE Conf. on Mobile Computing and Networking (MobiCom),* pp. 151-165, 2001.

[7] K. Romer and F. Mattern, “The design space of wireless sensor networks,” *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54-61, Dec. 2004.

[8] A. Tiwari, P. Ballal, and F. L. Lewis, "Energy-efficient wireless sensor network design and implementation for condition-based maintenance," ACM *Transactions on Sensor Networks (TOSN)*, vol. 3, no. 1, March 2007.

[9] M.A. Pasha, S. Derrien and O. Sentieys, "Toward ultra low-power hardware specialization of a wireless sensor network node," *IEEE INMIC*,2009.

[10] P.K. Dutta and D.E. Culler, "System software techniques for low-power operation in wireless sensor networks," *IEEE/ACM ICCAD*, 2005.

[11] X. Liang, W. Li and T. A. Gulliver, "Energy efficient modulation design for wireless sensor networks," *IEEE PacRim*,2007.

[12] W. Ye, J. Heidemann and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," *IEEE INFOCOM*, 2002.

[13] T. Spyropoulos, K. Psounis and C.S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the single-copy case," *IEEE/ACM Trans. Networking*, pp. 63-76, Feb. 2008.

[14] C. Ma and Y. Yang, "A battery-aware scheme for routing in wireless ad hoc networks," *IEEE Trans. Vehicular Technology*, vol. 60, no. 8, pp. 3919-3932, Oct. 2011.

[15] C. Ma and Y. Yang, "Battery-aware routing for streaming data transmissions in wireless sensor networks," in *ACM/Kluwer Mobile Networks and Applications (MONET)*, vol. 11, no. 5, pp. 757-767, October 2006.

[16] A. Scaglione and S. D. Servetto, "On the interdependence of routing and data compression in multi-hop sensor networks," *MobiCom*, 2002.

[17] D. Marco, E.J. Duarte-Melo, M. Liu and D. Neuhoff, "On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data," *IPSN*, April 2003.

[18] D. England, B. Veeravalli, and J. Weissman, "A robust spanning tree topology for data collection and dissemination in distributed environments," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 5, pp. 608-620, 2007.

[19] K. Jain, J. Padhye, V. N. Padmanabhan and L. Qiu, "Impact of interference on multi-hop wireless network performance," *MobiCom*, 2003.

[20] E. J. Duarte-Melo and M. Liu, "Data-gathering wireless sensor networks: organization and capacity," *Elsevier Computer Networks*, vol. 43, pp. 519-537, 2003.

[21] H. E. Gamal, "On the scaling laws of dense wireless sensor networks: the data gathering channel," *IEEE Trans. Information Theory*, vol. 51, no. 3, pp. 1229-1234, March 2005.

[22] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," *ACM Sensys*, Nov. 2003.

[23] U. Akyol, M. Andrews, P. Gupta, J. Hobby, I. Saniee and A. Stolyar, "Joint scheduling and congestion control in mobile ad-hoc networks," *IEEE INFOCOM*, 2008.

[24] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," *Hawaii Int. Conf. System Sciences (HICCS)*, Jan. 2000.

[25] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach," *IEEE INFOCOM*, 2004.

[26] A. Manjeshwar and D. P. Agrawal, "Teen: a routing protocol for enhanced efficiency in wireless sensor networks," *IEEE IPDPS*, April 2001.

[27] Z. Zhang, M. Ma and Y. Yang, "Energy efficient multi-hop polling in clusters of two-layered heterogeneous sensor networks," *IEEE Trans. Computers*, vol. 57. no. 2, pp. 231-245, Feb. 2008.

[28] B. Sheng, Q. Li and W. Mao, "Data storage placement in sensor networks," *in Proc. of ACM Mobihoc*, Florence, Italy, May 2006.

[29] R. Shah, S. Roy, S. Jain and W. Brunette, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," *Elsevier Ad Hoc Networks Journal*, vol. 1, pp. 215-233, Sept. 2003.

[30] W. Zhao, M. Ammar and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," *ACM MobiHoc*, 2004.

[31] A. Pentland, R. Fletcher and A. Hasson, "Daknet: rethinking connectivity in developing nations," *IEEE Computer*, Jan. 2004.

[32] A. Chakrabarty, A. Sabharwal and B. Aazhang, "Using predictable observer mobility for power efficient design of a sensor network," *Second International Workshop on Information Processing in Sensor Networks (IPSN)*, April 2003.

[33] D. Jea, A. A. Somasundara and M.B. Srivastava, "Multiple controlled mobile elements (data mules) for data collection in Sensor Networks," *IEEE/ACM DCOSS*, June 2005.

[34] M. Ma and Y. Yang, "SenCar: an energy-efficient data gathering mechanism for large-scale multihop sensor networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 10, Oct. 2007.

[35] M. Ma and Y. Yang, "Data gathering in wireless sensor networks with mobile collectors," *IEEE IPDPS*, 2008.

[36] W. Zhao, M. Ammar and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," *IEEE INFOCOM 2005*.

[37] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," *IEEE Realtime Systems Symposium*, Dec. 2004.

[38] E. Ekici, Y. Gu, and D. Bozdag, "Mobility-Based Communication in Wireless Sensor Networks," *IEEE Communications Magazine*, July 2006.

[39] J. Luo and J. P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," *IEEE INFOCOM 2005*.

[40] H. Nakayama, N. Ansari, A. Jamalipour and N. Kato, "Fault-resilient sensing in wireless sensor networks," *Computer Communications*, vol. 30, pp. 2375-2384, Sept. 2007.

[41] S. Nesamony, M. K. Vairamuthu, M. E. Orlowska, "On optimal route of a calibrating mobile sink in a wireless sensor network," *Fourth International Conference on Networked Sensing Systems*, June 2007.

[42] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli and Z. M. Wang. "Controlled sink mobility for prolonging wireless sensor networks lifetime". *ACM Wireless Networks*, 2007.

[43] G. Xing, T. Wang, W. Jia, and M. Li, "Rendezvous design algorithm for wireless sensor networks with a mobile base station," *ACM Mobihoc*, May 2008.

[44] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal and G. S. Sukhatme, "Robomote: enabling mobility in sensor networks," *Fourth International Workshop on Information Processing in Sensor Networks (IPSN)*, April 2005.

[45] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, "Real-time power-aware routing in sensor networks," *in Proc. of IEEE IWQoS*, 2006.

[46] R. Pon, M. A. Batalin, J. Gordon, A. Kansal, D. Liu, M. Rahimi, L. Shirachi, Y. Yu, M. Hansen, W. J. Kaiser, M. Srivastava, G. Sukhatme and D. Estrin, "Networked infomechanical systems: a mobile embedded networked sensor platform," *in Proc. of ACM/IEEE IPSN*, 2005.

[47] M. A. Batalin, M. Rahimi, Y. Yu, D. Liu, A. Kansal, G. S. Sukhatme, W. J. Kaiser, M. Hansen, G. J. Pottie, M. Srivastava and D. Estrin, "Call and response: experiments in sampling the environment," *in Proc. of ACM SenSys*, 2004.

[48] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," *in Proc. of Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2002.

[49] T. Small and Z. Haas, "The shared wireless infostation model - a new ad hoc networking paradigm (or where there is a whale, there is a way)," *in Proc. of ACM MobiHoc* 2003.

[50] B. Kusy, H. Lee, M. Wicke, N. Milosavljevic, and L. Guibas, "Predictive QoS routing to mobile sinks in wireless sensor networks,"*in Proc. of International Conference of Information Processing in Sensor Networks (IPSN)*, 2009.

[51] K. Karenos and V. Kalogeraki, "Traffic management in sensor networks with a mobile sink," *IEEE Trans. on Parallel and Distributed systems*, vol. 21, no. 10, pp. 1515 - 1530, Oct. 2010.

[52] X. Xu, J. Luo and Q. Zhang, "Delay tolerant event collection in sensor networks with mobile sink," *in Proc. of IEEE INFOCOM*, 2010.

[53] Y. Yun and Y. Xia, "Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications," *IEEE Trans. on Mobile Computing*, vol. 9, no. 9, pp. 1308 - 1318, Sept. 2010.

[54] D. Bote, K. Sivalingam and P. Agrawal, "Data gathering in ultra wide band based wireless sensor networks using a mobile node," *in Proc. of Fourth International Conference on Broadband Communications, Networks, and Systems (BroadNets)*, Raleigh, NC, Sept. 2007.

[55] S.J. Roundy, "Energy scavenging for wireless sensor nodes with a focus on vibration to electricity conversion," *Ph.D. Dissertation, EECS, UC Berkeley*, 2003.

[56] A. Kurs, A. Karalis, R. Moffatt, J.D. Joannopoulos, P. Fisher and M. Soljacic, "Wireless power transfer via strongly coupled magnetic resonances," *Science* vol.317, pp.83, 2007.

[57] A. Karalis, J.D. Joannopoulos and M. Soljacic, "Efficient wireless non-radiative mid-range energy transfer," *Annals of Physics*, vol. 323, no. 1, pp 34-48, 2008.

[58] Intel, "Wireless resonant energy link (WREL) demo," *http://software.intel .com/en-us/videos/wireless-resonant-energy-link-wrel-demo/*.

[59] B. Kang and G. Ceder, "Battery materials for ultrafast charging and discharging," *Nature* vol. 458, pp.190-193, 2009.

[60] M. Ma and Y. Yang, "Adaptive triangular deployment algorithm for unattended mobile sensor networks," *IEEE Trans. Computers*, vol. 56, no. 7, pp. 946-958, 2007.

[61] X. Liu and M. Haenggi, "The impact of the topology on the throughput of interference-limited sensor networks with rayleigh fading", *Proc. IEEE SECON*, pp. 317-327, 2005.

[62] K. S. Prabh and T. Abdelzaher, "On scheduling and realtime capacity of hexagonal wireless sensor networks", *Proc. the 19th Euromicro Conference on Real-Time Systems*, pp. 136-145, 2007.

[63] K. Kar, A. Krishnamurthy and N. Jaggi, "Dynamic node activation in networks of rechargeable sensors," *IEEE/ACM Trans. Networking*, vol. 14, no. 1, pp. 15-26, 2006.

[64] L. Lin, N.B. Shroff and R. Srikant, "Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources," *IEEE/ACM Trans. Networking*, vol. 15, no. 5, pp. 1021-1034, 2007.

[65] R.S. Liu, P. Sinha and C.E. Koksal, "Joint energy management and resource allocation in rechargeable sensor networks," *IEEE INFOCOM*, 2010.

[66] V. Sharma, U. Mukherji, V. Joseph and S. Gupta, "Optimal energy management policies for energy harvesting sensor nodes," *IEEE Trans. Wireless Communications*, vol. 9, no. 4, 2010.

[67] C. Vigorito, D. Ganesan and A. Barto, "Adaptive control of duty cycling in energy-harvesting wireless sensor networks," *IEEE SECON*, 2007.

[68] Y. Peng, Z. Li, W. Zhang and D. Qiao, "Prolonging sensor network lifetime through wireless charging," *IEEE RTSS*, 2010.

[69] Y. Shi, L. Xie, Y.T. Hou and H. D. Sherali, "On renewable sensor networks with wireless energy transfer," *IEEE INFOCOM*, 2011.

[70] M. Zhao, M. Ma and Y. Yang, "Efficient data gathering with mobile collectors and space-division multiple access technique in wireless sensor networks ," *IEEE Trans. Computers*, vol. 60 no. 3, pp. 400-417, 2011.

[71] M. Gatzianas and L. Georgiadis, "A distributed algorithm for maximum lifetime routing in sensor networks with mobile sink," *IEEE Trans. Wireless Communications*, vol. 7, no. 3, pp. 984-994, 2008.

[72] PowerCast Corp, "http://www.powercastco.com."

[73] M. Rahimi, H. Shah, G. Sukhatme, J. Heideman and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," *IEEE International Conference on Robotics and Automation*, 2003.

[74] A. Kansal, J. Hsu, M. Srivastava and V. Raqhunathan, "Harvesting aware power management for sensor networks," *43rd ACM/IEEE Design Automation Conference*, 2006.

[75] P. Vansteenwegen, W. Souffriau and D. Van Oudheusden, "The orienteering problem: a survey," *European Journal of Operation Research*, vol. 209, no. 1, 2011.

[76] T. Tsiligirides, "Heuristic methods applied to orienteering," *Journal of the Operation Research Society*, vol. 35, no. 9, pp. 797-809, 1984.

[77] B. Golden, A. Assad and R. Dahl, "The orienteering problem," *Naval Research Logistics* 34, pp. 307-318, 1987.

[78] I. Chao, B. Golden and E. Wasil, "A fast and effective heuristic for the orienteering problem," *European Journal of Operation Research*, vol. 88, no. 3, 1996.

[79] M. Fischetti, J. S. Gonzalez and P. Toth, "Solving the orienteering problem through branch-and-cut," *INFORMS Journal on Computing*, vol. 10, no. 2, pp. 133-148, 1998.

[80] B. Tong, Z. Li, G. Wang and W. Zhang, "How wireless power charging technology affects sensor network deployment and routing," *IEEE ICDCS*, 2010.

[81] S. He, J. Chen, F. Jiang, D. Yau , G. Xing and Y. Sun, "Energy provisioning in wireless rechargeable sensor networks," *IEEE INFOCOM*, 2011.

[82] M. Zhao, J. Li and Y. Yang, "Joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks," *IEEE ITC*, 2011.

[83] Y. Shi, L. Xie, T. Hou and H. Sherali, "On renewable sensor networks with wireless energy transfer," *IEEE INFOCOM*, 2011.

[84] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs and R. Braynard, "Networking named content," *ACM CoNEXT*, 2009.

[85] I. Chatzigiannakis, A. Kinalis, S. Nikoletseas and J. Rolim, "Fast and energy efficient sensor data collection by multiple mobile sinks," *ACM MobiWac*, 2007.

[86] Panasonic Ni-MH battery handbook, "http://www2.renovaar.ee/userfiles/Panasonic_Ni-MH_Handbook.pdf".

[87] Y. Wang, M. C. Vuran and S. Goddard, "Stochastic analysis of energy consumption wireless sensor networks," *Proc. of IEEE Secon*, 2010.

[88] R. Nelson, *Probability, stochastic processes, and queueing theory: the mathematics of computer performance modeling*, Springer, 1995.

[89] S. Ross, *A First Course in Probability*, 8th Ed, Prentice Hall, 2009.

[90] C. Miller, A. Tucker and R. Zemlin,"Integer programming formulation of traveling salesman problems," *Journal of the ACM*, pp. 326-329, 1960.

[91] B. Gavish, "A note on the formulation of the m-salesman traveling salesman problem," Management Science, 1976.

[92] M.W.P. Savelsbergh, "Local search in routing problems with time windows," *Annals of Operation Research*, pp. 285-305, 1985.

[93] S. Mitrović-Minić, R. Krishnamurti, "The multiple TSP with time windows: vehicle bounds based on precedence graphs," *Operations Research Letters*, pp. 111 - 120, 2006.

[94] M. Desrochers, J.K. Lenstra, M.W.P. Savelsbergh and F. Soumis, "Vehicle routing with time windows: optimization and approximation," Elsevier Science Publisher, 1988.

[95] N. Bansal, A. Blum, S. Chawla and A. Meyerson, "Approximation algorithms for deadline-TSP and vehicle routing with time windows," *ACM STOC*, 2004.

[96] M. Gendreau, A. Hertz, G. Laporte and M. Stan, "A generalized insertion heuristic for the traveling salesman problem with time windows," *Journal Operations Research*, vol. 46 no. 3, 1998, pp. 330-335.

[97] A. Tanenbaum, *Modern Operating Systems*, pp. 125, 3rd Ed, Prentice Hall, 2007.

[98] H. Jiang, S. Jin and C. Wang, "Prediction or not? An energy-efficient framework for clustering-based data collection in wireless sensor networks ," *IEEE Trans. Parallel and Distributed Systems*, vol. 22, issue 6, pp. 1064-1071, June 2011.

[99] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu and S. Singh, "Exploiting heterogeneity in sensor networks," *IEEE INFOCOM*, 2005.

[100] M. Ma and Y. Yang, "Clustering and load balancing in hybrid sensor networks with mobile cluster heads," *3rd International Conference on Quality of Service in Heterogeneous Wired/wireless Networks*, 2006.

[101] Z. Zhang, M. Ma and Y. Yang, "Energy-efficient multi-hop polling in clusters of two-layered heterogeneous sensor networks," *IEEE Trans. Computers*, vol. 57, no. 2, pp. 231-245, 2008.

[102] X. Kui, S. Zhang, J. Wang and J. Cao, "An energy-balanced clustering protocol based on dominating set for data gathering in wireless sensor networks ," *IEEE ICC*, 2012

[103] M. Zhao and Y. Yang, "Mobile data gathering with load balanced clustering and dual data uploading in wireless sensor networks," *IEEE Trans. Mobile Computing*, vol. 14, no. 4, pp. 770-785, 2014.

[104] M.D. Francesco, S.K. Das and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: a survey" *ACM Trans. Sensor Networks*, vol. 8, no. 1, Aug. 2011.

[105] M. Ma and Y. Yang, "SenCar: An energy efficient data gathering mechanism for large scale multihop sensor networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 10, pp. 1476-1488, 2007.

[106] S.J. Roundy, "Energy scavenging for wireless sensor nodes with a focus on vibration to electricity conversion," *Ph.D. Dissertation, Dept. of EECS, UC Berkeley*, May 2003.

[107] B. Gaudette, V. Hanumaiah, S. Vrudhula and M. Krunz, "Optimal range assignment in solar powered active wireless sensor networks," *IEEE INFOCOM*, 2012.

[108] T. Zhu, A. Mohaisen, Y. Ping and D. Towsley, "DEOS: dynamic energy-oriented scheduling for sustainable wireless sensor networks," *IEEE INFOCOM*, 2012.

[109] M. Zhao, J. Li and Y. Yang, "A framework of joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks," *IEEE Trans. Mobile Computing*, vol. 13, no. 12, pp. 2689-2705, 2014.

[110] Y. Shi, L. Xie, T. Hou and H. Sherali, "On renewable sensor networks with wireless energy transfer," *IEEE INFOCOM*, 2011.

[111] C. Wang, J. Li, F. Ye and Y. Yang, "Multi-vehicle coordination for wireless energy replenishment in sensor networks," *IEEE IPDPS*, 2013.

[112] C. Wang, J. Li, F. Ye and Y. Yang, "NETWRAP: An NDN based real-time wireless recharging framework for wireless sensor networks," *IEEE Trans. Mobile Computing,* vol. 13, no. 6, pp. 1283-1297, 2014.

[113] C. Wang, J. Li, F. Ye and Y. Yang, "'Recharging schedules for wireless sensor networks with vehicle movement costs and capacity constraints," *IEEE SECON*, Singapore, June 2014.

[114] M. Ma and Y. Yang, "Adaptive triangular deployment algorithm for unattended mobile sensor networks," *IEEE Trans. Computers*, vol. 56, no. 7, pp. 946-958, 2007.

[115] "Serial Peripheral Interface Bus," *https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus*

[116] VMEbus International Trade Association, "ANSI/VITA 57.1-2008 American National Standard for FPGA Mezzanine Card (FMC) Standard," *http://www.vita.com*

[117] J.L. Hennessy, D.A. Patterson, "Computer architecture: a quantitative approach"

[118] D.E. Taylor, A. Herkersdorf, A. Döring and G. Dittmann, "Robust header compression (ROHC) in next-generation network processors," *IEEE Trans. Networking*, vol. 13, no. 4, pp. 755-768, 2005.

[119] H. M. Ammari, and S. K. Das, "Data Dissemination to Mobile Sinks in Wireless Sensor Networks: An Information Theoretic Approach," *Proc. of IEEE MASS*, 2005.

[120] M. Zhao, M. Ma, and Y. Yang, "Efficient Data Gathering with Mobile Collectors and Space-Division Multiple Access Technique in Wireless Sensor Networks," *IEEE Trans. Computers*, vol. 60, no. 3, pp. 400-417, 2011.

[121] M. Ma and Y. Yang, "SenCar: An energy efficient data gathering mechanism for large scale multihop sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 10, pp. 1476-1488, October 2007.

[122] M. Zhao and Y. Yang, "Bounded relay hop mobile data gathering in wireless sensor networks," *IEEE Transactions on Computers*, vol. 61, no. 2, pp. 265-277, Feb. 2012.

[123] M. Ma, Y. Yang and M. Zhao, "Tour planning for mobile data gathering mechanisms in wireless sensor networks," *IEEE Transactions on Vehicular Technology,* vol. 62, no. 4, pp. 1472-1483, May 2013.

[124] M. Zhao and Y. Yang, "Optimization based distributed algorithms for mobile data gathering in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 10, pp. 1464-1477, Oct. 2012.

[125] M. Zhao, Y. Yang and C. Wang, "Mobile data gathering with load balanced clustering and dual data uploading in wireless sensor networks" *IEEE Transactions on Mobile Computing*, vol. 14, no. 4, pp. 1536-1233, March 2015.

[126] X. Xu, J. Luo, and Q. Zhang, "Delay Tolerant Event Collection in Sensor Networks with Mobile Sink," *Proc. of IEEE INFOCOM*, pp. 1-9, 2010.

[127] http://www.isi.edu/nsnam/ns/

[128] https://omnetpp.org/

[129] S. Sundresh, W. Kim, and G. Agha, "SENS: A Sensor, Environment and Network Simulator," *Proc. of ACM Annual Simulation Symposium*, pp. 221-228, 2004.

[130] J. Polley, D. Blazakis, J. McGee, D. Rusk, J. S. Baras, and M. Karir, "ATEMU: A Fine-grained Sensor Network Simulator," *Proc. of IEEE SECON*, Santa Clara, pp. 145-152, 2004.

[131] D. Johnson et al., "Mobile Emulab: A Robotic Wireless and Sensor Network Testbed," *Proc. of IEEE INFOCOM*, pp. 1-12, 2006.

[132] T. Ducrocq, J. Vandaële, N. Mitton, and D. Simplot-Ryl, "Large Scale Geolocalization and Routing Experimentation with The SensLAB Testbed," *Proc. of IEEE MASS*, pp. 751-753, 2010.

[133] A. Förster et al., "Poster Abstract: MOTEL: Towards Flexible Mobile Wireless Sensor Network Testbeds," *Proc. of International Conference on Embedded Wireless Systems and Networks (EWSN) Poster Session*, pp. 1-2, 2011.

[134] P. De, A. Raniwala, S. Sharma, and T. Chiueh, "Mint: A miniaturized Network Testbed for Mobile Wireless Research," *Proc. of IEEE INFOCOM*, pp. 2731-2742, vol. 4, 2005.

[135] D. Raychaudhuri et al., "Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols," *Proc. of IEEE WCNC*, pp. 1664-1669, vol.3, 2005.

[136] A. Arora, E. Ertin, R. Ramnath, W. Leal, and M. Nesterenko, "Kansei: a high-fidelity sensing testbed," *IEEE Internet Computing*, vol. 10, no. 2, pp.35-47, 2006.

[137] J. Polo ., G. Hornero, C. Duijneveld, A. García, and O. Casas, "Design of A Low-cost Wireless Sensor Network with UAV Mobile Node for Agricultural Applications," *Computers and Electronics in Agriculture*, vol. 119, pp. 19-32, 2015.

[138] G. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities,"*ACM AFIPS*, 1967

[139] T. Good, and M. Benaissa, "AES on FPGA from the Fastest to the Smallest," *Proc. of International Conference on Cryptographic Hardware and Embedded Systems (CHES)*, pp. 427-440, 2005.

[140] S. Ganeriwal, R. Kumar and M. B. Srivastava, "Timing-Sync Protocol for Sensor Networks," *Proc. of ACM SENSYS*, pp. 138-149, 2003.

[141] K. Noh, E. Serpedin, and K. Qaraqe, "A New Approach for Time Synchronization in Wireless Sensor Networks: Pairwise Broadcast Synchronization," *IEEE Trans. Wireless Communications*, vol. 7, no. 9, pp. 3318-3322, 2008.

[142] T.H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, MIT Press, 2001.