Stony Brook University

Battery-Aware and Energy-Efficient Algorithms for Wireless Networks

A Dissertation Presented

by

Chi Ma

to

The Graduate School
in Partial Fulfillment of the
Requirements
for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

August 2007

Stony Brook University

The Graduate School

Chi Ma

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

Dr. Yuanyuan Yang, Advisor
Professor
Department of Computer Science
Department of Electrical and Computer Engineering

Dr. Jie Gao, Chairperson of Defense
Assistant Professor
Department of Computer Science

Dr. Michael A. Bender
Associate Professor
Department of Computer Science

Dr. Samir R. Das
Associate Professor
Department of Computer Science

Dr. Sangjin Hong
Associate Professor
Department of Electrical and Computer Engineering

This dissertation is accepted by the Graduate School
Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

Battery-Aware and Energy-Efficient

Algorithms for Wireless Networks

by

Chi Ma

Doctor of Philosophy

in

Computer Science

Stony Brook University

2007

This thesis proposes an integrated suite of battery-aware and energy-efficient algorithms for routing and scheduling in wireless mobile ad hoc networks, wireless sensor networks and wireless mesh networks. The thesis includes several closely related topics in wireless networks: (1) On-line computable mathematical battery models for efficient battery capacity calculation on wireless devices; (2) Battery-aware routing schemes for wireless mobile ad hoc networks; (3) A battery-aware backbone scheduling algorithm for self-organized wireless sensor networks; (4) A cross-layer scheduling scheme for urban area high density sensor networks; (5) Battery-aware hot spot covering algorithms for wireless mesh networks; (6) Battery-aware transmission radius scheduling algorithms for wireless mesh networks; (7) Battery-aware client driven mesh router scheduling algorithms for wireless mesh networks.

Wireless networks, such as wireless mobile ad hoc networks, wireless sensor networks and wireless mesh networks, have played an increasingly important role in a wide range of

applications. Different wireless networks are composed of different wireless devices with various network architectures. The wireless mobile ad hoc network consists of various mobile and battery-powered wireless devices, such as PDAs, laptops and cellular phones. These wireless personal devices form the network in an ad hoc way to let devices communicate with each other. The wireless sensor network is composed of hundreds or thousands of distributed wireless sensors, with each sensor having limited battery energy supply, transmission radius and sensing capability. Sensed data are continuously propagated to a data sink. The wireless mesh network consists of a mix of fixed routers and mobile clients interconnected via access points. Wireless mesh network provides Internet connections to its mobile clients. More and more mesh network applications require mesh routers to be battery-powered due to their flexibility to deploy and their independency of wall power. Battery power has emerged as a key component for energy management in these wireless networks, especially for packet routing and wireless device scheduling. With battery technology lagging behind, the batteries on wireless personal devices, wireless sensors and mesh routers can only last a few hours for work. Although the energy capacity of batteries has been increased by 10% to 15% per year, it still cannot keep up with the increasing energy demand of wireless devices. It is critical to improve energy efficiency of these wireless networks.

This thesis provides a suite of battery-aware and energy-efficient algorithms and schemes for routing and scheduling in wireless mobile ad hoc networks, wireless sensor networks and wireless mesh networks. (1) The on-line computable, discrete time mathematical battery models are designed to accurately calculate battery discharging loss and battery residual capacity in an energy-efficient way. The calculation of battery discharging loss in the models is simplified and requires low computation complexity and little memory.

(2) A battery-aware power metric is introduced for battery-aware routing in wireless mobile ad hoc networks. Based on the metric, battery-aware routing scheme and prioritized battery-aware routing scheme are proposed to improve energy efficiency of packet routing in wireless mobile ad hoc networks. (3) A virtual backbone scheduling scheme for data propagation and distribution among sensors is proposed based on the mathematical battery model. The scheme constructs a battery-aware connected dominating set to let fatigue sensors recover batteries and prolong the lifetime of wireless sensor networks. (4) A cross-layer scheduling scheme is proposed for urban area high density wireless sensor networks. The scheme consists of three parts: the spanning tree partition algorithm to control the topology, the intersection MAC layer protocol to provide efficient MAC communication around urban intersections and the urban emergency service algorithm for differentiated network services. (5) A router-level battery lifetime optimization scheduling algorithm is proposed to maximize the lifetime of battery-powered mesh routers. A network-wide spanning tree scheduling algorithm is designed to improve lifetime of wireless mesh networks with battery-awareness. (6) A battery-aware mesh network energy scheduling scheme is proposed to dynamically schedule multiple input multiple output mesh routers' radii based on battery behaviors. The scheme consists of two algorithms: the coverage algorithm and the backhaul routing algorithm. Both algorithm adopt the battery model designed for multiple input multiple output mesh routers. (7) A cross-layer battery-aware client driven scheme is designed to efficiently schedule mesh network coverage. The key idea of this scheme is to let neighboring mesh routers collaboratively adjust their transceiver radii based on positions of mesh clients. This scheme also propose the MAC layer algorithm to jointly minimize the cost among different layers.

Contents

List of Tables

List of Figures

Acknowledgements

Firstly, I would like to acknowledge my advisor, Dr. Yuanyuan Yang, for her continual support and advice throughout this work. Without her inspiration, insight, perceptiveness and guidance through these years, I would not have been able to carry out this work.

I would like to thank my colleagues Zhenghao Zhang, Hui Zhang, Ming Ma, Deng Pan, Min Yang, Lin Liu, Miao Zhao, Xi Deng, Jin Wang, Guowen Han, Ji Li, Bin Tang and Xiangdong Qin in High Performance Computing and Networking Research Laboratory for their friendship and all of their help over years.

Many thanks to my beloved wife, Jin Zhou, who has been a great source of strength all through this work. I would particularly appreciate my parents, Jianyao Ma and Huiqiu Gao, and parents-in-law, Mingzuo Zhou and Huifen Gu, for their support and love in these years.

I would also like to thank Dr. Chellury R. Sastry from Siemens Corporate Research, Princeton, New Jersey, for inviting me to spend a summer internship in his group in the summer of 2005. I enjoyed excellent working climate and many fruitful discussions with him and with my colleagues: Peng Zhao, Michael Loiacono, Ling Shen, Xavier Leaute and Visvanathan Ramesh. Special thanks to Michael Loiacono, who helped me learn how to write nesC codes for Telos sensor motes. I would also like to express my gratitude to the many friends and interns I met in Siemens Corporate Research: Lingyun Liu, Lan Dong,

Yan Yang, Hui Xie, Shaorong Liu, Yijian Bai, Jian Li, Jie Shao, Chunxiao Zhou, Wanmei Ou and Minmin Han. Thank you for making my internship a pleasant and memorable experience.

Finally, thanks to my committee members, Dr. Michael A. Bender, Dr. Samir R. Das, Dr. Jie Gao, and Dr. Sangjin Hong, who offered guidance and support.

Chapter 1

Introduction

This chapter explains the motivation, design goals, contributions and outline of the thesis.

1.1 Motivation and Design Goals

In recent years, wireless mobile ad hoc networks (MANET), wireless sensor networks (WSN) and wireless mesh networks (WMN) have played an increasingly important role in a wide range of applications [1, 2, 3, 4, 30, 67, 68, 69]. A MANET consists of various mobile wireless devices, such as PDAs, laptops and cellular phones [1, 2]. Applications in MANETs are multimedia transmission service, video conference, IP telephony and interactive games. A WSN is a distributed wireless network which is composed of a large number of self-organized unattended sensor nodes [28, 30, 60]. A typical function of WSNs is to collect data in a sensing environment. Usually the sensed data in such an environment is routed to a data sink, which is the central unit of the network [38]. WMNs

have emerged as a low-cost, convenient and flexible extension to the wired network infrastructure [67, 68, 69]. A WMN is a hybrid network which consists of a mix of fixed routers and mobile clients interconnected via access points [27, 67]. The mesh routers together with wireless mesh clients, such as personal wireless devices, form multi-functional wireless communication systems [66, 69, 72]. More and more wireless routers, especially in outdoor mesh network applications, are powered by batteries for their flexibility to deploy and their independency of wall power [68, 70].

Personal devices, sensors and mesh routers in these wireless networks have limited power supply. Energy efficiency is critical for these networks. On one hand, applications of these wireless networks require support of high data throughput and long lifetime. For example, outdoor mesh routers need to maintain activities for at least 24 hours and wireless sensors are typically expected to work for years. On the other hand, with battery technology lagging behind, batteries on wireless personal devices, sensors, and routers do not achieve long enough lifetimes for work. Although battery capacity has been increased by 10% to 15% per year [11, 12], it still does not keep up with the increasing energy demands from energy-intensive applications on wireless devices. Battery lifetime therefore emerges as a key factor that affects the performance of wireless networks [5, 6, 11]. Carefully scheduling and budgeting battery energy in wireless networks has become an urgent and critical issue. In the thesis we will propose several schemes to efficiently schedule routing and activities of devices in MANETs, WSNs and WMNs.

The design goals of this thesis can be classified into the following categories.

- **Mathematical Battery Models.** The first goal is to design on-line computable mathematical battery models to describe the special battery behaviors. Recent study in battery technology reveals the energy consumed from a battery is not equivalent to

the energy dissipated in the device [5, 10]. When discharging, batteries tend to consume more energy than needed. They reimburse the over-consumed energy later if they have sufficient recovery, where "recovery" means the battery is disconnected from its load or the current is lowered down. The process of the reimbursement is often referred to as *battery recovery*. The over-consumed energy is referred to as *discharging loss*. A "fatigue" battery is a battery with high discharging loss, while a "well-recovered" battery is a battery with low discharging loss. Experiments show that the discharging loss might take up to 30% of the total battery capacity [10]. Hence, precisely capturing and predicting battery behavior is essential for optimizing system performance in wireless networks. Mathematical battery models have been introduced in recent years to capture the relationships among battery capacity, battery discharging loss and electric current [9, 8, 10, 12]. Previous battery models either employ large look-up tables that require considerable efforts to configure, or highly depend on experimental inputs. To make it feasible for energy critical, time sensitive and low storage capacity wireless devices, a good battery model need to on-line compute the battery capacity accurately with low energy requirement. In this sense, previous battery models, although have good performance when applied to wired networks, are not suitable for wireless scenarios. In this thesis we will develop battery models that are feasible on-line computation in MANET, WSN and WMN, according to their different network features.

- **Battery-Awareness in MANETs, WSNs and WMNs.** The second goal is to apply our battery models to routing and scheduling in wireless networks. Different wireless networks have different network features and scheduling requirements. (i) Routing is the most important network activity in MANETs. Battery-aware routing schemes for

MANET should route packets in the ad hoc network topology and, at the meantime, minimize energy consumption on devices. MANETs also require differentiated services for connections with different priorities. The battery-aware MANET routing scheme need to consider the quality of service of network connections. (ii) WSNs adopt backbone routing with packets relayed by a set of backbone sensors. Therefore backbone scheduling is the most energy-consuming network activity in WSNs. Battery-aware backbone scheduling schemes for WSNs need to determine sensor battery status: Only the well-recovered sensors can be chosen as backbone sensors, the rest should to be put into sleep for recovery. An efficient scheme should construct the backbone without consuming too much sensor energy. (iii) We also propose to design battery-aware scheduling algorithms for hot spot covering in WMN. For those mesh routers with multiple transceivers, special router scheduling algorithms will be designed to consider the multiple current inputs. Finally we will consider client driven algorithms in WMN coverage scheduling. In this way routers can efficiently minimize the power consumption to cover mesh clients.

- **Energy-Efficiency in Urban Area High Density WSNs.** This goal is to design an energy-efficient scheduling scheme for urban area high density (UAHD) WSNs by reducing their high density. Energy-efficient scheduling in UAHD sensor networks is very important. Research has revealed that without efficient network-wide power scheduling, large scale and high density sensor networks cannot achieve good power performance. In this thesis we will study the unique characteristics of UAHD sensor networks. We will propose novel solutions to reduce sensor density in UAHD networks by partitioning a network into multiple layers of overlapped low-density subnetworks. We will also design a media access control (MAC) layer algorithm

for UAHD networks to jointly improve system performance among layers. Finally, handling urban emergencies is a critical issue in urban security. We will design an algorithm to provide differentiated services for urgent packets.

1.2 Contributions

The novel contribution of this thesis includes:

- **On-Line Computable Discrete Time Battery Models.**[21] We derive accurate analytical battery models for capturing the special battery behavior on different wireless devices. The models also employ an approach to simplify the computation of recovery battery capacity. We customize our battery models for MANET, WSN and WMN, according to their different network features. Performance evaluations show that our models can accurately describe battery behavior with little memory requirement and low computation complexity.

- **Battery-Aware Routing Schemes for MANETs.**[22, 23, 26] The battery-aware routing scheme (BAR) is designed to dynamically facilitate battery-awareness in MANET routing. BAR is independent of specific routing protocols. It enables routing protocols to set up routing paths from a battery-awareness point of view. In addition, we design an enhanced prioritized battery-aware routing scheme (PBAR) for time sensitive applications in MANETs. The objective of PBAR is to guarantee the end-to-end routing connections with a set of measurable attributes, in terms of routing delay, data throughput and device lifetime.

- **Battery-Aware Backbone Scheduling Algorithm for WSNs.**[24] By employing the mathematical battery model, we design a virtual backbone scheme to schedule data

collecting in sensor networks. The battery-aware connected dominating set (BACDS) is introduced to take advantage of battery recovery. BACDS organizes fatigue sensors to recover batteries and let well-recovered sensors work as backbone nodes. A distributed approximation algorithm is designed to construct the BACDS in an efficient way.

- **Cross-Layer Scheduling for Urban Area WSN.** We propose novel solutions to reduce sensor density in UAHD sensor networks by partitioning a network into multiple layers of overlapped low-density subnetworks. Node densities among different layers are therefore balanced. We then propose an efficient Cross-Layer Power Aware Scheduling (CLPAS) scheme. CLPAS consists of three parts: Spanning Tree Partition (STP) topology control algorithm, Intersection MAC (I-MAC) protocol and Urban Emergency Service (UES) algorithm. STP is a distributed algorithm that can partition an UAHD sensor network in polynomial time. Our simulation results demonstrate that it achieves very close performance to the optimal algorithm in various scenarios. We also designed the I-MAC protocol for sensor communications around intersections. It employs an adaptive time division schedule to dynamically manage packet traffic based on the observations on recent traffic. Finally, the UES algorithm is designed to provide differentiated services. We conduct simulations to evaluate the performance of the CLPAS scheme in the Times Square area of New York City.

- **Battery-Aware Hot Spot Covering Algorithms for WMNs.**[25] The battery lifetime optimization scheduling algorithm (BLOS) is designed to maximize the lifetime of battery-powered mesh routers based on the study of the relationships between discharging time and total battery lifetime. With each mesh router follows the BLOS policy, our goal is to keep the mesh network covering all hot spots for as long as

possible. The spanning tree scheduling algorithm (STS) is designed to dynamically organize routers to cover hot spots based on the BLOS policy. The time complexity of the STS algorithm is $O(r)$ for a network with r mesh routers. Performance evaluations demonstrate that the STS algorithm can greatly improve the lifetime, data throughput and energy consumption efficiency of a wireless mesh network.

- **Battery-Aware Router Scheduling Algorithms for Multiple Input Multiple Output WMNs.**[27] We study the battery model for battery-powered mesh routers to show that mesh routers can improve their energy efficiency by collaboratively adjusting their transceiver radii to alternatively recover their batteries. An efficient battery-aware multiple input multiple output (MIMO) mesh network power scheduling scheme (BAMPS) is proposed to maximize the lifetime of mesh networks. The scheme consists of two algorithms: the coverage algorithm and the backhaul routing algorithm which are used to schedule mesh radio transceivers for network coverage and network backhaul routing, respectively. We also conduct performance evaluations to show the improvements of network lifetime by implementing BAMPS scheme.

- **Cross-Layer Battery-Aware Client Driven Scheduling Algorithms for WMNs.**[29] We design a cross-layer client driven battery-aware (CDBA) scheduling scheme to efficiently schedule mesh network coverage. CDBA scheme consists of two algorithms: CDBA coverage algorithm and CDBA MAC algorithm. The key idea of CDBA coverage algorithm is to let neighboring mesh routers collaboratively adjust their transceiver radii based on positions of mesh clients. This algorithm is a distributed algorithm with $O(n)$ time complexity where n is the maximum number of neighbors of a router in the network. To further jointly improve system performance

among layers, we design the CDBA MAC algorithm to provide a seamless and fast service for handoff among mesh routers. We conduct simulations to evaluate the performance of the proposed CDBA scheme.

Fig. 1.1 depicts the tasks, design goals and contributions of this research.

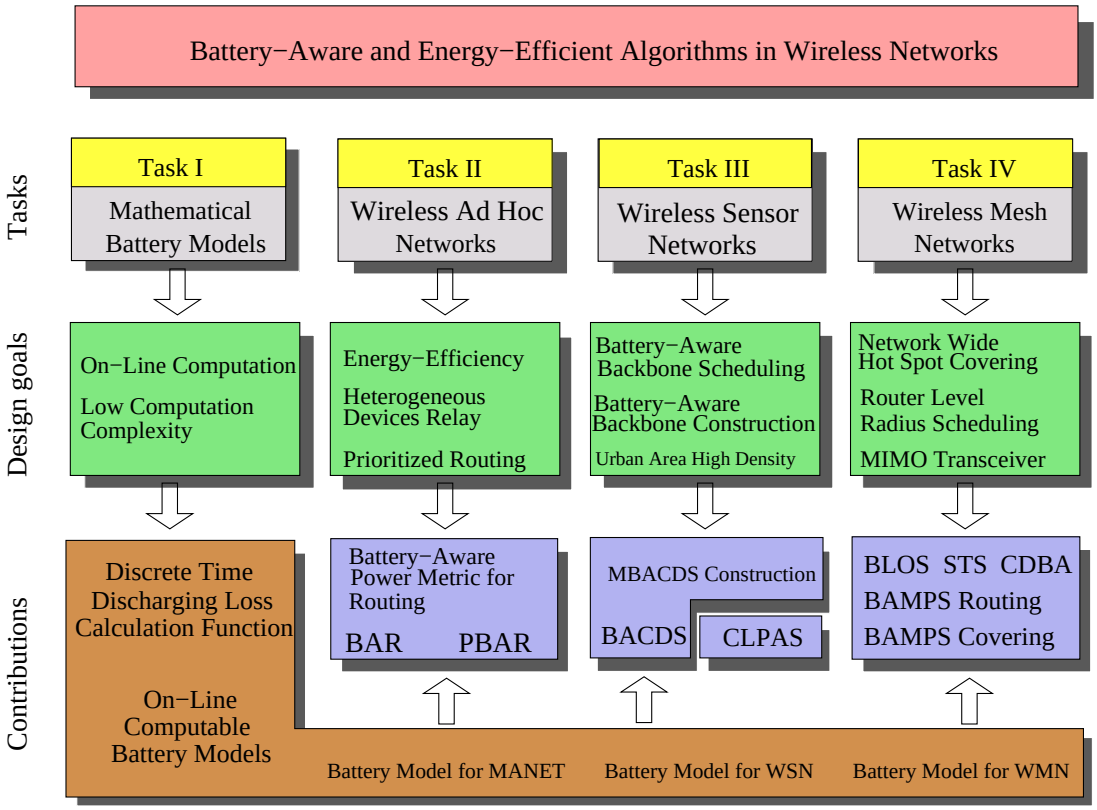


Figure 1.1: The tasks, design goals and contributions of this thesis.

This research combines protocol design, algorithm design, analytical, probabilistic and simulation techniques to conduct comprehensive studies on the above issues. The research will have a significant impact on fundamental design principles and infrastructures for the development of future wireless networks. The outcome of this project will be applicable to a wide spectrum of applications, including space, military, environmental, health care,

home and other commercial areas.

1.3 Thesis Outline

The rest of the thesis is organized as follows. We introduce battery-aware routing schemes in Chapter 2. We present our on-line battery model for MANET along with its performance analysis. Based on this on-line model we develop battery-aware schemes: battery-aware routing scheme and prioritized battery-aware routing scheme, for MANET routing. Chapter 3 is contributed to study sensor network scheduling. This chapter designs a battery model for WSN and gives the BACDS model along with its performance comparison with the previous model, then presents a distributed approximation algorithm to construct BACDS. Chapter 4 presents a cross-layer scheduling scheme for urban area high density wireless sensor networks. Chapter 5 presents the battery-aware hot spot covering algorithms for WMNs. An approximation algorithm is also introduced in this chapter. Chapter 6 presents the battery-aware router scheduling algorithms for MIMO WMNs. Performance evaluations are conducted as well. Chapter 7 presents the battery-aware client driven scheduling scheme for WMNs. Finally, we conclude the thesis in Chapter 8.

Chapter 2

Battery-Aware Routing for MANET

This chapter presents the battery-aware routing schemes for MANETs. We first propose an on-line computable, discrete time mathematical model to capture battery discharging behavior. The model has low computational complexity and data storage requirement. It is therefore suitable for on-line battery capacity computation in routing. Our evaluations indicate that the model can accurately capture the behavior of battery discharging. Based on this battery model, we then propose a battery-aware routing scheme (BAR) for MANETs. BAR is a generic scheme that implements battery-awareness in routing protocols and is independent of any specific routing protocol. By dynamically choosing the devices with well recovered batteries as routers and leaving the "fatigue" batteries for recovery, BAR scheme can effectively recover the device's battery capacity to achieve higher energy efficiency. Our simulation results demonstrate that by adopting BAR scheme, network lifetime and data throughput can be increased by up to 28% and 24%, respectively. The results also show that BAR achieves good performance in various networks composed of different devices, batteries and node densities. Finally, we also propose an enhanced prioritized battery-aware routing scheme (PBAR) for time sensitive applications in MANETs. Our

simulation results illustrate that PBAR achieves good performance in terms of end-to-end delay and data throughput.

The rest of this chapter is organized as follows. In Section 2.1 we discuss related work to place our work in context. We review the previous off-line battery models in Section 2.2 and present our on-line battery model in Section 2.3 along with its performance analysis. Based on this on-line model we develop battery-aware schemes for MANET routing in Section 2.4. In Section 2.4.1, we discuss the previous power metric for routing and introduce a more accurate battery-aware power metric. By using this power metric, we propose a battery-aware routing scheme (BAR) in Section 2.4.2, and the enhanced prioritized battery-aware routing scheme (PBAR) in Section 2.4.3. Section 2.5 evaluates the performance of the proposed schemes. Finally we give concluding remarks in Section 2.6.

2.1 Related Work

In a MANET each node is willing to forward data for other nodes, and the determination of which nodes forward data is made dynamically based on the network connectivity. Packets are routed from sender node to destination node through neighbor MANET nodes. Routing protocols can be classified into two categories: topology-based routing and position-based routing [4, 14]. Topology-based routing protocols [13, 14], which are originally used in wired networks, depend on the link information to make routing decisions. However, the topology of a MANET changes too frequently to be updated timely. Maintaining a routing table at each node consumes a significant amount of energy in wireless devices. To avoid maintaining routing tables, position-based routing protocols are proposed [17, 18, 19]. In general, this type of protocol can achieve high flexibility and low

energy dissipation [15]. In position-based routing protocols, each node determines its own position using *Global Positioning System* (GPS) [16, 20]. The sender can forward packets based on only the location information of the destination nodes and its one-hop neighbors. Depending on the way to choose the next hop routing node, several different position-based protocols have been proposed. *Most Forward within Radius (MFR)* [18] chooses the next routing hop farthest away within communication distance, therefore to find a minimum hop-count routing path, while *Nearest with Forward Progress (NFP)* [19] attempts to choose the nearest node as the next forwarding node to minimize the energy required per routing task. The *compass routing* method, also referred to as DIR [17], aims to route the packet to the neighbor with the closest angle to the destination, while *Geographic Distance Routing* (GEDIR) protocol [17] selects a routing hop that is the closest to the destination, and the routing path provided in this algorithm is loop-free.

The wireless devices in a MANET are typically PDAs, laptops and cellular phones. These wireless devices use nickel-cadmium or lithium-ion batteries as their energy providers, and each of them may have different battery capacity and energy dissipation characteristics [1]. The rising of MANET comes up with a critical energy issue. Energy-efficient routing is necessary for MANETs.

2.2 Battery Discharging and Recovery

In this section we study the battery chemistry and previous battery behaviors models. We first discuss battery behaviors for different operations. We then study previous off-line battery models and compared them with each other.

2.2.1 Battery Chemistry

Nickel-cadmium and lithium-ion batteries are the most commonly used batteries in battery-powered wireless devices. Such a battery consists of cells arranged in series, parallel, or a combination of both. Two electrodes, an anode and a cathode, separated by an electrolyte, constitute the active material of each cell. When the cell is connected to a load, a reduction-oxidation reaction transfers electrons from the anode to the cathode. To illustrate this phenomenon, Fig. 2.1 shows a simplified symmetric electrochemical cell. In a fully charged cell as shown in Fig. 2.1(a), the electrode surface contains the maximum concentration of active species. When the cell is connected to a load, an electrical current flows through the external circuit. Active species are consumed at the electrode surface and replenished by diffusion from the bulk of the electrolyte. However, this diffusion process cannot keep up with the consumption, and a concentration gradient builds up across the electrolyte as shown in Fig. 2.1(b). A higher load electrical current I results in a higher concentration gradient and thus a lower concentration of active species at the electrode surface [7]. When this concentration falls, the battery voltage drops. As the voltage is below a certain cutoff threshold, the electrochemical reaction can no longer be sustained at the electrode surface, and the battery stops working as shown in Fig. 2.1(e). The electro active species that have not yet reached the electrode are not used. The unused charge, referred to as discharging loss, is not physically "lost," but simply unavailable due to the lag between the reaction and the diffusion rates. Before the battery dies, if the battery current I is reduced to zero or a very small value, that is, in the battery recovery as shown in Fig. 2.1(c), the concentration gradient flattens out after a sufficiently long time, reaching equilibrium again. The concentration of active species near the electrode surface following this recovery period makes unused charge available again for extraction as shown in Fig.

2.1(d). Effectively recovering the battery can reduce the concentration gradient and recover discharging loss, hence prolong the lifetime of the battery as shown in Fig. 2.1(f). Next we analyze battery discharging functions to mathematically model battery behaviors.

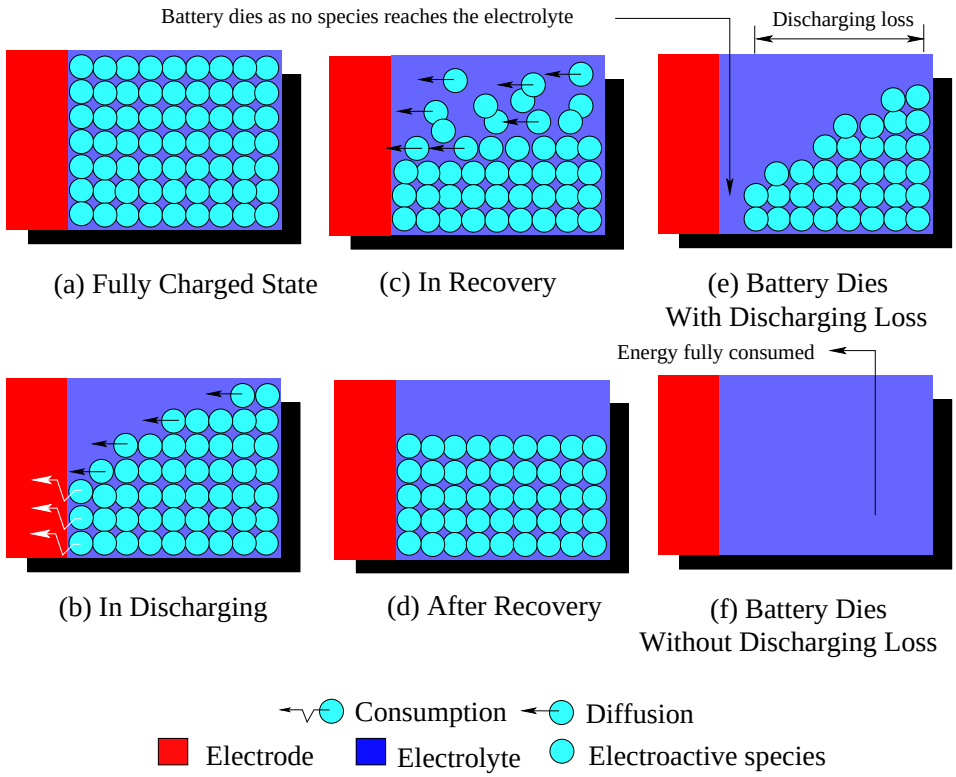


Figure 2.1: Battery operation at different states.

2.2.2 Previous Battery Models

In this subsection we review previous mathematical battery models that describe battery behaviors for battery activity scheduling. Mathematical battery models have been introduced in recent years to capture the relationships among battery capacity, battery discharging loss and electric current [9, 8, 12, 10]. Mathematical battery models can be classified into three categorizes: look-up table based models, experimental input based models

and battery capacity function based models: (1) An abstract model based on look-up tables was provided in [8] to represent battery recovery behavior. This model treats discharge and recovery as a decreasing exponential function and represents discharge and recovery as a transient stochastic process. However, as pointed out in [12], this method adopts large lookup tables that require considerable effort to configure, and its accuracy and computational complexity are barely acceptable. Therefore, it has limited utility for implementation in energy sensitive MANET. (2) An analytical battery model which can accurately estimate the battery behavior was proposed in [10]. This model combines a high-level representation of the battery and analytical expressions based on physical laws. The high-level representation is determined by experimental data input. This model can effectively capture the effect of battery discharge and recovery. But depending on experimental inputs highly limits its general usage. (3) Battery models with an analytical battery capacity function were proposed in [9, 10]. They adopt an off-line battery capacity function which sums parameters from 1 to ∞ to calculate discharged battery capacity. This summation requires long computing time that is not feasible for time sensitive real-time computation in routing. A modification to these models was introduced in [9] by employing pre-computed tables to avoid summing from 1 to ∞. However the great number of pre-computed parameters makes the model not suitable for wireless devices with low memory capacity.

To make it feasible for energy critical, time sensitive and low storage capacity wireless devices, a good battery model need to on-line compute the battery capacity accurately with low energy requirement. In this sense, previous battery models, although have good performance when applied to wired networks, are not suitable for MANETs. The key point of on-line computable battery model is to develop an accurate on-line function for battery capacity computation. To serve for this purpose we first analyze off-line battery capacity

function developed in [9, 10].

Off-line function models the energy α dissipated by the battery during time $[t_{begin}, t_{end}]$ as

$$\alpha = I \times F(T, t_{begin}, t_{end}, \beta) \tag{2.1}$$

where

$$F(T, t_{begin}, t_{end}, \beta) = t_{end} - t_{begin} + 2 \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 (T - t_{end})} - e^{-\beta^2 m^2 (T - t_{begin})}}{\beta^2 m^2} \tag{2.2}$$

This model can be explained as follows. The dissipated energy α in (2.1) contains two terms. The first term, $I \times (t_{end} - t_{begin})$, is simply the energy consumed in the device during time $[t_{begin}, t_{end}]$. The second term, $2\sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 (T - t_{end})} - e^{-\beta^2 m^2 (T - t_{begin})}}{\beta^2 m^2}$, is the amount of battery discharging loss in the duration. β (> 0) is a constant, which is an experimental chemical parameter and may vary from battery to battery. The larger the β, the faster the battery diffusion rate, hence the less the discharging loss. T is the battery lifetime and I is the battery discharge current.

As we discussed, though this off-line function can precisely capture the battery behavior, it is difficult to directly use it in MANETs due to the high computation complexity. Also this is a continuous time function. It is applied to the duration $[t_{begin}, t_{end}]$ for an arbitrary length. Because a MANET has a packetized nature where its time is split into discrete time slots with a fixed slot length, this makes it possible to model the battery energy consumption in a simpler discrete way with on-line computation ability.

2.3 Mathematical Battery Model for MANET

In this subsection, we develop an on-line computable battery model with on-line battery capacity function. This model is suitable for discrete time MANET communication. We first reduce the computation complexity of summation in a battery capacity function. Note that in (2.2) the term $e^{-\beta^2 m^2 (T-t_{end})} - e^{-\beta^2 m^2 (T-t_{begin})}$ decreases exponentially. We introduce an approximation function $\overline{F}(T, t_{begin}, t_{end}, \beta)$ to reduce the computational complexity of $F(T, t_{begin}, t_{end}, \beta)$ as follows

$$\begin{aligned} \overline{F}(T, t_{begin}, t_{end}, \beta) &= (t_{end} - t_{begin}) + 2 \sum_{m=1}^{\infty} \frac{e^{-\beta^2 (T-t_{end})} - e^{-\beta^2 (T-t_{begin})}}{\beta^2 m^2} \\ &= (t_{end} - t_{begin}) + 2 \times (\sum_{m=1}^{\infty} \frac{1}{\beta^2 m^2}) \times \left[e^{-\beta^2 (T-t_{end})} - e^{-\beta^2 (T-t_{begin})} \right] \end{aligned} \tag{2.3}$$

Considering the convergent progression in (2.3)

$$\sum_{m=1}^{\infty} \frac{1}{\beta^2 m^2} = \frac{\pi^2}{6\beta^2} \tag{2.4}$$

the simplified function is

$$\overline{F}(T, t_{begin}, t_{end}, \beta) = (t_{end} - t_{begin}) + \frac{\pi^2}{3\beta^2} \left[e^{-\beta^2 (T-t_{end})} - e^{-\beta^2 (T-t_{begin})} \right] \tag{2.5}$$

Note that $\overline{F}(T, t_{begin}, t_{end}, \beta)$ does not roll up to ∞ thus can greatly simplify the battery capacity computation.

We now show that (2.5) can closely approximate (2.2) so that we can use $\overline{F}(T, t_{begin}, t_{end}, \beta)$

to replace $F(T, t_{begin}, t_{end}, \beta)$ in (2.1), that is, (2.1) can be rewritten as

$$\alpha = I \times \overline{F}(T, t_{begin}, t_{end}, \beta) \tag{2.6}$$

where $\overline{F}(T, t_{begin}, t_{end}, \beta)$ is described as in (2.5). Given t_{begin} and t_{end}, we define the difference of two functions as $P(\beta, T)$

$$\begin{aligned} P(\beta, T) &= \overline{F}(T, t_{begin}, t_{end}, \beta) - F(T, t_{begin}, t_{end}, \beta) \\ &= 2 \sum_{m=1}^{\infty} \left[\frac{1}{\beta^2 m^2} e^{-\beta^2 (T - t_{end})} (1 - e^{-\beta^2 (t_{end} - t_{begin})}) - \frac{e^{-\beta^2 (T - t_{end}) m^2}}{\beta^2 m^2} + \frac{e^{-\beta^2 (T - t_{begin}) m^2}}{\beta^2 m^2} \right] \end{aligned} \tag{2.7}$$

Let

$$A = e^{-\beta^2 (T - t_{end})} (1 - e^{-\beta^2 (t_{end} - t_{begin})}), \;\; B = \beta^2 (T - t_{end}), \text{ and } C = \beta^2 (T - t_{begin})$$

A, B and C are independent of m. Then (2.7) can be rewritten as

$$P(\beta, T) = 2 \sum_{m=1}^{\infty} \left[\frac{A}{\beta^2 m^2} \right] + 2 \sum_{m=1}^{\infty} \left[\frac{e^{-Cm^2}}{\beta^2 m^2} - \frac{e^{-Bm^2}}{\beta^2 m^2} \right] \tag{2.8}$$

Since $e^{-Bm^2} < 1$ and $e^{-Cm^2} < 1$, $\sum_{m=1}^{\infty} \frac{e^{-Bm^2}}{\beta^2 m^2}$ and $\sum_{m=1}^{\infty} \frac{e^{-Cm^2}}{\beta^2 m^2}$ both are convergent.

At the same time, since $\sum_{m=1}^{\infty} \frac{A}{\beta^2 m^2}$ is also convergent, (2.8) is convergent. Using (2.4) we obtain

$$P(\beta, T) < \frac{\pi^2 (A + 1)}{3\beta^2} \tag{2.9}$$

For today's battery, β usually is in the range of $[0.4, 1]$ and the battery lifetime T ranges from $1/2$ to 2 hours [10]. The orders of $P(\beta, T)$ and $F(T, t_{begin}, t_{end}, \beta)$ are about 10 and 10^5, respectively. Considering this, the difference between the two functions is negligible.

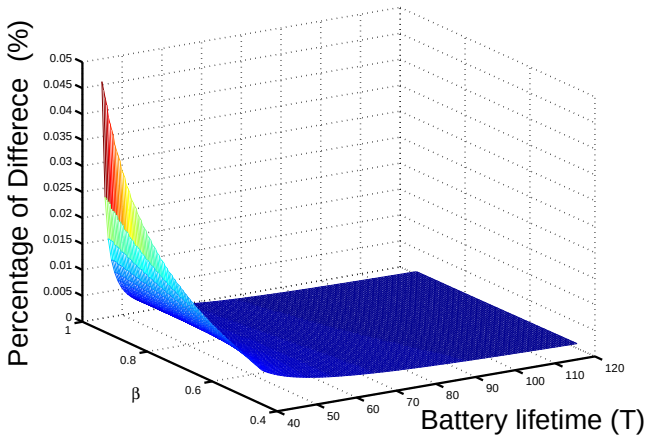
Fig. 2.2(a) verifies our observation, where we calculate $\frac{P(\beta,T)}{F(T,t_{begin},t_{end},\beta)}$ within $\beta \in [0.4,1]$ and $T \in [40min, 120min]$. It shows that the values of two functions are very close, where the difference is at most 4.5%. In Fig. 2.2(b) we use the data measured from a nickel-cadmium battery in pocket computers [10]. The discharge current I is 912mA for the first 25 min. After that the battery is recovered for 10min. We compare the two models in (2.1) and (2.6). As shown in Fig. 2.2(b), two models achieve very similar results. Thus, the capacity computation function (2.5) can accurately capture the battery behavior, and thus can be used to replace (2.2) for on-line computation.

Now we have simplified the battery model as in (2.6). We give an example to demonstrate how to use this on-line function to calculate battery lifetime. Given a battery with $\alpha = 100,000mAmin$, $I = 1,000mA$ and $\beta = 0.3$, we calculate its lifetime according to different ways of discharging it. (i) Assume this battery is discharged in its entire lifetime. In this case, $t_{begin} = 0$ and $t_{end} = T$ where T is the lifetime to be computed. Plugging these parameters to (2.6) and (2.5), we obtain its lifetime $T = 64min$. (ii) Assume we put this battery into recovery during [60*min*, 80*min*] then discharge it afterwards. In this case the battery is discharged in two durations: [0*min*, 60*min*] and [80*min*, T] where T is the lifetime to be computed. From the function

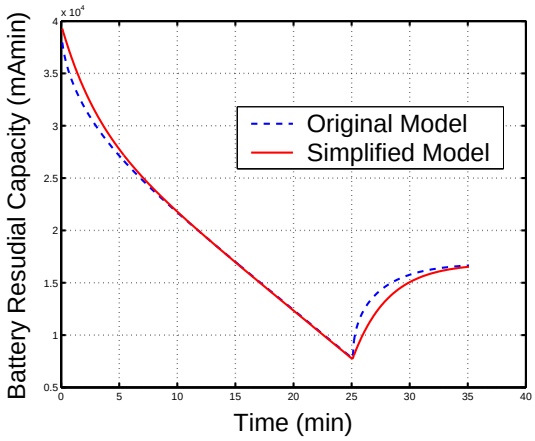
$$\begin{aligned}\alpha = \quad & I \quad \times \left[(60-0) + \frac{\pi^2}{3\beta^2}\left[e^{-\beta^2(T-60)} - e^{-\beta^2(T-0)}\right]\right] \\ + \quad & I \times \left[(T-80) + \frac{\pi^2}{3\beta^2}\left[e^{-\beta^2(T-T)} - e^{-\beta^2(T-80)}\right]\right]\end{aligned}$$

we can obtain $T = 94min$.

Now we further modify the battery model for discrete time modeling. As we discussed,



(a)



(b)

Figure 2.2: Evaluation of this battery model. (a) The comparison of two capacity computation functions (2.2) and (2.5). $\beta \in [0.4, 1]$ and $T \in [40min, 120min]$. Z axis is the ratio of the difference of the values computed by the two functions. (b) The comparison of the models in (2.1) and (2.6) for a pocket computer battery discharging process. The current is $I = 912m$A in the first 25min and the battery gets recovery in the next 10min. Y axis accounts for the battery residual energy, which increases during battery recovery.

due to the packetized nature of network communications, battery lifetime can be divided into a sequence of discrete time slots. Length of a time slot is δ. We use I_n and α_n to denote the discharge current through the battery and the dissipated battery energy in the n_{th} time slot, respectively. In the n_{th} time slot the battery is either in discharging ($I_n > 0$) or idle ($I_n = 0$). A time slot is called a *discharging slot* when the battery is in discharging. Without loss of generality, we assume that the discharge current I is a constant in a discharging slot. Fig. 2.3(a) shows an example of the discrete time battery model. Slots $1, 2, 3$ and 6 are discharging slots. The battery is idle during slots 4 and 5. According to (2.5) and (2.6), there are two types of energy dissipated in a discharging slot: the energy consumed in the device and the discharging loss. We define ζ_n as the battery discharging loss consumed in the n_{th} time slot $[n\delta, (n+1)\delta]$ for $n \geq 1$. That is,

$$\zeta_n = I_n \times \frac{\pi^2}{3\beta^2} \left[e^{-\beta^2 (T - n\delta)} - e^{-\beta^2 (T-(n-1)\delta} \right] \tag{2.10}$$

ζ_n is consumed in the n_{th} slot and recovered step by step in the following $n+1, n+2, \ldots$ slots until the battery dies. Clearly, ζ_n decreases as time goes by. Fig. 2.3(b) shows the decrease of ζ_1. Furthermore, we define ζ_n^{κ} as the ζ_n after next κ (≥ 1) slots, that is,

$$\begin{aligned} \zeta_n^{\kappa} &= I_n \times \frac{\pi^2}{3\beta^2} \left[e^{-\beta^2 ((n+\kappa)\delta - n\delta)} - e^{-\beta^2 ((n+\kappa)\delta - (n-1)\delta)} \right] \\ &= I_n \times \frac{\pi^2}{3\beta^2} \left[e^{-\beta^2 \kappa \delta} - e^{-\beta^2 (\kappa+1)\delta} \right] \end{aligned} \tag{2.11}$$

The recovery activity of a discharging slot is independent of those of other time slots. For example, in the 3_{rd} slot in Fig. 2.3, there are 2 discharging loss, ζ_1^1 and ζ_2^0, being recovered at the same time. It should be mentioned that discharging loss ζ_n is only a potential type of energy. In Fig. 2.3, if the battery dies at t, energy ζ_1^4, ζ_2^3 and ζ_3^2 do not

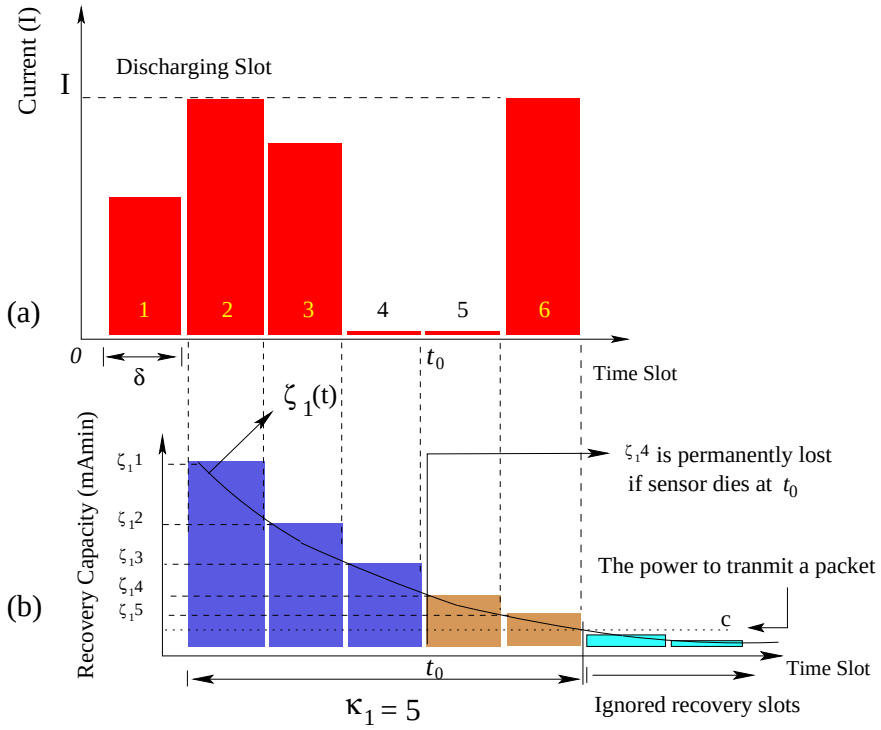


Figure 2.3: Discrete time battery model with slot length δ. (a) Slots $1, 2, 3$ and 6 are discharging slots. The battery is idle during slots 4 and 5. (b) ζ_1^i is the discharging loss of slot 1 at the i_{th} slot. As shown in the figure, if this battery dies at t_0, ζ_1^4 is permanently lost.

have a chance to be recovered. Thus, the battery permanently loses the energy.

In order to implement our model in energy sensitive wireless devices, we give a method to further simplify the computation of ζ_n. Assume that the energy for transmitting a single packet is c. By observing (2.11), we know that if ζ_n^κ is less than c, then the recovery of ζ_n after κ slots can be ignored. That is,

$$I_n \times \frac{\pi^2}{3\beta^2}\left[e^{-\beta^2\kappa\delta} - e^{-\beta^2(\kappa+1)\delta}\right] < c \tag{2.12}$$

From (2.12) we obtain

$$\kappa > \frac{1}{\beta^2\delta}\log\frac{\pi^2(1-e^{-\beta^2\delta})}{3\beta^2 c/I_n} \tag{2.13}$$

Therefore, we can truncate recovery slots and only look ahead limited $\kappa = \lceil \frac{1}{\beta^2\delta}\log\frac{(1-e^{-\beta^2\delta})\pi^2 I_n}{3\beta^2 c}\rceil$ time slots to calculate the approximate value of ζ_n, where δ is the slot length and I_n is the discharging current in the n_{th} slot. κ is referred to as the *recovery length*. In the example in Fig. 2.3(b), the recovery length of slot 1 is from slot 2 to slot 6 ($\kappa = 5$). The recovery of ζ_1 after slot 6 is ignored.

In fact, (2.13) also gives a way to compute ζ_n on-line. Given a battery, β is known, and δ is pre-determined. Hence we can simply pre-compute $\frac{1}{\beta^2\delta}\log\frac{(1-e^{-\beta^2\delta})\pi^2}{3\beta^2 c}$ off-line and take I_n on-line for the computation.

In order to guarantee $\kappa \geq 0$ in (2.13), we let

$$\frac{\pi^2(1-e^{-\beta^2\delta})}{3\beta^2 c/I_n} \geq 1 \tag{2.14}$$

Thus we obtain a lower bound on the length of the time slot δ

$$\delta \geq \frac{1}{\beta^2}\log\left(1-\frac{3\beta^2 c/I_n}{\pi^2}\right)^{-1} \tag{2.15}$$

As long as δ satisfies its lower bound $\lceil \frac{1}{\beta^2}\log(1-\frac{3\beta^2 c}{\pi^2 I_n})^{-1}\rceil$, (2.13) and (2.15) guarantee that recovery can be truncated to limited κ time slots.

κ depends on the battery diffusion parameter β, time slot length δ and the discharging current I. We calculated κ and δ for various values of β and I, and the results are shown in Table 1. As can be seen, the larger the β, the faster diffusion of the battery, and the fewer

Table 1: Relationship between battery diffusion β, battery current I, effective recovery length κ and slot length δ ($c = 0.3mAhr$)

	I_n	$\delta = 5$min	$\delta = 10$min	$\delta = 15$min
$\beta = 0.7$	$300mA$	$\kappa = 2$	$\kappa = 1$	$\kappa = 1$
	$600mA$	$\kappa = 3$	$\kappa = 2$	$\kappa = 1$
	$900mA$	$\kappa = 3$	$\kappa = 2$	$\kappa = 1$
$\beta = 0.5$	$300mA$	$\kappa = 4$	$\kappa = 2$	$\kappa = 2$
	$600mA$	$\kappa = 5$	$\kappa = 3$	$\kappa = 2$
	$900mA$	$\kappa = 6$	$\kappa = 3$	$\kappa = 2$
$\beta = 0.4$	$300mA$	$\kappa = 7$	$\kappa = 4$	$\kappa = 3$
	$600mA$	$\kappa = 8$	$\kappa = 4$	$\kappa = 3$
	$900mA$	$\kappa = 8$	$\kappa = 5$	$\kappa = 3$

effective recovery slots κ. Also, as the battery current increases, κ increases, while as the slot length δ increases, κ decreases. This can be interpreted as that with higher current, the battery works less like an ideal battery, therefore needs more time for recovery.

2.4 Battery-Aware Routing Scheme

Now we are in the position to apply the battery model to assist MANET routing. Although having advantages in flexibility and easy implementation, previous routing protocols are not aware of the status of batteries. The power metric used in these protocols considers only the radiation dissipation of the energy during routing. This is a rather rough metric and might not precisely model the energy dissipation in MANETs. In this section we first introduce a power metric for battery-aware routing. Based on the metric we develop the battery-aware routing (BAR) scheme and prioritized battery-aware routing (PBAR) scheme. Both BAR and PBAR are introduced to assist previous routing approaches

Table 2: Battery performance of various portable devices

	C	$I(m\text{A})$	β	$\zeta(m\text{Amin})$	ζ/C
Cellular Phone	$72,000$ mAmin	400	0.5	5600	7.79%
		600	0.5	8400	11.67%
Tablet PC	$100,000$ mAmin	2000	0.5	28000	28.00%
		2500	0.5	35000	35.00%
PDA	$120,000$ mAmin	1500	0.5	21000	17.50%
		1000	0.5	14000	11.67%
Laptop	$200,000$ mAmin	3500	0.5	49000	24.75%
		3000	0.5	41000	20.50%

rather than to replace them. They provide a battery-aware mechanize for upper layer routing protocols.

2.4.1 Power Metric for Battery-Aware Routing

There has been much work discussing the power metric for multi-hop wireless transmissions [4]. To transmit a packet from node a to node b in a MANET, the energy consumption is usually modeled as

$$E = e + \gamma d^n_{(a,b)} \tag{2.16}$$

where $d_{(a,b)}$ is the distance between node a and node b, n is the propagation loss coefficient, which is a constant determined by the transmission media, $\gamma d^n_{(a,b)}$ accounts for the radiated energy necessary to transmit over a distance of $d_{(a,b)}$, and e is the energy utilized in the transceiver such as CPU processing and memory flashing.

However, the metric in (2.16) is rather rough for battery-powered devices. As discussed in aforementioned sections, batteries over consume energy during discharging and recovery

the discharging loss if battery activity is scheduled well. To accurately model energy dissipation in battery-powered devices, discharging loss (ζ) should be included in the power metric for multi-hop wireless transmissions. We therefore introduce a more accurate energy consumption metric that includes discharging loss ζ

$$E = e + \gamma d^{n}_{(a,b)} + \zeta \tag{2.17}$$

For the energy dissipation in today's wireless devices, ζ is a significant amount of energy. We compare the battery performance of various widely used wireless devices in Table 2. Each device is assumed to be discharged at a current I during its entire lifetime. C is the total battery capacity and β is the battery chemical parameter. We assume these devices are normally discharged without considering their battery recovery. We calculated the ζ using our on-line computable battery model introduced in Section 2.2 and evaluated the ratio of discharging loss ζ/C for each device. The higher the ζ/C, the lower the energy efficiency it has. The results show that the battery discharging loss might be up to 35% of the total battery capacity. Note that ζ is not a constant. It depends on how a communication protocol schedules the battery activities. We believe energy-efficient routing should be battery-aware. A device in the network should know its battery status in order to carefully schedule the battery activity and budget energy consumption.

2.4.2 Battery-Aware Routing Scheme

In this subsection we present a battery-aware routing scheme (BAR) based on battery-aware metric. We assume that nodes are randomly deployed in a MANET. Each node knows its geographic position. Each node is powered by a battery with parameter β. We

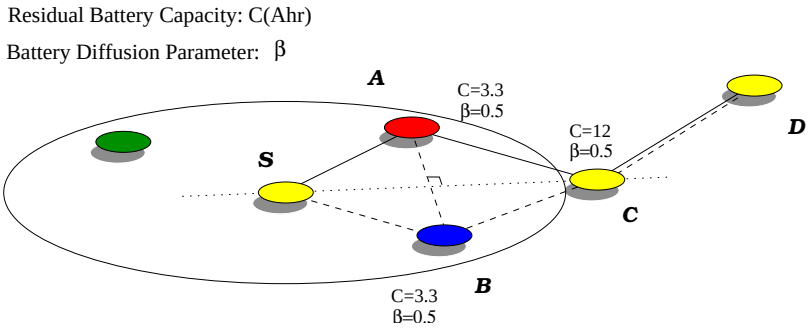


Figure 2.4: Battery-aware routing in a MANET. The current at each node is $I = 3.5$A. By switching between node A and node B, the network achieves longer lifetime.

also assume that the source node transmits a stream of packets to the destination. This models the applications such as video conferences or multimedia transmissions where transmission is viewed as a stream. A node is called a routing node if it is on a routing path from the source to the destination. The time is divided into discrete time slots with fixed length δ. In each time slot, a routing node can be assigned a task or idle. A task may be a routing activity, video displaying, software execution or any other energy consuming function at this node. Multiple tasks may be assigned in the same slot.

We first look at a simple example of battery-aware routing as shown in Fig. 2.4. In this MANET, source node S transmits packets to destination node D. The battery residual capacity C and parameter β are indicated in the figure. We compare the following two approaches.

In the first approach, S sends packets to D through a multi-hop path $S \rightarrow A \rightarrow C \rightarrow D$. 45.35 minutes later, node A uses up its energy. After that the routing path is changed to $S \rightarrow B \rightarrow C \rightarrow D$. The total connection lasts 90.7 minutes. However, the lifetime can be extended in a simple way by switching between the above two paths. In the second approach, node A and node B alternate each other as the router. A recovers its battery while

B is routing, and so on. This way the total lifetime is 113.15min which is 24.8% longer than that of the first approach.

We introduce the BAR scheme as described in Table 3. This scheme sets up a routing path from a source node to a destination node. At the very beginning source node calls the *Sender BAR Procedure*, all its one hop neighbors runs the *Receiver BAR Procedure*. After the first router is selected, this selected router node calls *Sender BAR Procedure* to select its next hop router. This repeats until the entire routing path is set up. A node needs to collect battery discharging loss status from all neighbors to find a next hop. In order to reduce the communication overhead BAR adopts κ instead of ζ to measure the battery discharging loss. Because κ is an integer and ζ is a floating number, adopting κ can reduce the packet size as well as the computational complexity. The larger the recovery length κ, the higher the discharging loss ζ, and the less the battery is recovered.

We introduce $\overrightarrow{R}$ and $\overrightarrow{C}$ to record battery status. Each routing node maintains these two vectors. $\overrightarrow{R}$ and $\overrightarrow{C}$ are the recovery status and the residual capacity at each time slot, respectively. For example a device is assigned 2 tasks, both begin from the first time slot. Tasks have recovery lengths as $\kappa_1 = 3, \kappa_2 = 4$. Hence $\overrightarrow{R} = [0,1,1,1,0,0] + [0,1,1,1,1,0] = [0,2,2,2,1,0]$. The value of $\overrightarrow{C}$ is typically as $\overrightarrow{C} = [500,420,375,390,405,410]$*(mAhr)*. The source also needs two vectors $\overrightarrow{R}'$ and $\overrightarrow{C}'$. $\overrightarrow{R}'$ and $\overrightarrow{C}'$ are the recovery vector and energy needed for this transmission, respectively. For example, if a transmission takes 2δ time, the recovery length $\kappa = 4$ and battery energy consumption is $50mAhr$ per time slot, then we have $\overrightarrow{R}' = [0,1,1,1,1,0] + [0,0,1,1,1,1] = [0,1,2,2,2,1]$ and $\overrightarrow{C}' = [50,50,0,0,0,0]$.

Along the routing path from the source to the destination, each hop is a sender and its next hop is the receiver. BAR is employed to assist routing protocols to select the next hop for a sender. In the table, we use four routing protocols, MFR, NFP, GEDIR and DIR,

as examples to show how BAR implement battery-aware routing. However, BAR is not restricted to these routing protocols. It is a scheme suitable for general routing protocols. On receiving a RTS (Request to Send) packet, available receivers reply a CTS (Clear to Send) packet. Their $\overrightarrow{R}$ and $\overrightarrow{C}$ are included in CTS. The sender receives CTS from all its n_1 one-hop neighbors. It selects the best available node by checking the following two rules for each node $j = 1, \ldots, n_1$. Firstly $\overrightarrow{C}_j - \overrightarrow{C}' > 0$ must be satisfied to ensure the next hop will not use up its battery during the transmission. Secondly $\overrightarrow{R}_j + \overrightarrow{R}'$ is computed for each receiver. If there exits a node with minimal $\overrightarrow{R}_j + \overrightarrow{R}'$, it is selected as the next router. Otherwise we select the next hop according to different strategies of routing protocols as shown in the description of the BAR scheme in Table 3. In the *NextHopSelect procedure* BAR complements MFR, NFP, GEDIR and DIR protocols to achieve battery-awareness.

The communication complexity and time complexity of the BAR scheme depend linearly on the diameter of the network. Thus its complexity is $O(\sqrt{n})$, where n is the number of nodes in the network. The source node updates the routing path every t time, where t depends on the mobile speed of the nodes in the MANET and is a multiple of δ. The higher the speed, the larger the t. As will be seen in Section 2.5, adopting BAR can prolong network lifetime, increase data throughput and reduce energy dissipation.

2.4.3 Prioritized Battery-Aware Routing Scheme

In this subsection we enhance the battery-aware routing scheme to provide prioritized service. We call the enhanced scheme *prioritized battery-aware routing (PBAR)*. The key idea of PBAR is that the scheme allows routing connections to have higher priority to be set up on "well recovered" nodes than other low priority connections, and therefore provides differentiated routing service with prolonged network lifetime. The routing connections in

the network are assigned with different priorities from 1 to q with q as the highest priority. PBAR let the receivers make the decision whether or not to accept a routing request by considering its priority.

With a higher priority, a transmission source can set up a path to its destination more quickly on well recovered nodes. Along the routing path from the source to the destination, each hop is a sender and its next hop is the receiver. Once a path is set up, the path is reserved for this connection for a relatively long time t. Each connection is associated with a priority. PBAR scheme is also independent of specific routing protocols. However, considering the time sensitiveness of setting up a prioritized routing path, MFR protocol is more suitable for PBAR. This is because that, on one hand, more computation is required if adopting GEDIR or DIR protocols, and on the other hand, routing paths set up by NFP protocol generally contain more routing hops than those set up by MFR protocol. Having more hops in a routing path naturally implies longer routing delay which is not preferred in high prioritized communication. Therefore, we adopt MFR protocol in PBAR scheme. If not specially mentioned, in the rest of this chapter we use PBAR to stand for PBAR-MFR.

During path setting up, if node i sends a RTS_i to select its next hop, the priority $priority_i$ is included in its RTS_i packet. The details of the PBAR scheme is described in Table 4 and Table 5. The scheme is composed of two procedures: the sender PBAR procedure and the receiver PBAR procedure. In the receiver procedure, a receiver r keeps receiving RTS_k ($k = 1, \ldots, n_2$), which request to set up a routing path through r. The n_2 RTS requests are collected in a short period of $\bar{t}$ time. The receiver sorts RTS_is in a Priority Set by their priorities and replies to the highest priority inquiry. In the sender procedure, senders use two different algorithms depending on different priorities. To set up a low priority path ($priority < q$), a sender first calls *Low Priority Service* in "Sender PBAR" module. It sends

out RTS to potential candidate nodes. Then it collects replies of accepts from receivers into a Hop Queue. Among these receivers it selects the most feasible one with best battery status. If no receiver accepts this sender's query, the sender has to wait for a random time and repeats from the beginning. The premium path ($priority = q$) does not need to wait for the receiver's confirmation, because its inquiry is always the highest priority inquiry in the receiver's set. In the PBAR scheme, the sender simply calls the *Premium Service* in "Sender PBAR" module.

Fig. 2.5 gives an example of applying the PBAR scheme in a MANET. There are two routing paths to be set up from S_1 and S_2 to D_1 and D_2, respectively, with $priority_1 > priority_2$. Because node C has a better battery status than node E, node C is at the top of Hop Queue in both A and B. Node C accepts the highest priority request of node A. Thus node B has to choose the next available node in its Hop Queue, in this case node E, as its next hop. As can be seen, by reserving resource, PBAR can statistically provide high performance for high priority connections.

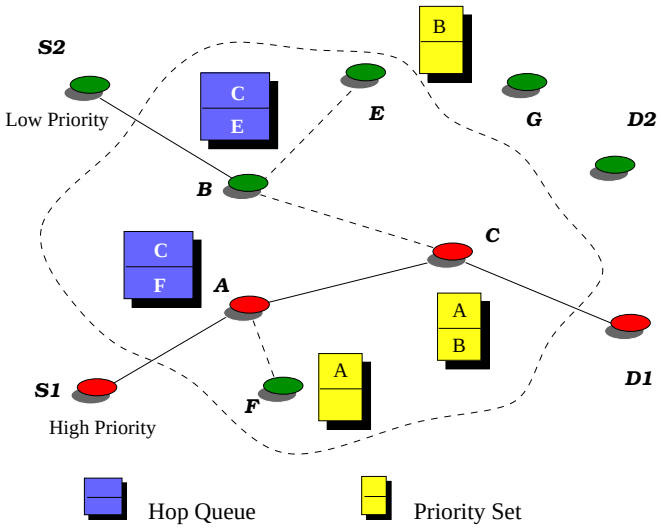


Figure 2.5: An example of PBAR providing differentiated routing service.

2.5 Performance Evaluations

In this section we evaluate the performances of BAR and PBAR in terms of network lifetime, data throughput, routing delay and energy dissipation.

2.5.1 Simulation Setup

In the simulation, we assume that wireless devices are randomly deployed in a 150×150 field. The transmission radius of each device is 15. The devices are mobile, and we assume that each device moves in a random direction at the speed of 0.1 per *min*. The length of a time slot δ is set to $10min$. Each device randomly generates packets and routes them to a specific destination. Each packet has its priority p if applied. A source stops transmitting when no routing path can be set up between the source and its corresponding destination.

To model real-world applications, network nodes are composed of several wireless portable devices, including cellular phones, laptops, PDAs and tablet PCs. We collected their full battery energy values (α) and battery parameters (β) from actual mobile devices. The full battery energy values of cell phone, PDA, Tablet PC and Laptop are $72,000mA$min, $120,000mA$min, $100,000mA$min and $200,000mA$min, respectively. Their battery parameters range from 0.4 to 0.6. We use device composition to indicate the percentage of different mobile devices in a randomly generated network. We will evaluate the lifetime, data throughput and energy dissipation of MANETs that have different compositions, distinct device types, various node densities and various batteries with different α and β values to comprehensively evaluate the performance of BAR scheme.

2.5.2 Performance Evaluations of BAR Scheme

We compare the performance of aforementioned protocols employing BAR scheme with those without BAR scheme. We first evaluate the number of alive nodes, energy dissipation and data throughput for a network with 200 devices. We adopt a randomly generated composition of 3% cell phones, 7% PDAs, 35% tablet PCs and 55% laptops. Each device has a battery parameter $\beta = 0.5$.

Number of Alive Nodes. We first evaluate the number of alive nodes during the network lifetime. By adopting BAR scheme, routing protocols tend to leave batteries that have high discharging loss in recovery. Therefore BAR enables nodes to use up battery energy gradually. A network with BAR scheme has more alive nodes compared with the network without it. Fig. 2.6 shows that the number of the nodes in the network decreases as transmissions go on. We can see that since the battery-aware scheme is sensitive to battery status and carefully recovers batteries, the decrease of the number of alive nodes is slower when employing BAR. Also note that the network lifetime is extended as well. The simulation shows that the lifetime can be prolonged by up to 28%. Next we study its energy dissipation performance.

Energy Dissipation. We now evaluate the energy dissipation of the network. When not using BAR, a routing node tends to use up its battery energy without recovery. Then the routing path is switched to another alive node. This will cause several problems. First, the residual battery energy of each node is very different. Some nodes almost use up their energy, while other nodes are left with full battery energy. Second, the residual energy at nodes is very low because discharging losses of batteries are not recovered. Third, routing paths in the network tend to be re-set up more frequently, because a path has to be disconnected even if only a single router on it uses up its energy. The operations of detecting

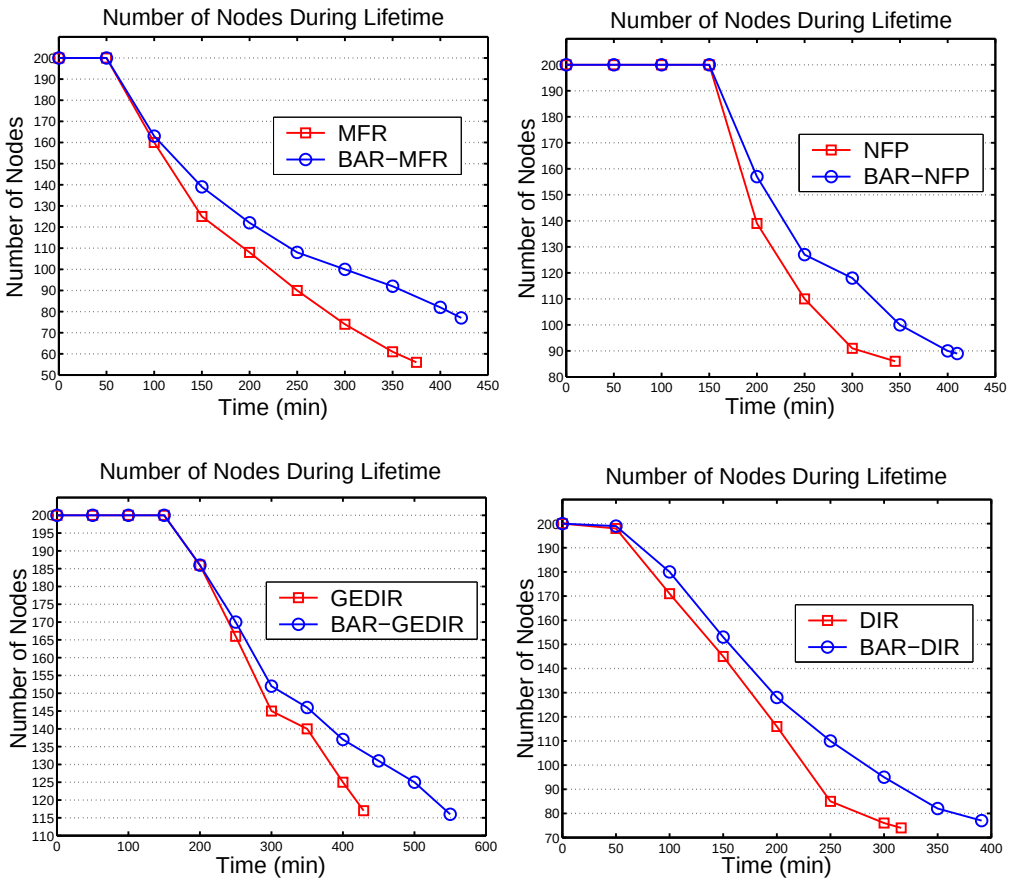


Figure 2.6: The number of nodes decreases during the lifetime of the network. With BAR scheme, the decrease is slower.

disconnected paths and re-setting up a transmission consume much extra energy. As can be seen, by adopting BAR scheme, the energy consumed by nodes is more uniform: the residual energy at each node is almost at the same level. Fig. 2.7 shows the energy distribution of the nodes in the middle of network transmission (at the $200_{th}min$). The X axis and Y axis show the geographic positions of nodes in the network. The Z axis stands for the residual battery energy of nodes. It can be seen that by adopting BAR, nodes can preserve higher battery energy. The simulation results show that the average battery energy of a node can be increased by up to 45% compared with the case without BAR. It can also be observed

that the alive nodes are distributed more uniformly under BAR scheme.

Data Throughput. BAR improves the data throughput as well. Two nodes cannot set up a routing path if some routing nodes between them die. BAR scheme keeps more alive nodes therefore achieves larger gross data throughput during the network lifetime. Fig. 2.8 compares the gross data throughput with different protocols. We can see that the improvement achieved by BAR can be up to 24%. We also observe that BAR-GEDIR achieves the highest data throughput. This is because GEDIR protocol uses the least number of nodes to set up a routing path compared with other routing protocols. It therefore achieves the longest lifetime as well as largest data throughput.

Next we evaluate network performance for various densities, different devices, compositions, α and β values.

Network Lifetime with Various Node Densities. In Fig. 2.6, we have seen that the network lifetime can be prolonged by BAR. Now we evaluate the network lifetime with various node densities. In our simulation we set up networks with 100, 150, 200, 250 and 300 nodes deployed. For consistency the network composition is still of 3% cell phones, 7% PDAs, 35% tablet PCs and 55% laptops. Each device has a battery parameter $\beta = 0.5$. Fig. 2.9 shows the network lifetimes with different node densities. We can see that battery-awareness can greatly increase the network lifetime. Also note that the rate of lifetime increase is higher with lower node density. For example, in the comparison between DIR and BAR-DIR, the lifetime can be increased by 44.4% with 100 nodes, and increased by 13.8% with 300 nodes. This is because that a network with lower density is more likely to have an insufficient number of nodes as routers to construct routing paths. BAR can carefully budget node energy dissipation and preserve more alive nodes. Therefore, a lower density network benefits more from such battery energy saving, and its rate of

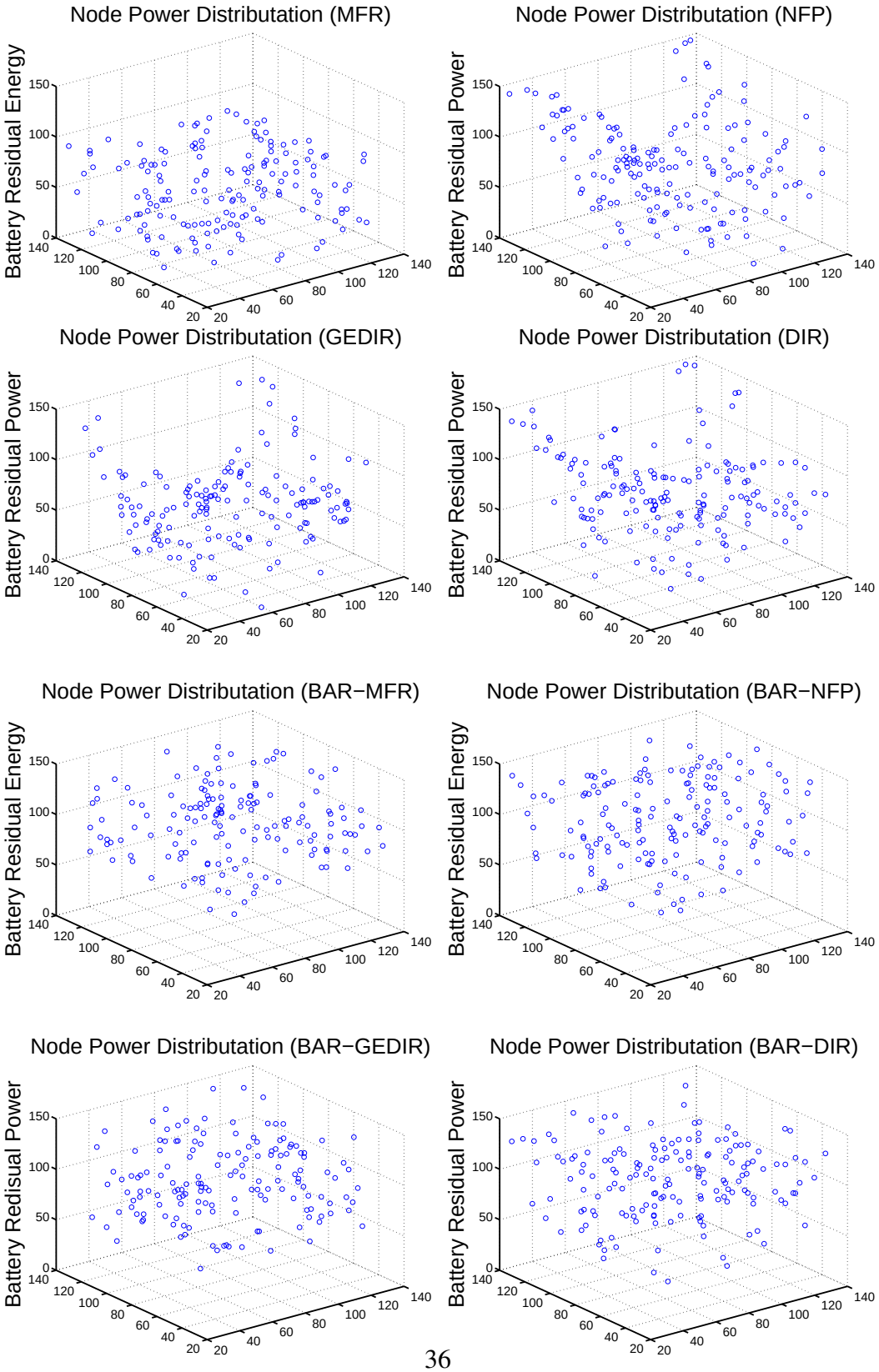


Figure 2.7: The distribution of node power in the middle of network transmission with different protocols. Z is residual battery power at routing nodes.

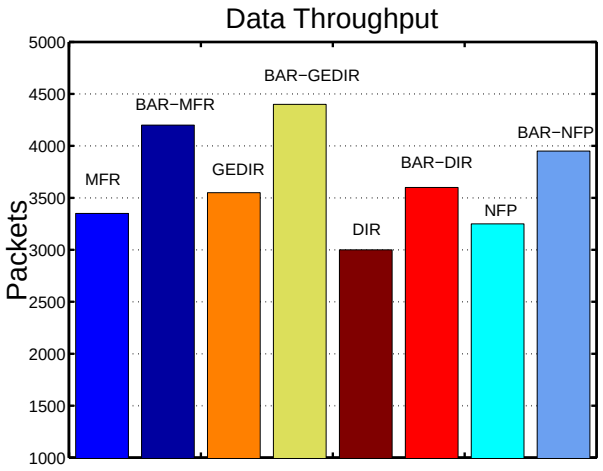


Figure 2.8: The gross data throughput of different protocols during their network lifetime.

lifetime increase is higher than that of a higher density network.

Network Lifetime with Different Devices. As mentioned earlier, laptops, PDAs, cellular phones and tablet PCs are typically used battery-powered wireless devices. Different devices come with different batteries, which have different discharging current (I), battery capacities (α) and battery parameters (β). To see how BAR can enhance network performance for a specific type of device, we evaluate the performance of four networks, each of which is composed of one type of device among cellular phones, PDAs, tablet PCs and laptops. Their full battery energy values (α) are listed in Section 2.5.1. We assume their battery parameters (β) are 0.5. Each network has 200 devices. Fig. 2.10 shows the lifetimes of each network with various protocols. From the figure we observe that overall BAR increases the lifetime of all types of devices. Clearly, the network of laptops achieves the longest lifetime as laptops have the largest battery capacity ($200,000 mAmin$) among all devices.

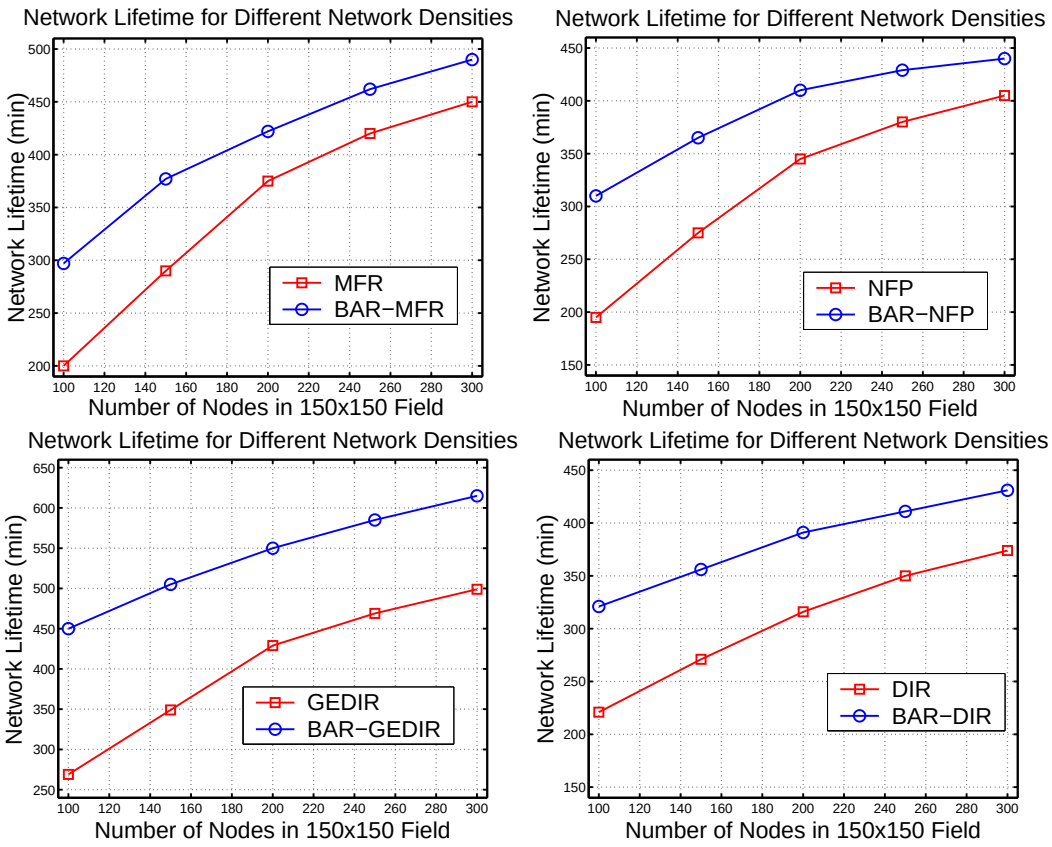


Figure 2.9: Network lifetime with different node densities.

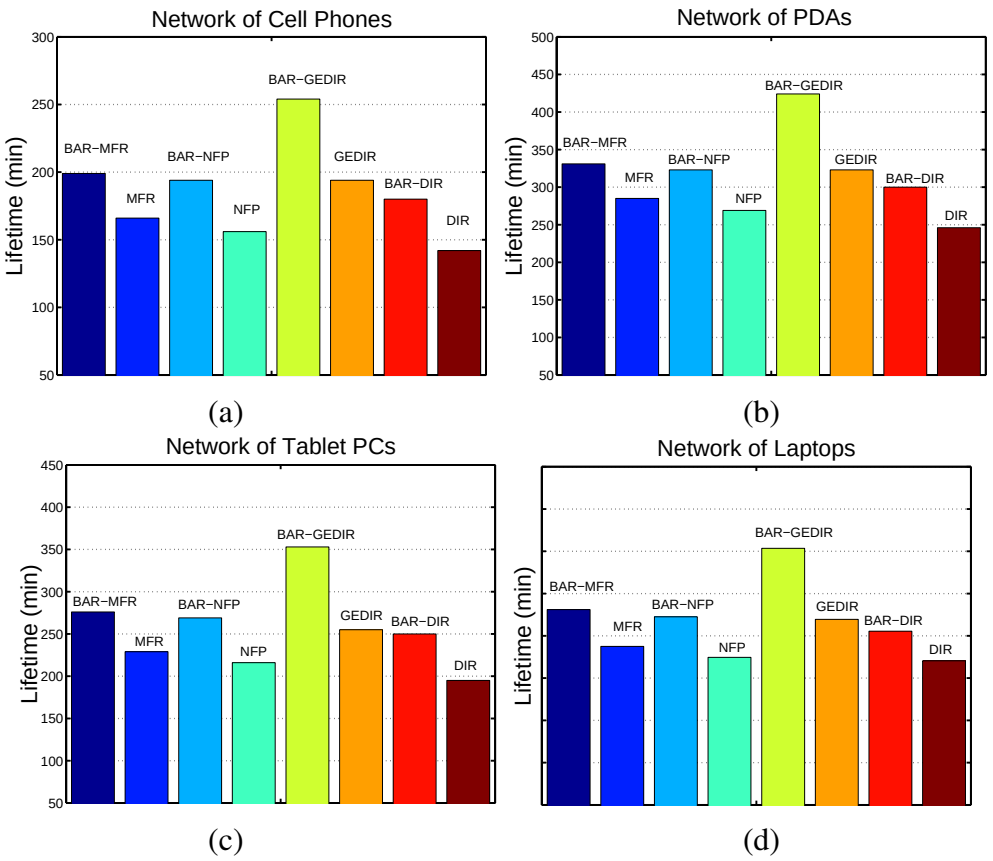


Figure 2.10: The lifetimes of networks with different devices: (a) 200 cellular phones; (b) 200 PDAs; (c) 200 tablet PCs; (d) 200 laptops.

Network Lifetime with Different Device Compositions. In Fig. 2.6, we used a network with a randomly generated composition to evaluate BAR scheme. To study its performance in more general cases we now compare networks with different compositions. The networks in Fig. 2.11 (a), (b), (c) and (d) have 200 wireless devices each. Their compositions are different from each other: network in (a) has 28% cellular phones, 24% PDAs, 26% tablet PCs and 22% laptops; network in (b) has 43% cellular phones, 17% PDAs, 19% tablet PCs and 21% laptops; network in (c) has 12% cellular phones, 54% PDAs, 11% tablet PCs and 23% laptops; network in (d) has 25% cellular phones, 25% PDAs, 25% tablet PCs

and 25% laptops. The compositions of the four networks vary significantly so that we can evaluate the performance of BAR for various cases. In Fig. 2.11 we observe that BAR prolongs network lifetimes in all four sets, with the most by up to 28% percentage. Among them network (c) achieves the longest lifetime. This is because that this network contains devices that have more battery capacities.

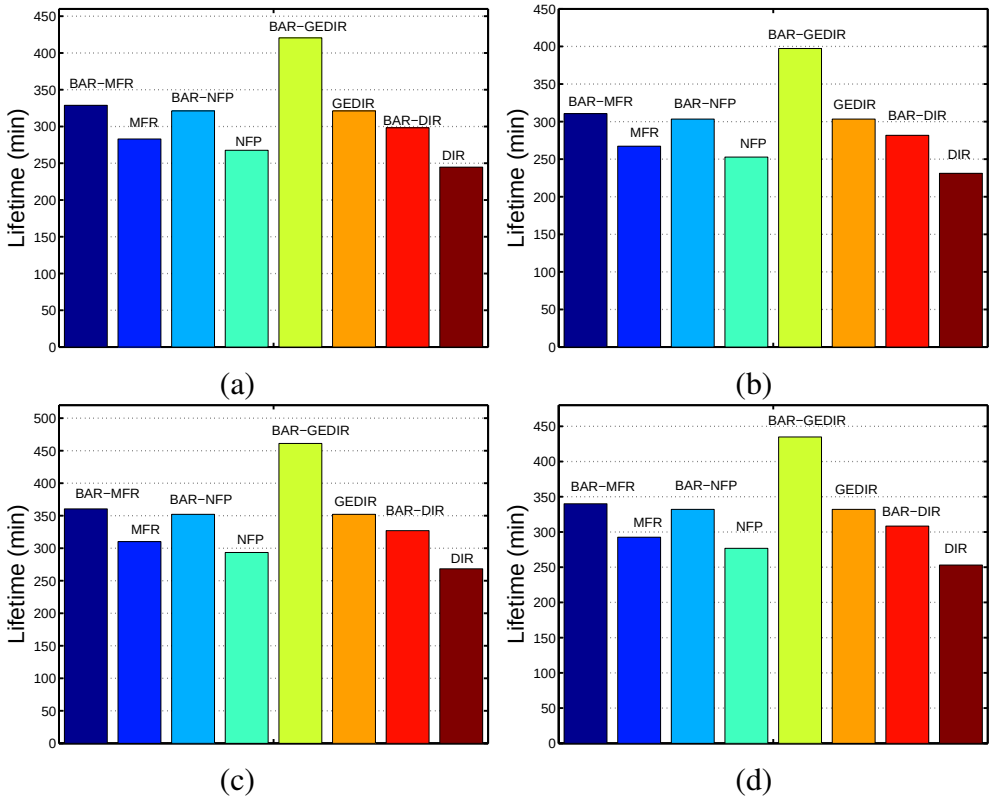


Figure 2.11: The lifetimes of networks with different devices compositions. (a), (b), (c) and (d) are four networks of 200 nodes. Each network contains different percentages of various wireless devices.

Effects of Different α and β values on Network Lifetime. In this part of the simulation, we study the effects of different battery capacities (α) and different battery parameters (β) on network lifetime. We first consider the battery capacity α. We compare two

networks A and B, each contains 200 devices. The devices in the two network, A and B, are equipped with batteries with large capacity $\alpha = 300,000 mAmin$ and small capacity $\alpha = 100,000 mAmin$, respectively, as shown in Fig. 2.12 (a) and (b). We evaluate the decrease of the number of alive nodes with MFR and BAR-MFR. The results show that BAR scheme greatly enhances network lifetimes in both cases. We then compare two networks C and D, each contains 200 devices which are equipped with batteries with large parameter $\beta = 0.6$ and small parameter $\beta = 0.4$, respectively, as shown in Fig. 2.12 (c) and (d). The results show that BAR-MFR in both networks achieves almost equal lifetimes, while network lifetime with MFR in (d) is shorter than MFR in (c). This is because the smaller the β, the larger the discharging loss. Batteries with smaller β values give BAR scheme more room to improve their lifetimes.

2.5.3 Performance Evaluations of PBAR Scheme

In this subsection we evaluate the performance of the PBAR scheme. We compare network performance among routing protocols with and without adopting PBAR. In the simulations, devices are deployed in the 150×150 area. The number of devices in a network ranges from 50 to 400. We adopt a composition of 3% cell phones, 7% PDAs, 35% tablet PCs and 55% laptops. Each has $\beta = 0.5$. The performance evaluated here are energy dissipation, end-to-end packet delay and data throughput.

Energy Dissipation. We first evaluate a 200 nodes network. Fig. 2.13 shows the energy distribution of the nodes with PBAR in the middle of network transmission (at the $45 thmin$). The X and Y axes show the geographic positions of nodes in the network. The Z axis stands for the residual battery energy of each alive device. As can be seen, by adopting PBAR scheme, the energy consumed at nodes are generally more uniform, that is, the

residual energy at each node is almost at the same level. Our results show that the average battery energy on a node is increased by up to 39.5% compared with existing protocols at the $45thmin$. It also shows that the residual energy of alive nodes are distributed more uniformly under PBAR protocol.

End-to-End Delay. To see the performance of end-to-end routing delay with PBAR, we compare the packet round trip time of routing connections with different priorities. A source generates a packet to send to a destination on the other side of the network. As soon as a packet is successfully delivered to the destination, an ACK packet is returned to the source. We use the round trip time (rtt) to measure the end-to-end delay. rtt is calculated as the length of the time period from the time the packet is sent to the time the ACK is received. In the simulation, we consider two cases.

In the first case, there are two routing priorities in the network: the high priority and the low priority. In the simulation, the routing connections choose their priorities randomly, and encapsulate the priority information into the head of each packet. We compare the average rtt of routing connections with different priorities. To evaluate the performance of our scheme, we also conducted simulations for networks with various numbers of initial wireless nodes: $50, 100, 200, 300$ and 400. Fig. 2.14(a) shows the packet rtt against the different number of initial nodes in the network. The $PBAR(high)$ and $PBAR(low)$ stand for high priority and low priority routing connections, respectively. We compare them with $PBAR(no\ priority)$, which is the average rtt when all packets in the network have no priority. As can been seen in Fig. 2.14 (a), $PBAR(high)$'s rtt is much lower than $PBAR(no\ priority)$, which indicates that PBAR can provide prioritized routing service for high priority connections. The higher priority a routing connection has, the lower its routing delay. Also, we note that the rtt for the low priority is reduced a little as the number of

nodes in the network is increased. This is due to that there are more available routing nodes when more nodes in this network can recovery their battery well.

In the second case, we show that PBAR can always provide stable *rrt* for high priority routing connections. We increase the number of priorities to three different levels: high, middle and low, in Fig. 2.14 (b)). We can see that although there are more priorities, the *rtt* for the highest priority is not affected. This is because that PBAR always reserves the resource for the highest priority. However, the lowest priority's *rtt*, compared with the one in the first case, is increased.

Data Throughput. PBAR improves the data throughput as well. This simulation consider two cases. In the first case, we compare the successfully transmitted packets among different protocols in Fig. 2.15 (a) and (b). The Y axis is the total number of the successfully transmitted packets during the network lifetime. We can observe that the data throughput of PBAR is higher. This is because the lifetime is prolonged by PBAR and it in turn increases the data throughput. Also, by adopting PBAR scheme there are more alive nodes in the network. Thus it is more likely to successfully set up routing paths between source-destination pairs. Fig. 2.15 (a) and (b) show that PBAR achieves better performance. The data throughput with PBAR is improved by up to 42.7%. In the second case, we compare the throughput of different routing priorities in Fig. 2.15 (c) and (d). According to PBAR, a connection with a higher priority has greater probability to set up and reserve a routing path, therefore has larger data throughput than lower priority routing connections. We conducted two sets of simulations in (c) and (d) with 150 and 250 nodes initially deployed in the network, respectively. The results in both Fig. 2.15 (c) and (d) verify that the PBAR increases the data throughput for high priority routing connections.

2.6 Summaries

In this chapter, we have addressed the issue of achieving energy efficiency in MANET routing by adopting a novel battery-aware energy model. We proposed an on-line computable discrete time analytical model to mathematically model battery discharging behaviors. We presented a battery-aware routing scheme (BAR) to facilitate battery-awareness in routing protocols to achieve energy efficiency. BAR scheme is independent of specific routing protocols. It is sensitive to the battery status of routing nodes to avoid unnecessary energy loss. We then applied this battery model to prioritized battery-aware routing and gave PBAR scheme. We implemented both BAR and PBAR schemes to evaluate their performance. We considered networks of different device compositions, device types, batteries with various α and β values, and different node densities. Our simulation results show that by adopting our schemes much better performance can be achieved in terms of network lifetime, data throughput, end-to-end delay and energy dissipation.

Table 3: Battery-aware routing scheme (BAR)

Receiver BAR procedure:
begin
  Update $\overrightarrow{R}$ and $\overrightarrow{C}$;
  Receive RTS from sender; Reply CTS with $\overrightarrow{R}$ and $\overrightarrow{C}$;
  if selected as router **then**
    Receive $\overrightarrow{R}'$ and $\overrightarrow{C}'$ from sender;
    $\overrightarrow{R} = \overrightarrow{R} + \overrightarrow{R}'$; $\overrightarrow{C} = \overrightarrow{C} - \overrightarrow{C}'$;
    Call {*Sender BAR procedure*}; // *Selecting its next router*
   end if
end

Sender BAR procedure:
begin
  Hop Queue = {}; Broadcast RTS;
  Receive $\overrightarrow{R_1}, \overrightarrow{R_2}, \ldots, \overrightarrow{R_{n_1}}, \overrightarrow{C_1}, \overrightarrow{C_2}, \ldots, \overrightarrow{C_{n_1}}$ from n_1 neighbor nodes;
  for $j = 1 \ldots n_1$ **do**
    if $\overrightarrow{C_j} - \overrightarrow{C}' > 0$ and j is in forward direction **then**
      Hop Queue = Hop Queue$\cup\{j\}$; **end if**
  end for
  Call {*BAR NextHopSelect procedure*};
end
Send $\overrightarrow{R}'$ and $\overrightarrow{C}'$ to the selected nexthop;

BAR NextHopSelect procedure:
begin
  for each node j in Hop Queue **do** nexthop = j with min$\{\| \overrightarrow{R}_j + \overrightarrow{R}' \|\}$;
    if the minimum is not unique **then** // *Adopting different strategies:*
      if *BAR-MFR* **then** nexthop = j with max{distance from j};
      if *BAR-NFP* **then** nexthop = j with min{distance from j};
      if *BAR-GEDIR* **then** nexthop = j with min{distance of j and destination};
      if *BAR-DIR* **then** nexthop = j with min{angle of j and destination};
    end if
  end for
end

Table 4: Prioritized battery-aware routing scheme(PBAR), receiver procedure

Receiver PBAR procedure:
begin
  Update $\overrightarrow{R}$ and $\overrightarrow{C}$;
  Priority Set = $\{\}$, $k = 0$;
  repeat
    Receive RTS$_k$;
    Priority Set = Priority Set $\cup$ $\{$RTS$_k\}$;
    $k = k + 1$;
  until time $\bar{t}$ has passed or $priority(k) = p$;
  repeat
    Reply CTS$_i$ to i to accept, where $priority(i) =$
     $\max\{priority(k) | k \in$ Priority Set$\}$;
    if not receive reply from i **then**
      Priority Set = Priority Set $-\{i\}$;
  until i selects this node as i's next router;
  Receive $\overrightarrow{R}'_i$ and $\overrightarrow{C}'_i$ from i;
  Deny all other nodes in Priority Set except i;
  $\overrightarrow{R} = \overrightarrow{R} + \overrightarrow{R}'_i$;
  $\overrightarrow{C} = \overrightarrow{C} - \overrightarrow{C}'_i$;
  // Selecting the receiver's next hop
  if priority$(i) < q$ **then**
    Call $\{$*Sender PBAR Low Priority service*$\}$;
   elseif priority$(i) = q$ **then**
    Call $\{$*Sender PBAR Premium service*$\}$;
  end if
end

Table 5: Prioritized battery-aware routing scheme(PBAR), sender procedure

Sender PBAR procedure:
Low Priority Service (*For Priority*$< q$):
begin
while the next hop not selected **do**
Hop Queue = $\{\}$;
Broadcast RTS;
while not all nodes in Hop Queue deny request **do**
if j accepts and $\overrightarrow{C_j} - \overrightarrow{C}' > 0$ **then**
Hop Queue = Hop Queue$\cup\{j\}$; **end if**
end while
if Hop Queue is empty **then**
Wait for a random time;
else
Select node j from Hop Queue with $\min\{\| \overrightarrow{R}_j + \overrightarrow{R}' \|\}$;
if the minimum node is not unique **then**
nexthop = j with $\max\{$distance from $j\}$;
else nexthop = j; **end if**
end if
end while
Send $\overrightarrow{R}'$ and $\overrightarrow{C}'$ to nexthop;
end

Premium Service (*For Priority*$= q$):
begin
Hop Queue = $\{\}$;
Broadcast RTS;
Receive $\overrightarrow{R_1}, \overrightarrow{R_2}, \ldots, \overrightarrow{R_{n_1}}, \overrightarrow{C_1}, \overrightarrow{C_2}, \ldots, \overrightarrow{C_{n_1}}$
from n_1 neighboring nodes;
for $j = 1 \ldots n_1$ **do**
if $\overrightarrow{C_j} - \overrightarrow{C}' > 0$ **then**
Hop Queue = Hop Queue$\cup\{j\}$; **end if**
end for
Select node j from Hop Queue with $\min\{\| \overrightarrow{R}_j + \overrightarrow{R}' \|\}$;
if the minimum node is not unique **then**
nexthop = j with $\max\{$distance from $j\}$;
else nexthop = j; **end if**
Send $\overrightarrow{R}'$ and $\overrightarrow{C}'$ to the selected nexthop;
end

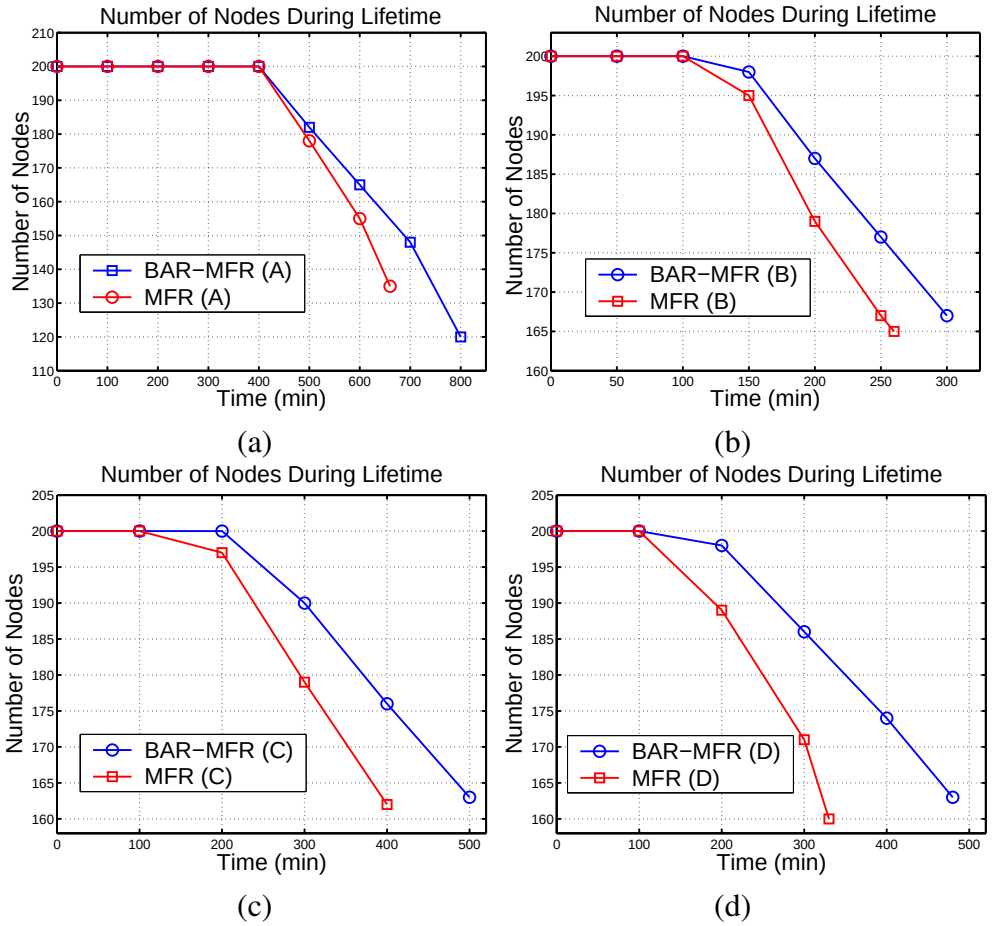


Figure 2.12: The effects of different α and β values on network lifetimes. (a), (b), (c) and (d) are four networks of 200 devices. They are equipped with different batteries: (a) batteries with large α $(300,000mAmin)$; (b) batteries with small α $(100,000mAmin)$; (c) batteries with large β (0.6); (d) batteries with small β (0.4).

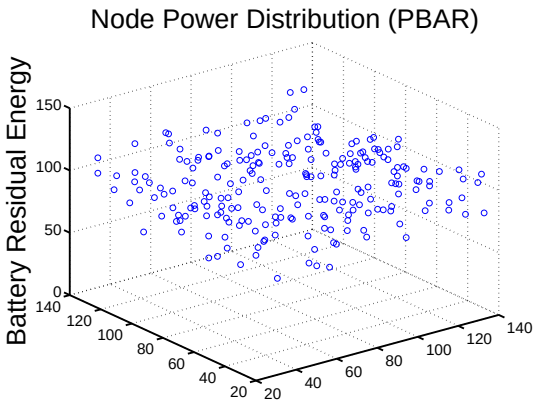
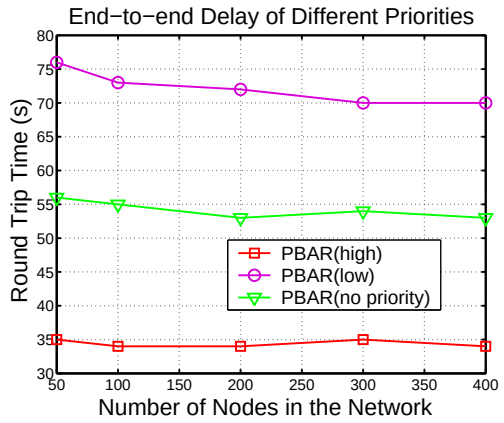
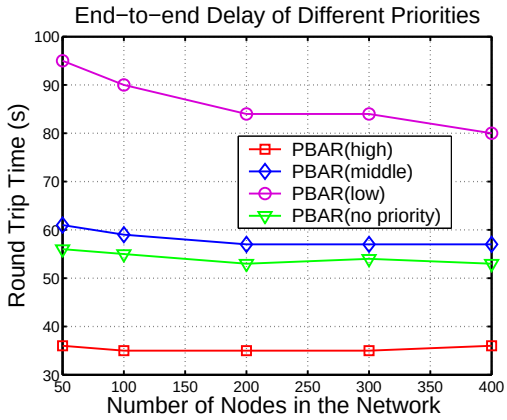


Figure 2.13: The residual battery energy of wireless devices at the 45*th* minute.



(a)



(b)

Figure 2.14: The average packet round trip time (rtt) of different priorities for various network sizes. (a) Two priorities: high and low. (b) Three priorities: high, middle and low.

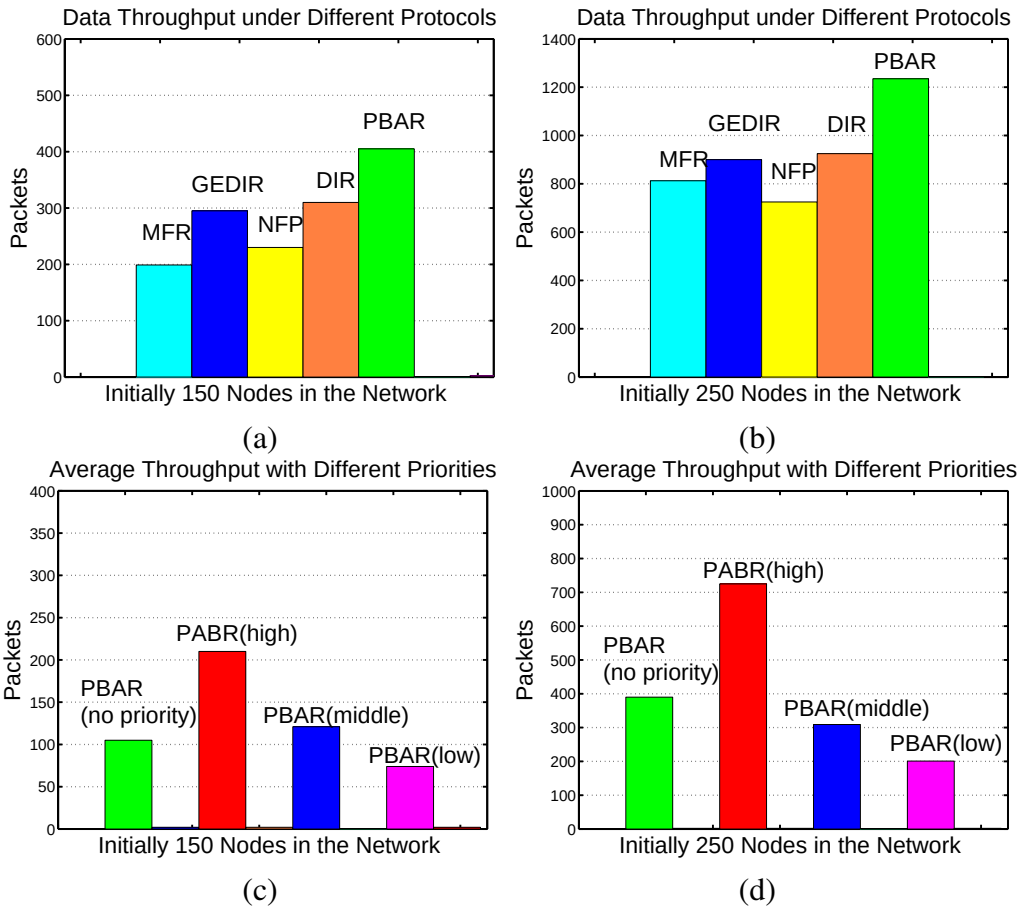


Figure 2.15: The data throughput during network lifetimes. (a) and (b): PBAR achieves the largest data throughput compared with existing protocols. (c) and (d): The average data throughput per connection with different priorities by PBAR.

Chapter 3

Battery-Aware Backbone Scheduling for WSN

This chapter presents battery-aware virtual backbones constructing algorithms for WSNs. In this chapter we first provide a mathematical battery model suitable for implementation in WSNs. We then introduce the concept of battery-aware connected dominating set (BACDS) and show that in general the minimum BACDS (MBACDS) can achieve longer lifetime than the previous backbone structures. Then we show that finding a MBACDS is NP-hard and give a distributed approximation algorithm to construct the BACDS. The resulting BACDS constructed by our algorithm is at most $(8+\Delta)opt$ size, where Δ is the maximum node degree and opt is the size of an optimal BACDS. The time and message complexities of the algorithm are $O(n)$ and $O(n(\sqrt{n}+\log n+\Delta))$, respectively, where n is the number of nodes in the network. Simulation results show that the BACDS can save a significant amount of energy and achieve up to 30% longer network lifetime than previous schemes.

The rest of the chapter is organized as follows. In Section 3.1 we discuss some background and related work to place our work in context. We present a simplified model suitable for sensor network applications in Section 3.2. Section 3.3 first gives the BACDS model along with its performance comparison with the MCDS model, then presents a distributed approximation algorithm to construct the BACDS. We also analyze the performance of the algorithm and give an upper bound on the size of the BACDS obtained in this section. Finally, we give simulation results in Section 3.4, and Section 3.5 gives the conclusion remarks of this chapter.

3.1 Related Work

In this section we will discuss the backbone routing in WSNs. The backbone scheduling problems are formalized as problems related to connect dominating set. We will review previous connect dominating set construction algorithms in this section.

3.1.1 Backbone Hierarchy in WSN

Wireless sensors are used to monitor or sense environment in many applications, such as medical treatment, outer-space exploration, battlefield surveillance, emergency response, etc. [37, 47, 48, 49, 50, 56, 58]. Although most current existing research on sensor networks is still at the prototype level, see, for example, UCB-smart dusts[51], MIT-μAMPS [52], ISI-pc104[54], UCLA-WINS[53], TmoteSky nodes[43], and Crossbow MICA[55], it is expected that sensor network technologies will be applied widely in various areas in the near future[37, 47, 48, 49, 50].

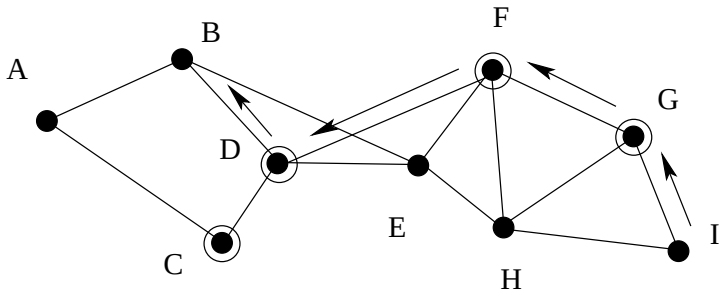


Figure 3.1: The minimum connected dominating set (MCDS) $\{C, D, F, G\}$ forms a backbone for the WSN. A packet is forwarded from node I to node B by the backbone.

A *sensing packet* is the reading of the event that triggers a sensor. A sensor node is composed of three components, *sensing component*, *routing component* and *control component*. In a sensor all these components can be turned on/off independently without affecting the rest of the sensor[46]. A WSN is usually expected to work for a long time, for example, sensors that are used to monitor wild animal habits are required to work for several years [37]. Sensors are powered by batteries[43, 44, 45, 52, 55]. Because sensors are deployed in unknown or tough areas, it is not easy to recharge or replace sensor batteries. It is very important to schedule WSNs in an energy-efficient way[57, 59]. A typical function of WSNs is to collect data in a sensing environment. Usually the sensed data in such an environment is routed to a sink, which is the central unit of the network [38]. Although a WSN does not have a physical infrastructure, a virtual backbone can be formed by constructing a *Connected Dominating Set (CDS)* [39, 40] in the network for efficient packet routing, broadcasting and data propagating. Fig. 3.1 shows a CDS for a given WSN. In this example, packets can be routed from the source (node I) to a neighbor in the CDS (node G), along the CDS backbone to a dominating set member (node D), which is closest to the destination (node B), and then finally to the destination.

In general, a WSN can be modeled as a graph $G = (V, E)$, where V and E are the sets of

nodes and edges in G, respectively. A CDS is a connected subgraph of G where all nodes in G are at most one hop away from some node in the subgraph. A node in the CDS is referred to as a *dominator*, and a node not in the CDS is referred to as a *dominatee*. Dominatees are put into sleep periodically to save their energy.

3.1.2 Previous MCDS Construction Algorithms

There has been a lot of work that dedicates to construct a *Minimum Connected Dominating Set (MCDS)* which is a CDS with a minimum number of dominators. Unfortunately, finding such an MCDS in a general graph was proven to be NP-hard [41]. So was in a *unit disk graph* (UDG) [42], where nodes have connections only within unit distance. Approximation algorithms for constructing CDS in WSNs have been studied by several researchers [34, 35]. The first algorithm reported in [35] is a greedy heuristic algorithm with bounded performance guarantees. In this algorithm, initially all nodes are colored white. The CDS is grown from one node outward by coloring black those nodes that have maximum number of white neighbors. This algorithm yields a CDS of size at most $2(1+H(\Delta))opt_{MCDS}$, where H is the harmonic function and Δ is the maximum node degree. Das, et al. proposed an algorithm based on [35] by selecting a node with the maximum degree as the root of a spanning tree T, then repeatedly running the coloring procedure to form the spanning tree [32]. This algorithm has an approximation ratio of $O(\log\Delta)$ for a general graph. However, these algorithms are centralized algorithms, which makes their applications quite limited.

Distributed algorithms for CDS constructions have been proposed in recently years. They can be categorized as: WCDS-based, Pruning-based and MIS-based approaches. In these distributed algorithms, each node in the network is assigned a unique ID. The *Weakly Connected Dominating Set (WCDS)* is a dominating set of graph G such that WCDS and

its neighbors induce a connected subgraph of G. In a WCDS the nodes are partitioned into a set of cluster heads and cluster members, such that each cluster member is within the radio range of at least one cluster head. In the algorithm proposed in [36], nodes elect their neighbor with the lowest ID as their cluster head. Whenever a node with a lower ID moves into the range of a cluster head, it becomes the new cluster head. A problem of this algorithm is that the cluster structure is very unstable: when a lower ID node moves in, a cluster may break into many sub-clusters. Apparently, this reorganization is unnecessary in many situations.

Pruning based approaches construct a CDS first and then prune the redundant nodes from this CDS [32]. In a pruning based algorithm, any node having two disconnected neighbors is marked as a dominator. The redundant dominator set is post-processed by pruning nodes out of the CDS. The closed neighbor set of node u is defined as the set of u's neighbors plus u itself. A node u is pruned out if there exists a node v with higher ID such that the closed neighbor set of u is a subset of the closed neighbor set of v. Unfortunately, the theoretical bound on the resulting CDS obtained by this algorithm remains unspecified.

Minimum Independent Set (MIS) is an independent set such that adding any new node to the set breaks the independence property of the set. Thus, every node in the graph is adjacent to some node in the MIS. MIS based algorithms construct a CDS by finding an MIS and then connect it. [31, 61] gave an algorithm for unit disk graphs with performance bounds. It first constructs a rooted spanning tree in G, then a labeling process begins from the root to the leaves by broadcasting "DOMINATOR/DOMINATEE" messages to form the MIS. The final phase connects the nodes in the MIS to form a CDS. The algorithm has time and message complexities of $O(n)$ and $O(n \log n)$, respectively, for an n-node graph. The resulting CDS has a size of at most $8opt_{MCDS}$. [33] presented a multi-leader

MIS based algorithm by rooting the CDS in multiple sources. Its performance ratio is $192|opt_{MCDS}|+8$.

The CDS computed by these algorithms has at most $8opt_{MCDS}$ size, where opt_{MCDS} is the size of the MCDS. Although previous CDS construction algorithms achieve good results in terms of the size of the CDS, a minimum size CDS does not necessarily guarantee the optimal network performance from an energy-efficient point of view. All the existing MCDS algorithms assume that the battery discharging of a sensor node is linear, which is a rough assumption. Next we will adopt a battery model to propose battery-aware connected dominating set (BACDS) scheduling.

3.2 Mathematical Battery Model for WSN

The battery discharging model presented in Chapter 2 is an on-line computable model for general MANETs. In this section, we further reduce the computational complexity to make it implementable in WSNs. We assume that a battery is in discharging during time $[t_{begin}, t_{end}]$ with current I. The consumed energy α is calculated in Chapter 2 as

$$\begin{aligned} \alpha = & \quad I \times (t_{end} - t_{begin}) \\ + & \quad I \times \frac{\pi^2}{3\beta^2} \times \left(e^{-\beta^2 (t - t_{end})} - e^{-\beta^2 (t - t_{begin})} \right) \end{aligned} \tag{3.1}$$

where t is the current time and β is a constant parameter. The right hand side of (3.1) contains two components. The first term, $I \times (t_{end} - t_{begin})$, is simply the energy consumed in device during $[t_{begin}, t_{end}]$. The second term is the discharging loss in $[t_{begin}, t_{end}]$ and it decreases as t increases. The constant β (> 0) is an experimental chemical parameter which may be different from battery to battery. In general, the larger the β, the faster the

battery diffusion rate is, hence the less the discharging loss.

In MANETs, current I is a continuous variable for various applications, such as operating systems, multimedia transmission, word processing and interactive games. However, in WSNs, the simple sensing and data propagating activities of sensor nodes may only require several constant currents. We define the constant currents of dominator nodes and dominatee nodes as I_d and I_e, respectively. A dominator needs to keep active and listen to all channels at all times. Compared with I_d, the I_e of a dominatee is very low. We divide the sensor lifetime into discrete time slots with slot length δ. In each time slot the battery of a node is either as a dominator ($I = I_d$) or a dominatee ($I = I_e$). From (3.1) we have

$$\zeta_n(t) = I_n \times \frac{\pi^2}{3\beta^2}(e^{-\beta^2(t-n\delta)} - e^{-\beta^2(t-(n-1)\delta)}) \tag{3.2}$$

where I_n is either I_d or I_e, and t is the current time.

We can see that $\zeta_n(t)$ is recovered gradually in the following $(n+1)_{th}, (n+2)_{th}, \ldots$ slots until t. It should be mentioned that discharging loss $\zeta_n(t)$ is only a potentially recoverable energy. At time t the gross discharging loss energy of this battery is

$$\zeta(t) = \sum_{i=1}^{m} \zeta_i(t) \tag{3.3}$$

where $m = \lfloor t/\delta \rfloor$. The lower the $\zeta(t)$, the better the battery is recovered. To be aware of the battery recovery status, $\zeta(t)$ needs to be calculated at each slot. However, the computation can be simplified by observing that $\zeta_i(t)$ decreases exponentially as t increases. Naturally, $\zeta_i(t)$ can be ignored if $\zeta_i(t)$ is less than a small amount of energy c, where c is the energy to transmit a single packet. We introduce κ_i as $\zeta_i(\kappa_i\delta) < c < \zeta_i((\kappa_i - 1)\delta)$, that is, after

$t = \kappa_i\delta$, $\zeta_i(t)$ is ignored. We have

$$\kappa_i = \left\lceil \frac{1}{\beta^2\delta} \log \frac{\pi^2(1-e^{-\beta^2\delta})}{3\beta^2 c/I_i} \right\rceil \tag{3.4}$$

where $I_i = I_d$ or I_e. In MANET's battery model, κ_i is calculated for different input currents I. However the I in WSN has only two values: I_d or I_e. Therefore we maintain κ_i in a recovery table. At the m_{th} slot if $\kappa_i < (m-i)$, which indicates that $\zeta_i(m\delta) < c$, then this entry i is removed from the table.

Introducing this recovery table has several advantages and is also feasible for WSNs. First, we can reduce the computational complexity of battery-awareness. Only the remained entries in the table are used for computing $\zeta(t)$. Now (3.3) can be rewritten as

$$\zeta(t) = \sum_j \zeta_j(t) \tag{3.5}$$

where entry j is the entry remained in the recovery table. Second, we can reduce the complexity of table maintenance. In order to check whether an entry needs to be removed, rather than calculating $\zeta_i(t)$ for every i at each time, we only need to read κ_i from the table and compare it with m. Because κ_i is computed once, maintaining the recovery table is simple. Third, the size of the table is feasible for sensor memory. According to (3.4) and (3.5), the total entries in the recovery table are no more than $\lceil \frac{1}{\beta^2\delta} \log \frac{\pi^2(1-e^{-\beta^2\delta})}{3\beta^2 c/I_d} \rceil$. For various possible values of I_d and c of sensors, Table 1 shows the maximum number of entries in a recovery table ($\beta = 0.4$). Considering that the memory capacity of today's sensor node is typically larger than 512KB [37], it is acceptable to store and maintain such a recovery table in sensor memory. Thus, we can reduce the computational complexity by maintaining a recovery table on a sensor node with a feasible table size.

Table 1: The maximum size of the recovery table ($\beta = 0.4$)

c / I_d	$1200mA$	$800mA$	$400mA$	$100mA$
$200mA$min	1	2	3	4
$400mA$min	2	3	3	5
$600mA$min	3	3	4	6
$800mA$min	3	4	5	6

In summary, in this section we use the battery model to achieve battery-awareness by capturing its recovery $\zeta(t)$. The lower the $\zeta(t)$, the better the sensor node is recovered at time t. We introduce the recovery table to reduce the computational complexity. We also show that maintaining such a table is feasible for today's sensor nodes. Next we apply this battery model to construct the BACDS to prolong network lifetime.

3.3 Battery-Aware Connected Dominating Set

In this section we introduce the concept of battery-aware connected dominating set (BACDS), and show that the BACDS can achieve better network performance. Then we provide a distributed algorithm to construct BACDS in WSNs.

3.3.1 Battery-Aware Dominating

Let a WSN be represented by a graph $G = (V,E)$ where $|V| = n$. For each pair of nodes $u,v \in V$, $(u,v) \in E$ if and only if nodes u and v can communicate in one hop. The maximum node degree in G is Δ. Each node v is assigned a unique ID_v. $P_{residual}(v)$ is v's residual battery energy. $P_{threshold}$ is the threshold energy adopted in CDS construction algorithms. $\zeta^v(t)$ is the discharging loss of v at time t. For any subset $U \subseteq V$, we define

$\zeta^U_{max} = \max\{\zeta^v(t) | v \in U\}$.

Next we explain how to use the battery model to construct an energy-efficient dominating set in a WSN. Intuitively, longer network lifetime can be achieved by always choosing the "most fully recovered" sensor nodes as dominators. For a graph $G = (V,E)$ at time t, we define a BACDS as a set $S_B (\subseteq V)$ such that

$$S_B \text{ is a CDS of } G \text{ and } \zeta^{S_B}_{max} = \min\{\zeta^S_{max} | S \text{ is a } CDS \text{ of } G\} \tag{3.6}$$

An optimal BACDS is a BACDS with a minimum number of nodes and is denoted as MBACDS. For notational convenience, we let

$$\zeta^{BACDS} \equiv \min\{\zeta^S_{max} | S \text{ is a } CDS \text{ of } G\} \tag{3.7}$$

BACDS can achieve better performance than MCDS since it balances the energy consumption among sensor nodes. Table 2 gives the outline of the MCDS and BACDS algorithms for forming a virtual backbone in a WSN, where node i is qualified to be selected as a dominator as long as its residual battery energy $P_{residual}(i)$ is no less than the threshold energy $P_{threshold}$.

We conducted simulations to compare the two algorithms. Fig. 3.3 shows the simulated lifetime under BACDS and MCDS models for the network in Fig. 3.1. At the beginning all nodes are identical with battery capacity $C = 4.5 \times 10^4 mA\text{min}$ and $\beta = 0.4$. The discharging currents are $I_d = 900mA$ and $I_e = 10mA$. An MCDS is chosen as $Set_1 = \{C,D,F,G\}$ (Fig. 3.1). Since MCDS does not consider the battery behavior, Set_1 remains as the dominator until the energy of all nodes in Set_1 drops to a threshold $0.1 \times 10^4 mA\text{min}$. After that another MCDS is chosen as $Set_2 = \{A,B,E,H\}$. After Set_2 uses up its energy, no node in the

Table 2: Outline of MCDS and BACDS algorithms for forming a virtual backbone

MCDS Algorithm:
Repeat
Every node i with $P_{residual}(i) \geq P_{threshold}$ is marked as *qualified*;
All other nodes are marked as *unqualified*;
Call *MCDS Construction Algorithm* for all *qualified* nodes;
If successful, use the MCDS constructed as the backbone;
Otherwise report 'no backbone can be formed' and exit;
Until Some dominator j has $P_{residual}(j) < P_{threshold}$;

BACDS Algorithm:
Repeat
Every node i with $P_{residual}(i) \geq P_{threshold}$ is marked as *qualified*;
All other nodes are marked as *unqualified*;
Call *BACDS Construction Algorithm* for all *qualified* nodes;
If successful, use the BACDS constructed as the backbone;
Otherwise report 'no backbone can be formed' and exit;
Until δ time has elapsed or some dominator j has $P_{residual}(j) < P_{threshold}$;

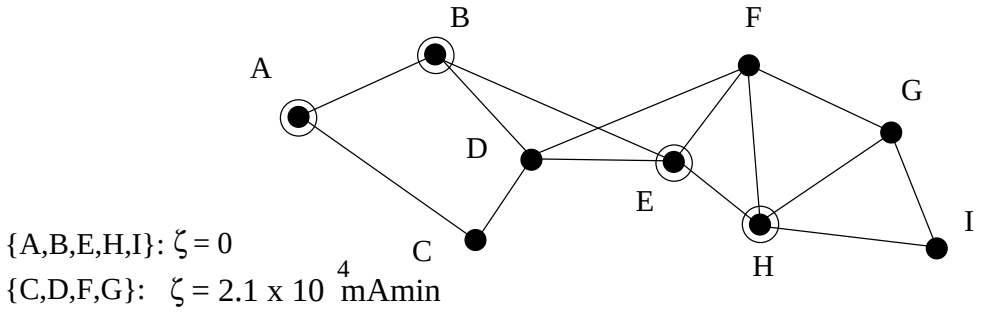


Figure 3.2: The battery-aware connected dominating set (BACDS) for the network in Fig. 3.1 at $t = 10$min. The $\zeta^i(t)$ for each node i is listed above.

network is qualified as a dominator. The total network lifetime is 56min (Fig. 3.3).

On the other hand, in the BACDS model a CDS is formed by the nodes with minimum $\zeta(t)$s. The network reorganizes the BACDS for every δ time. Suppose at the beginning the BACDS is still $Set'_1 = \{C, D, F, G\}$. After $\delta = 10$min the energy of nodes in Set'_1 is reduced to $2.1 \times 10^4 mA$min. At this time, a new BACDS is chosen as $Set'_2 = \{A, B, E, H\}$ (Fig. 3.2). During the next 10min, Set'_2 dissipates the energy to $2.1 \times 10^4 mA$min while Set_1 recoveries its nodes' energy from $2.1 \times 10^4 mA$min to $3.35 \times 10^4 mA$min. Then the BACDS is organized again. As shown in Fig. 3.3, the total network lifetime in the BACDS model is 69 mins, which is 23.2% longer than the MCDS model.

We have seen that BACDS can achieve longer lifetime than MCDS. Next we will consider the BACDS construction algorithm.

3.3.2 Formalization of BACDS Construction Problem

The BACDS construction problem can be formalized as: given a graph $G = (V, E)$ and $\zeta^i(t)$ for each node i in G, find an MBACDS. For simplicity, we use ζ to denote $\zeta(t)$ in the rest of the chapter. First, we have a theorem regarding the NP-hardness of the problem.

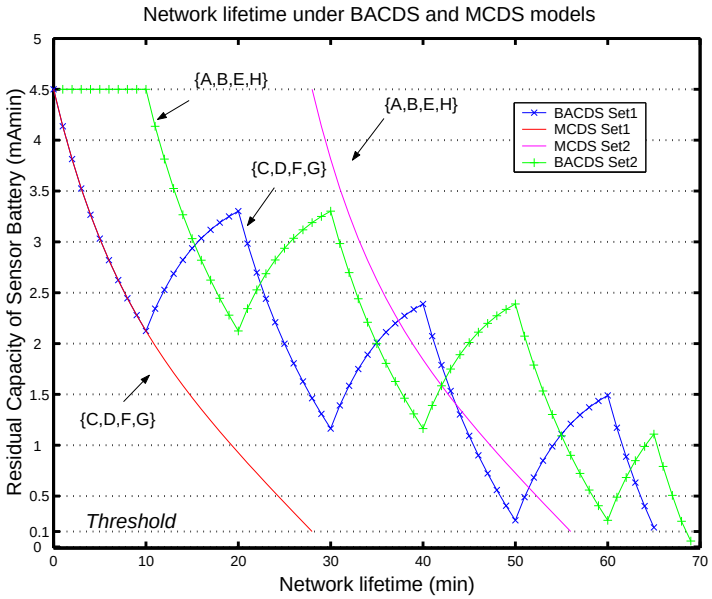


Figure 3.3: The WSN in Fig. 3.1 achieves longer lifetime using the BACDS model than the MCDS model. The results were simulated with the battery discharging model. In this case the network lifetime is prolonged by 23.2% in the BACDS model.

Theorem 1 *Finding an MBACDS in G is NP-hard.*

Proof. Consider a special case that all nodes in G have the same ζ values. In this case, finding an MBACDS in G is equivalent to finding an MCDS in G. Since the MCDS problem is NP-hard, the MBACDS problem is also NP-hard. ■

Now since the MBACDS construction is NP-hard, we will construct a BACDS by an approximation algorithm. By definition, the BACDS to be constructed, S_B, must satisfy the following two conditions: (i) $\zeta_{max}^{S_B} = \zeta^{BACDS}$; (ii) S_B is a CDS of G. Also, the size of S_B should be as small as possible.

Our algorithm is designed to find a set to satisfy these conditions. To satisfy condition (i), the algorithm finds a subset SET^+ in G, such that for any subset $S'(\subseteq SET^+)$, $\zeta_{max}^{S'} \leq \zeta^{BACDS}$. We will prove that, as long as a set $S'(\subseteq SET^+)$ is a CDS of G, we can guarantee that $\zeta_{max}^{S'} = \zeta^{BACDS}$.

To satisfy condition (ii), the algorithm will find two sets: CDS^+ and COV, both in SET^+. $CDS^+(\subseteq SET^+)$ is an MCDS of SET^+. $COV(\subseteq SET^+)$ is a cover of $V - SET^+$, which dominates all the nodes in $V - SET^+$. We will prove that $SET^0 = COV \bigcup CDS^+$ is the BACDS we want. The detailed algorithm will be presented in the next subsection. Table 3 gives an outline of the BACDS construction algorithm.

Table 3: The Outline of the BACDS Construction

1. Find a subset SET^+ in G;
2. Construct a subset COV in SET^+;
/* COV covers the nodes in $V - SET^+$ */
3. Construct a subset CDS^+ in SET^+;
/* CDS^+ is the CDS of SET^+ */
$SET^0 = COV \bigcup CDS^+$ is a BACDS;

We now show that $SET^0 = COV \bigcup CDS^+$ is indeed a BACDS.

Theorem 2 *$SET^0 = COV \bigcup CDS^+$ is a BACDS.*

Proof. We first define SET^+ as

$$SET^+ = \{v \in V | \zeta^v \leq \zeta^{BACDS}\} \tag{3.8}$$

We know that CDS^+ is a CDS of SET^+ and COV dominates the nodes in $V - SET^+$. To prove $SET^0 = COV \bigcup CDS^+$ is a BACDS, we need to show that set SET^0 is a connected dominating set of G, and $\zeta_{max}^{SET^0}$ is minimized.

First, we show that SET^0 is connected. Because $COV \subseteq SET^+$, every node in COV is at most one hop away from CDS^+. Also, since CDS^+ is connected, $SET^0 = COV \bigcup CDS^+$ is connected.

Second, we prove that SET^0 is a dominating set of G. Since CDS^+ is a CDS of SET^+, all nodes in SET^+ are dominated by CDS^+. Also, since all nodes in $V - SET^+$ are covered by COV, $SET^0 = COV \bigcup CDS^+$ is a dominating set of G.

Finally, we show that $\zeta_{max}^{SET^0}$ is minimized, i.e. $\zeta_{max}^{SET^0} = \zeta^{BACDS}$. Since $SET^0 \subseteq SET^+ = \{v \in V | \zeta^v \leq \zeta^{BACDS}\}$, we have $\zeta_{max}^{SET^0} \leq \zeta^{BACDS}$. However, SET^0 is also a CDS of G. Thus $\zeta_{max}^{SET^0} \geq \zeta^{BACDS}$. Therefore, $\zeta_{max}^{SET^0} = \zeta^{BACDS}$, and we have proved that $SET^0 = COV \bigcup CDS^+$ is a BACDS. ■

In summary, in this subsection we have shown that we can construct a BACDS by finding three subsets SET^+, COV and CDS^+ step by step. Then $SET^0 = COV \bigcup CDS^+$ is a BACDS. Next we will give the algorithms to construct SET^+, COV and CDS^+.

3.3.3 A Distributed Algorithm for BACDS Construction

Our algorithm for BACDS construction consists of three procedures for constructing SET^+ and COV in G. In the algorithm, we use $N(v)$ to denote the set of neighbor nodes of v. each node is colored in one of the three colors: black, white and gray. $N_{black}(v), N_{white}(v)$ and $N_{gray}(v)$ are the sets of black, white and gray neighbors of v, respectively. Also we use sets $list_{gray}(i)$ and $list(i)$ to store the neighbor information for node i.

First we consider the construction of SET^+. To find SET^+, we do the following: the ζ^i of each node i in G is collected in the sink; then the sink calculates $\zeta_{max}^{SET^+}$ and broadcasts it in a beacon; on receiving the beacon, all nodes with $\zeta^i \leq \zeta_{max}^{SET^+}$ form set SET^+. The procedure is described in Table 4. Initially, all nodes in G are colored gray. Then the algorithm colors some gray nodes black. In the end, the set of black nodes is SET^+.

Note that in SET^+ finding procedure, the sink collects information from sensors only once, and broadcasts the $\zeta_{max}^{SET^+}$ in a single beacon. The overhead of information collecting and broadcasting is minimum.

Now we consider the construction of COV. This problem can be formulated as: Given two graphs $\overline{G}_1$ and $\overline{G}_2$. Nodes in $\overline{G}_1$ and $\overline{G}_2$ are interconnected with edges. A minimum size COV is to be found in $\overline{G}_1$ such that $\overline{G}_2 \subseteq N(\overline{G}_1)$. We show that finding a minimum size COV is NP-hard even in an UDG graph. If we can show that this problem is NP-hard in an UDG, it is clear that this problem is also NP-hard in a general graph. We have the following theorem regarding its NP-hardness.

Theorem 3 *Finding a minimum COV in an UDG is NP-hard.*

Proof. To see this, we use the fact that it is NP-hard to find a minimum dominating set in an UDG. Given any UDG instance, say, $\overline{G}$, which has vertex set $\overline{V} = \{v_1, v_2, \ldots, v_n\}$, we construct another UDG $\overline{G}'$, which has vertex set $\overline{V}' = \{v_1, v_2, \ldots, v_n, v_1', v_2', \ldots, v_n'\}$. $\overline{G}'$

Table 4: SET^+ finding procedure

1. All nodes in G are colored gray;
2. Each gray node i in the network does the following:
3. *begin*
4. Broadcast a packet which includes its ID_i;
5. Receive neighbors' IDs;
6. Calculate ζ^i;
7. Route a packet with (ID_i, $ID_{N(i)}$, ζ^i) to the sink;
8. Listen to the beacon for $\zeta_{max}^{SET^+}$;
9. **if** $\zeta^i \leq \zeta_{max}^{SET^+}$ **then** node i is colored black;
10. *end*;

11. The sink does the following:
12. *begin /* Calculate* $\zeta_{max}^{SET^+}$ **/*
13. Collect packets from sensor nodes;
14. $S_0 := \{\zeta^i | i = 1, 2, \ldots, n\}$;
15. $S_1 := \emptyset$;
16. **repeat**
17. $S_2 :=$ {those nodes with the smallest ζ in S_0};
18. $S_0 := S_0 - S_2$;
19. $S_1 := S_1 \cup S_2$;
20. **until** S_1 is a CDS **or** $S_0 = \emptyset$;
21. **if** S_1 is a CDS **then**
22. $\zeta_{max}^{SET^+} := \zeta_{max}^{S_1}$;
23. Broadcast $\zeta_{max}^{SET^+}$ in a beacon;
24. **end if**;
25. **else** report 'no CDS can be found'
26. *end*;

$SET^+ :=$ {black nodes};

is actually constructed by adding n new vertices $v'_1, v'_2, \ldots, v'_n$ to $\overline{G}$, where v'_i and v_i are geographically close enough such that v'_i has the same set of neighbors as v_i. Then let $\overline{G}_1 = \{v_1, v_2, \ldots, v_n\}$ and $\overline{G}_2 = \{v'_1, v'_2, \ldots, v'_n\}$. We will find the COV in $\overline{G}_1$, where $\overline{G}_2$ is the set of vertices to be covered. It is not hard to see that the optimal solution to the COV problem in the new UDG is also a minimum dominating set in the original UDG. Because constructing a minimum dominating set in UDG is a NP-hard problem [61], the problem of finding COV is therefore NP-hard. ■

An approximation algorithm to construct a COV has been given in [62]. Initially, $S_1 = SET^-$ and $S_2 = V - SET^+$. This algorithm greedily selects node i in S_1 such that $|N(i) \bigcap S_2|$ is maximized. Then node i is moved to COV and $N(i) \bigcap S_2$ are removed from S_2, respectively. This algorithm can finally obtain a COV with $|COV| \leq \log(|SET^-|)$. However, this approximation algorithm is difficult to be implemented in our scenario, because it is a centralized algorithm. At each step this algorithm has to select a node with a maximum number of neighbors, which requires the collection and analysis of global information. Next we will present a distributed algorithm to construct a COV. In our algorithm, nodes only need to know neighbors at most 2 hops away.

Since $COV (\subseteq SET^+)$ is a cover of $V - SET^+$, to find COV, we only need to consider the nodes which are one hop away from $V - SET^+$. We use SET^- to denote these nodes. SET^- is defined as

$$SET^- = \{v \in SET^+ | \exists u \in (V - SET^+), (u, v) \in E\} \tag{3.9}$$

Fig. 3.4 gives the SET^+ and SET^- for a given WSN with their associated ζ values. SET^- can be found by the procedure described in Table 5.

Now we construct COV from SET^-. COV can be obtained as follows. Initially, $COV =$

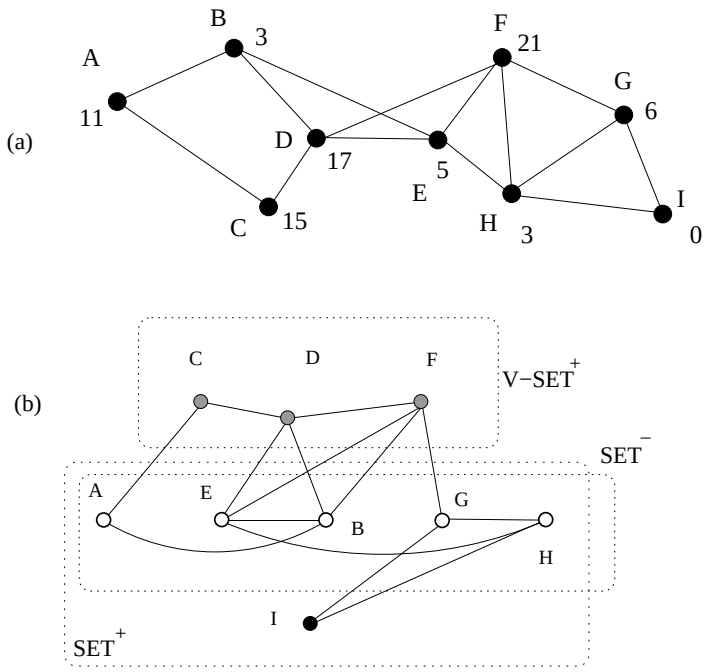


Figure 3.4: Constructing SET^+ and SET^- in graph G. (a) Nodes are associated with their ζ values. (b) SET^+ and SET^- in (a).

Table 5: SET^- finding procedure

1. $SET^- := \emptyset$;
2. Each node $i \in SET^+$ does the following:
3. *begin*
4. **if** $\exists j \in N(i)$ and $j \in (V - SET^+)$ **then**
5. $SET^- := SET^- \cup \{i\}$;
6. *end*;

$\emptyset$, $S_1 = (V - SET^+)$. Then, for node $v \in S_1$ with the lowest ID number, add all nodes $u \in SET^-$ to COV, where u and v are one hop neighbors. Then we remove $N(u) \cap S_1$ from S_1. Repeatedly doing so until S_1 is empty. Then we obtain set COV. The localized procedure for finding COV from SET^- is described in Table 6. There are three colors used to color a node: white, black and gray. Initially, nodes in SET^- are colored white, and nodes in $V - SET^+$ are colored gray. Then the procedure colors some white nodes black. In the end, the set of black nodes is the COV. Four messages are used: *addblack*, *black*, *remove* and $update$.

Our algorithm only requires at most 2-hop information to construct the COV. We now prove the correctness of the *COV Finding Procedure*. We will show that the COV found by this procedure is a cover of $V - SET^+$. We will also give an upper bound on the size of set COV.

Lemma 1 *Set COV found by COV Finding Procedure covers $V - SET^+$. Its size $|COV|$ is at most Δopt_c, where Δ is the maximum node degree of G and opt_c is the size of an optimal cover.*

Proof. By contradiction. Suppose when the *COV Finding Procedure* terminates, there exists a gray node i which is not covered by COV. It indicates that $N(i) \cap COV = \emptyset$, which means $N(i) \cap SET^-$ is not colored black. Since $N(i) \cap SET^-$ is not colored black, all nodes in $N(i) \cap SET^-$ should receive *remove* messages from i, otherwise their $list_{gray} \neq \emptyset$ and the procedure does not terminate. However, i broadcasts *remove* message if and only if one of its white neighbors is colored black. Thus, $N(i) \cap COV \neq \emptyset$, which is a contradiction. Hence COV is a cover.

Now we consider the size of COV. Initially COV is empty, and we add some nodes to it in later steps. We will show that: (i) at each step, at least one node, which belongs to the

Table 6: *COV* finding procedure

1. Each nodes in $SET^- \bigcup (V - SET^+)$ does the following:
2. *begin*
3. Nodes in SET^- are colored white;
4. Nodes in $V - SET^+$ are colored gray;
5. $COV := \emptyset$;
6. Each gray node i broadcasts ID_i to its white neighbors $N_{white}(i)$;
7. Each white node j adds the received IDs to a set $list_{gray}(j)$ and broadcasts $list_{gray}(j)$ to its gray neighbors $N_{gray}(j)$;
8. Each gray node i does the following:
9. *begin*
10. Receive all $list_{gray}(N_{white}(i))$;
11. $list(i) := \bigcup list_{gray}(N_{white}(i))$;
12. **while** $list(i) \neq \emptyset$ **do**
13. **if** $ID_i = \min\{id | id \in list(i)\}$ **then**
14. Broadcast *addblack* to $N_{white}(i)$;
15. $list(i) := \emptyset$;
16. **else if** *black* message is received **then**
17. broadcast *remove* to $N_{white}(i)$;
18. $list(i) := \emptyset$;
19. **else if** $update(k)$ message is received **then**
20. $list(i) := list(i) - \{k\}$;
21. **end while**;
22. *end*;
23. Each white node j does the following:
24. *begin*
25. **while** $list_{gray}(j) \neq \emptyset$ **do**
26. **if** *addblack* message is received **then**
27. Color itself black;
28. $list_{gray}(j) := \emptyset$;
29. Broadcast *black* to $N_{gray}(j)$;
30. **else if** *remove* message is received from k **then**
31. Broadcast $update(k)$ to $N_{gray}(j)$;
32. $list_{gray}(j) := list_{gray}(j) - \{k\}$;
33. **end while**;
34. *end*;
35. *end;*

$COV := \{$black nodes$\}$;

optimal cover, is added to COV; and (ii) at each step, at most $\Delta - 1$ extra nodes, which do not belong to the optimal cover, are added to the COV. By combining (i) and (ii), we obtain that $|COV|$ is at most Δopt_c, where opt_c is the size of an optimal cover.

We first consider (i). According to the algorithm, at each step, node i sends *addblack* messages to $N_{white}(i)$. Therefore, there are at most Δ nodes added to COV in each step. Among the Δ nodes there must exist at least one node which is in the optimal cover, because an optimal cover has to cover node i. Next we consider (ii). Since among the Δ nodes added to COV there are at least one node belonging to the optimal cover, each time we add at most $\Delta - 1$ extra nodes not belonging to the optimal cover to COV. Thus, $|COV|$ is at most Δopt_c.

■

The worst case occurs when nodes in $V - SET^+$ are not connected by nodes in SET^-. Fig. 3.5 gives an example of $|COV| = \Delta opt_c$. Because node 1 has the lowest ID in $list(i) = \{1\}$, 1 broadcasts *addblack* message to its neighbors: A, B, C; and adds them to COV set. The same thing happens for node 2 and 3. Clearly $opt_c = 3$. Thus we have $|COV| = \Delta opt_c = 9$, where $\Delta = 3$ is the maximum degree of nodes in $V - SET^+$.

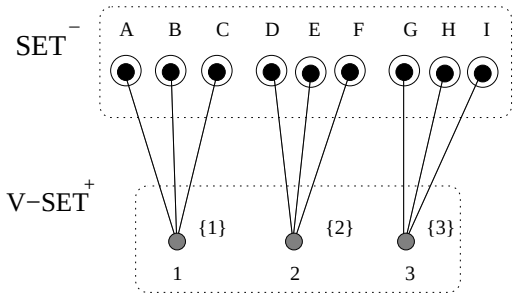


Figure 3.5: The size of finding COV is at most Δopt_c. opt_c is the size of an optimal cover.

Now we prove that the *COV Finding Procedure* can finally terminates, i.e. all $list_{gray}(j)$

and $list(i)$ will finally be empty. We also give the time complexity of *COV Finding Procedure*.

Lemma 2 *All $list_{gray}(j)$ and $list(i)$ will finally be empty. COV Finding Procedure has time complexity of $O(n-|SET^{+}|+|SET^{-}|)$.*

Proof. First, we show that $list_{gray}(j)$ will finally be empty. Suppose there is a white node j where $list_{gray}(j) \neq \emptyset$. Without loss of generality, we have a node $u_1 \in list_{gray}(j)$ with the lowest ID. In this procedure u_1 is the node that colors j black. If u_1 does not send *addblack* to j, there must be a node $u_2 \in list(u_1)$ such that $ID(u_1) > ID(u_2)$. Then if u_2 does not broadcast *addblack*, there must be a node $u_3 \in list(u_2)$ such that $ID(u_2) > ID(u_3)$, and so on. Then we have $ID(u_1) > ID(u_2) > ID(u_3) > \cdots$. Since there are a finite number of nodes, there must exist a node, u_k, such that $ID(u_k) = \min\{id | id \in list(u_k)\}$. The node u_k should send *addblack*, which indicates that finally $list_{gray}(j)$ should be empty.

Second, we show that $list(i)$ will finally be empty. By Lemma 1, each gray node i is finally covered by some black node, which indicates that all gray nodes received *black* messages. Therefore, finally $list(i)$ will be $\emptyset$. In this procedure, because it colors at least one node black in each step, the time complexity is $O(n-|SET^{+}|+|SET^{-}|)$. ■

Fig. 3.6 shows the worst case of the *COV Finding Procedure*. With the lowest ID in $list(1)$, node 1 broadcasts *addblack* message. Node *A* is colored black and informs node 2 by broadcasting *black* message. From the right to the left, the procedure adds one node to *COV* at each step.

Now we consider the CDS^{+} construction. We can adopt any existing MCDS construction algorithm for this purpose. Since the MIS-based algorithm in [31, 61] achieves the best performance in terms of its set size and time and message complexities, we adopt this

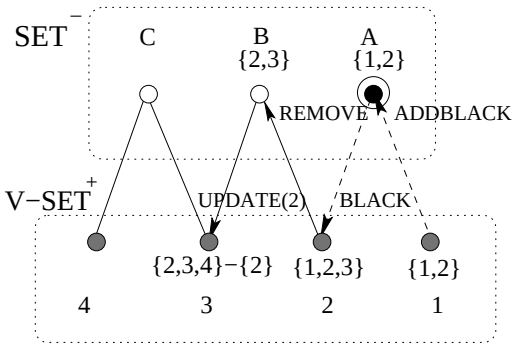


Figure 3.6: In the worst case, the complexity of finding COV is $O(n-|SET^{+}|+|SET^{-}|)$. It processes one node in each step.

Table 7: The algorithm for finding a BACDS in graph G

1. *begin*
2. call SET^{+} *Finding Procedure* for G;
3. call SET^{-} *Finding Procedure* for SET^{+};
4. call *COV Finding Procedure* for SET^{-};
5. call CDS^{+} *Finding Algorithm for* SET^{+}*;*
6. $SET^{0} := COV \bigcup CDS^{+}$;
7. *end*;

algorithm here. This algorithm is a UDG based algorithm. However, our BACDS constructing algorithm is not restricted to a UDG. In fact, we can employ any other general graph based MCDS algorithm to construct CDS^{+}. The complete algorithm for finding a BACDS is given in Table 7.

Fig. 3.7 gives an example of finding a BACDS. First, SET^{+} and SET^{-} are found (Fig. 3.7(a)); then COV is constructed by the algorithm (Fig. 3.7(b)-(e)); finally, CDS^{+} is found and SET^{0} is formed (Fig. 3.7(g)). In this example, compared with the MBACDS (Fig. 3.7(h)), SET^{0} contains one more dominator.

In the next subsection, we will analyze the performance of our BACDS construction

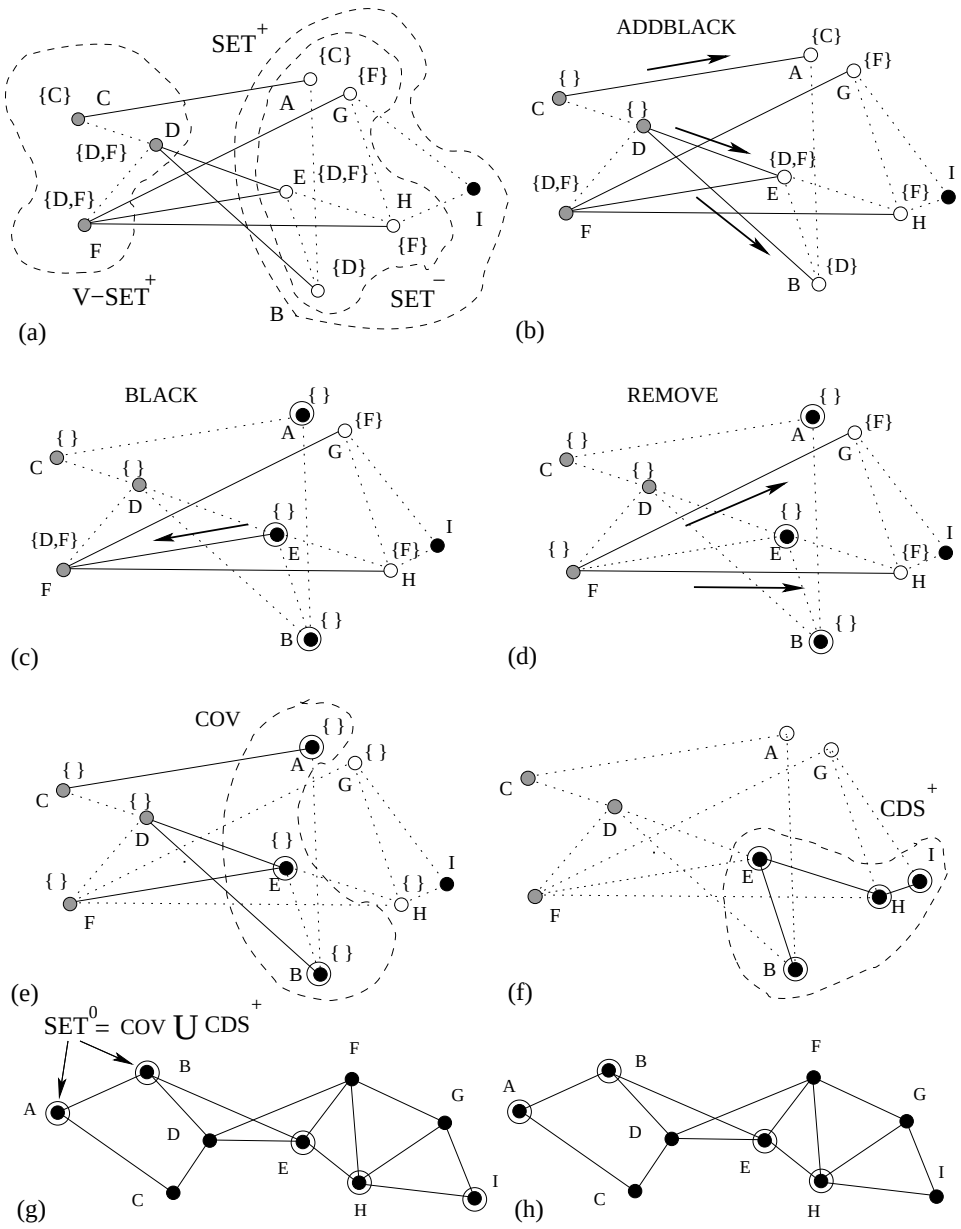


Figure 3.7: Finding a BACDS in the network in Fig. 3.4(a). (a) SET^{+} and SET^{-} in G, set $list(i)$ or $list_{gray}(i)$ of each node i are listed above. (b)-(e) Finding COV in SET^{-}. (f) Finding CDS^{+} in SET^{+}. (g) SET^{0} is the resulting BACDS. (h) The optimal BACDS of this network.

algorithm, and give an upper bound on the size of SET^0.

3.3.4 Complexity Analysis of the Algorithm

In this subsection we analyze the approximation ratio, the time complexity and the message complexity of the algorithm. We also consider mobility issues for this algorithm. We use opt, opt_{MCDS}, opt_c, and $opt_{MCDS}^{SET^+}$ to denote the sizes of MBACDS in G, MCDS in G, the minimum cover in SET^- and the minimum CDS^+ in SET^+, respectively.

Theorem 4 *The size of SET^0 is at most $(8+\Delta)opt$. The time complexity and the message complexity of the algorithm is $O(n)$ and $O(n(\sqrt{n}+\log n+\Delta))$, respectively, where n is the number of nodes in G.*

Proof. First, we consider the size of SET^0. Note that $MBACDS \cap SET^-$ is a cover and $|MBACDS| = opt$. It indicates that

$$opt_c \leq |MBACDS \cap SET^-| \leq opt \tag{3.10}$$

Since we have proved in Lemma 1 that $|COV| \leq \Delta opt_c$, we have

$$|COV| \leq \Delta opt \tag{3.11}$$

We employ the MIS based CDS finding algorithm for CDS^+ construction. It can obtain a CDS^+ with a size of at most $8opt_{MCDS}^{SET^+}$. Thus,

$$|CDS^+| \leq 8opt_{MCDS}^{SET^+} \tag{3.12}$$

Because constructing an MBACDS needs to consider the extra battery parameter, the size of the MBACDS is no less than opt_{MCDS}. Thus we have

$$opt_{MCDS} \leq opt \tag{3.13}$$

Clearly, the size of a minimum CDS^+ in SET^+ is at most opt_{MCDS} because it needs to consider extra nodes in $V - SET^+$. Thus,

$$opt_{MCDS}^{SET^+} \leq opt_{MCDS} \tag{3.14}$$

By (3.12), (3.14) and (3.13) we obtain

$$|CDS^+| \leq 8opt \tag{3.15}$$

Therefore from (3.11) and (3.15) we have

$$|SET^0| = |COV \cup CDS^+| \leq (8+\Delta)opt$$

Now we analyze the complexity of the algorithm. Our algorithm consists of three phases: SET^+ and SET^- constructions, COV construction and CDS^+ construction. In the first phase, to find SET^+ and SET^-, each node only needs to propagate their messages to the sink and wait for a beacon. A node needs at most $\sqrt{n}$ hops to relay a message to the sink. Hence the time complexity for sending the message and receiving a beacon are $O(\sqrt{n})$ and $O(1)$, respectively. The time complexity of the second phase is $O(n - |SET^+| + |SET^-|)$ as Lemma 2 shows. The complexity of the third phase is $O(|SET^+|)$. Therefore, the total time complexity of the algorithm is $O(n)$.

Now we consider the number of messages transmitted. In the first phase, the total number of messages relayed in the network is at most $O(n\sqrt{n})$. The beacon has $O(1)$ message complexity. At the beginning of the second phase, to set up the lists, all white nodes and gray nodes need to send $|SET^-|$ and $n-|SET^+|$ messages, respectively. After that, all the gray nodes send at most $|SET^-|$ *addblack* messages and $\Delta|SET^-|$ *remove* messages. All the white nodes send at most $\Delta(n-|SET^+|)$ *update* messages, and all the black nodes send no more than $|SET^-|$ *black* messages. Thus, in the second phase, there are totally at most $O(n\Delta)$ messages. If we employ the algorithm in [31], the message complexity in the third phase is $O(n+n\log n)$. Hence, the total message complexity of our algorithm is $O(n(\sqrt{n}+\log n+\Delta))$. ■

In our BACDS construction algorithm, the network is required to reorganized every δ period by selecting those most fully recovered nodes. This periodical organization seems an extra overhead. However, we observe that as long as we employ a CDS infrastructure in a WSN, there is always such overhead, and the overhead in the MCDS model might be even higher than that in the BACDS model. This is because that a CDS has to be reorganized whenever a dominator dies. The overheads might be even higher in the MCDS case. Because MCDS algorithms select those nodes with a maximum node degree, lowest ID or shortest distance to neighbors as dominators. These nodes, which we referred to as burden nodes, remain as the dominators all the time and have larger ζ and lower available energy. An MCDS containing such burden nodes is more likely to be reorganized from time to time whenever any burden node uses up its energy. While the metric of our BACDS construction algorithm is to balance the energy consumption among sensor nodes, in other words, to avoid the burden nodes. Therefore, it is not surprising to see that the reorganization overhead in the BACDS model may be lower than that in the MCDS model. The

simulation results in Section 3.4 verify our observations.

Furthermore, compared with MCDS construction algorithms, the BACDS algorithm has less overhead even during the construction. Naturally, the BACDS partitions the network into different sets: SET^+, SET^-, COV, etc. In each step of the construction, only the sensors in a specified set work on constructing the BACDS, and other sensors turn them into sleep or low energy state to save energy. For instance, when constructing COV in SET^-, the sensors in $SET^+ - SET^-$ turn to sleep; and when constructing the CDS^+ the sensors in $V - SET^+$ turn to sleep. Thus, energy dissipation is reduced. However, when constructing an MCDS, all sensor nodes have to keep active and listen to their channels all the time, even at the time they do not contribute to the MCDS construction. Consequently, more energy is dissipated than the BACDS algorithm.

Our BACDS construction algorithm also considers the mobility issue. By monitoring the sensor mobility and their battery status, it can dynamically adjust the BACDS locally. There are three changes that should be handled to maintain the BACDS: (1) A dominator $v \in SET^0$ moves away from $N(v)$; (2) A new node v moves in; (3) A dominator finds out that one of its dominatees v has $\zeta^v \leq \zeta^{SET^+}$. Fig. 3.8 illustrates the situation. In case (1) a new dominator must be selected. Since there must exist another dominator node u in $N(v)$ to connect the original dominating set, a spanning tree rooted at u can be formed among the nodes in $N(v)$. Nodes in this spanning tree become the new dominators. In both case (2) and case (3), node v needs to check if making itself as a new dominator can reduce the size of the BACDS. In our algorithm, v collects all $N(u_i)$ information from its dominator neighbor nodes u_i, $i = 1, 2, \ldots$. Node v is selected as a new dominator if and only if there are more than one u_i such that $N(u_i) \subset N(v)$. After v is selected as the new dominator, u_i are turned to dominatees. Therefore, all nodes in this network are still dominated by at

least one dominator while the size of the BACDS is reduced.

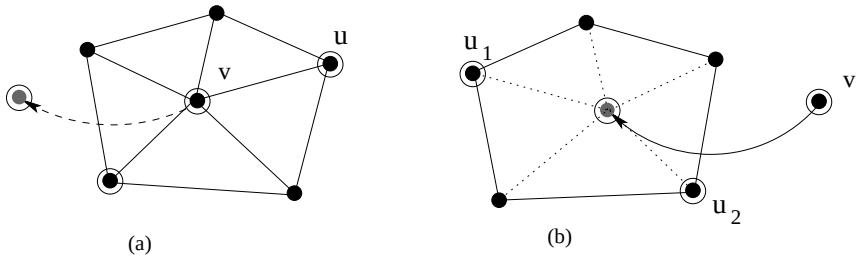


Figure 3.8: Mobility issue of BACDS. (a) A dominator moves away. (b) A new dominator is selected.

3.4 Performance Evaluations of BACDS

We conducted extensive simulations to evaluation the network performances under the BACDS model and MCDS model, in terms of network lifetime, energy dissipation and set size. We simulated the network lifetime under different models. Our simulation results show that the BACDS achieves longer network lifetime. Their re-organization overheads are also compared. By implementing our BACDS construction algorithm, we also compare the set sizes of SET^0 with MBACDS, where the MBACDS of a network is computed off-line. The simulation results verify the upper bound of SET^0 in Theorem 4.

Network Lifetime: We first compare the network lifetime achieved by MCDS and BACDS. We generated a WSN with $n = 100$ randomly deployed nodes. The communication radius is $r = 1$. Any pair of nodes are connected if and only if their distance is shorter than r. We let $d = 6$ be the average degree of nodes, where d indicates the density of the network. Each sensor node i is associated with a discharging loss value ζ^i ($i \leq n$). ζ^i is uniformly randomly distributed in $[0, 2 \times 10^4](mA\text{min})$. For simplicity, we assume that initially the available energy of node i is $C_i = 4.5 \times 10^4 mA\text{min}$ and $\beta_i = 0.4$ as in the example

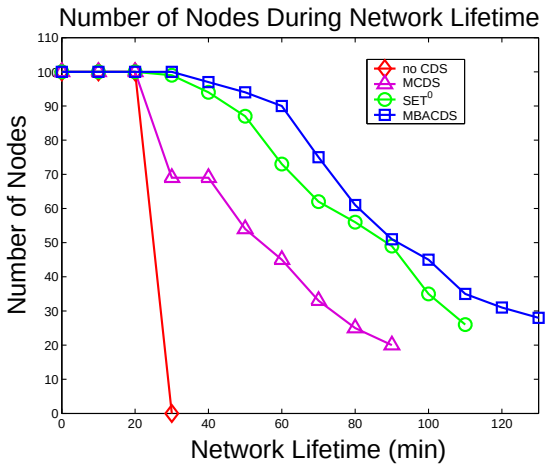


Figure 3.9: Comparing the network lifetime under different models.

of Fig. 3.3. The discharging currents are $I_d = 900mA$ and $I_e = 10mA$, respectively.

Fig. 3.9 shows the number of nodes decreases with respect to the lifetime. Four models: no-CDS, MCDS, MBACDS and SET^0 are implemented. A network without a CDS infrastructure terminates at 30min as all its nodes use up their energy. MCDS achieves about 83min total lifetime. MBACDS organizes its dominators every 10min. The lifetime is prolonged by up to 52% in MBACDS model. SET^0 is the BACDS constructed by our algorithm. It obtains a lifetime prolonged up to 30%. We also note that MBACDS terminates with more nodes remained in the network. This is because MBACDS balances the energy consumption of each nodes, thus more nodes are preserved in the network.

Power Dissipation: We compare the available energy per sensor node under different models in Fig. 3.10. In this comparison, the average power is the total available power of the network over the number of active nodes. The average power is normalized. MBACDS achieves higher average power during the lifetime. It should be mentioned that at 30min the average power of MCDS suddenly increases. This is because many of its dominators suddenly die at that time. Consequently, the average power increases as the total number

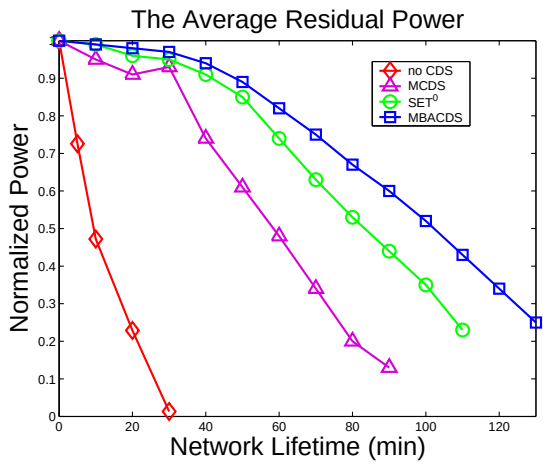


Figure 3.10: The average power per node in the network.

of nodes decreases.

Set Size: We implemented our BACDS construction algorithm to verify the upper bound on the set size. The size of the generated network is $n = 20, \ldots, 140$. Four sets are compared: SET^+, SET^-, SET^0 and MBACDS. Fig. 3.11 compares the sizes of these sets with different node degree d and discharging loss ζ. It shows that as d increases, the sizes of MBACDS and SET^0 are reduced. This is due to the larger degree of the dominators. While as ζ increases its distribution space from $[0,50]$ to $[0,100]$, the sizes of MBACDS and SET^0 increase. The results verify the upper bound of SET^0 in Theorem 4.

3.5 Summaries

In this chapter we have considered constructing battery-aware energy-efficient backbones in WSNs to improve the network performance. We first gave a simplified battery model in order to accurately describe the battery behavior on-line and reduce the computational complexity on a sensor node. Then by using the battery model, we have showed that

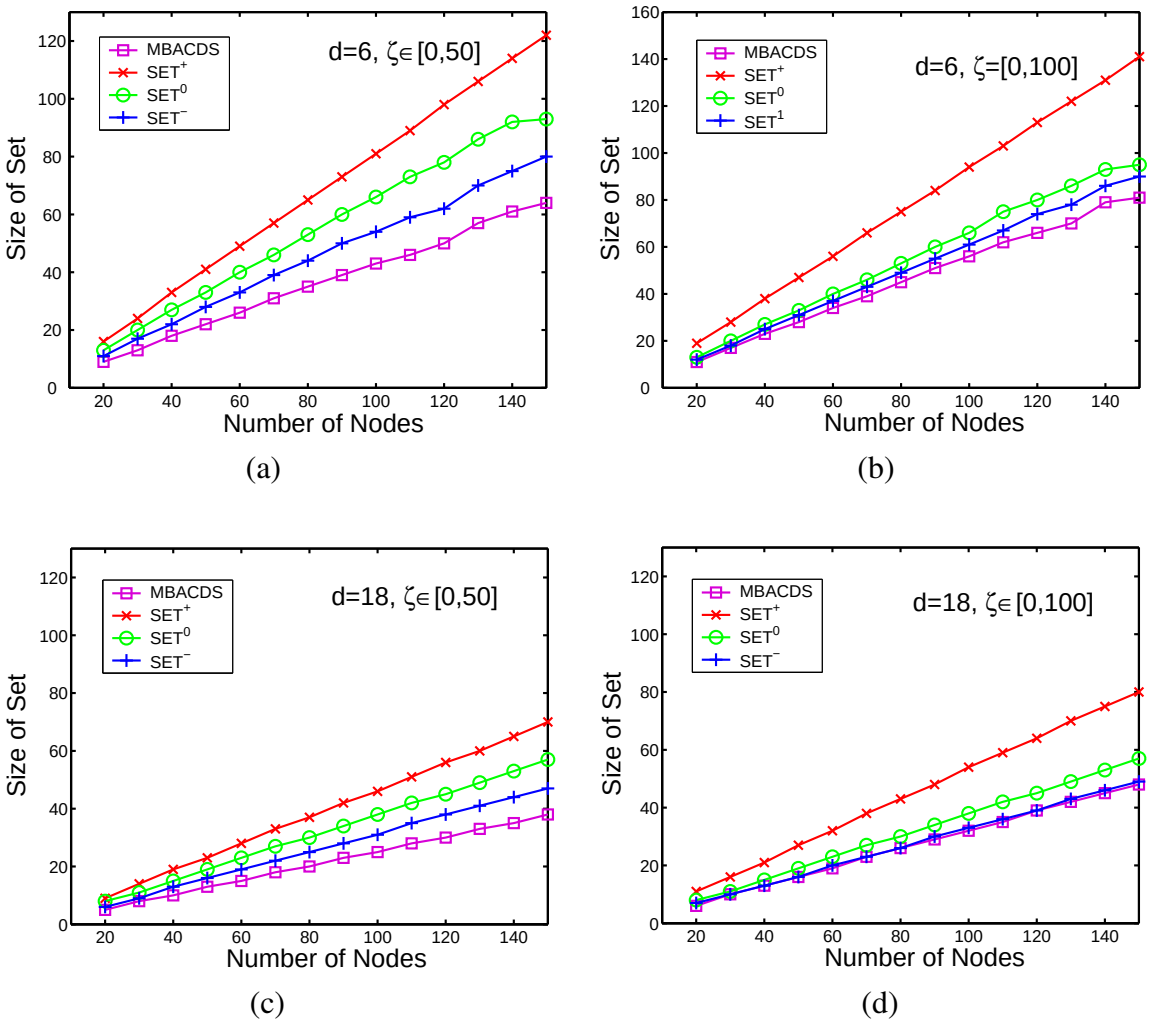


Figure 3.11: The size of constructed sets with different ζ and d.

an MCDS does not necessarily achieve maximum lifetime in a WSN. We introduced the BACDS model and proved that the minimum BACDS can achieve longer lifetime than the MCDS. We showed that the MBACDS construction is NP-hard and provided a distributed approximation algorithm to construct a BACDS in general graphs. We also analyzed the size of the resulting BACDS as well as its time and message complexities. We proved that the resulting BACDS is at most $(8+\Delta)opt$ size, where Δ is the maximum node degree and opt is the size of the MBACDS. The time and message complexity of our algorithm are $O(n)$ and $O(n(\sqrt{n}+\log n+\Delta))$, respectively. Our BACDS construction algorithm also considers the mobility issues to dynamically maintain the BACDS backbone. The simulation results show that the BACDS model can save a significant amount of energy and achieve up to 30% longer network lifetime than the MCDS in WSNs.

Chapter 4

Cross-Layer Scheduling for Urban Area WSN

This chapter presents a cross-layer scheduling scheme for urban area high density (UAHD) sensor networks. We first study the unique characteristics of UAHD sensor networks. We propose a novel solution to reduce sensor density in UAHD networks by partitioning a network into multiple layers of overlapped low-density subnetworks. Node densities among different layers are therefore balanced. We then propose an efficient Cross-Layer Power Aware Scheduling (CLPAS) scheme. CLPAS consists of three parts: Spanning Tree Partition (STP) topology control algorithm, Intersection MAC (I-MAC) protocol and Urban Emergency Service (UES) algorithm. STP is a distributed algorithm that can partition an UAHD sensor network in polynomial time. Our simulation results demonstrate that it achieves very close performance to the optimal algorithm in various scenarios. We also study the MAC protocol in an UAHD network. We show that adopting the traditional S-MAC at urban intersections incurs longer routing paths and lower power efficiency

network-wide. The I-MAC protocol is then designed for sensor communications around intersections. It employs an adaptive time division schedule to dynamically manage packet traffic based on the observations on recent traffic. Finally, handling urban emergencies is a critical issue in urban security. The UES algorithm provides differentiated services to give urgent packets higher priority. We conduct simulations to evaluate the performance of the CLPAS scheme in the Times Square area of New York City.

The rest of this chapter is organized as follows. Section 4.1 discusses some background and related work to place our work in context. Section 4.2 studies the density reduction in UAHD sensor networks and give examples. Section 4.3 presents the new cross-layer power aware scheduling (CLPAS) scheme. Section 4.4 gives the simulation results for the CLPAS scheme, and Section 4.5 concludes the chapter.

4.1 Related Work

In recent years urban security has become a critical and urgent issue in major cities [78, 79]. With the increasing possibility of terrorism attacks, disasters and accidents in metropolis, it is critical to provide early detection of suspicious activities or accidents in public areas and classification of biological or chemical attacks. Wireless sensors [80] have been widely considered as one of the best solutions for urban security [79, 82]. Recent advances in wireless communications and electronics have enabled the development of low-cost, multi-functional sensor nodes that are small in size and easy to deploy in urban facilities, such as street lamps, entrances of buildings, avenue intersections, parks, squares and urban drainage systems. A large number of tiny sensor nodes, each of which consists of sensing, detecting, data processing and communicating components, can be used to form a high density multi-functional urban area monitoring and detecting system. The sensors are

organized to capture continuous, real-time information, collect sensed data and propagate them to base stations. In this chapter, we refer to an urban area high density sensor network as an UAHD sensor network. In the rest of the chapter, if not specially mentioned, we assume that an UAHD network is a connected network.

In general, the main characteristics of urban area sensor networks can be summarized as follows.

- **Large volume and densely deployed sensors**

  Due to the special features of the urban terrain, the number of sensor nodes deployed in an urban area is typically very large. Moreover, sensors are often densely deployed to avoid sensor failure. As suggested in [81], a typical sensor network might have as many as 20 nodes per m^3. In an urban area, since sensors are usually attached to crowded urban facilities such as street lamps on narrow streets. the density of sensors deployed in an urban area is generally extremely high.

- **High packet collision and channel contention**

  Wireless sensor networks densely deployed in an urban area without adopting a specially designed scheduling scheme tend to have high packet collision, large channel contention overhead and continuous interference over wireless medium [79, 23], which will tremendously reduce the energy efficiency of the network.

- **Building interference**

  An urban terrain is typically composed of many regular building blocks with narrow streets and avenues crossing among them, which may block wireless communications. For example, in New York City, the narrow width of streets (in the order of 10 m's) and the height of buildings (in the order of 10^2 m's) can disconnect wireless communications among the sensors in adjacent streets.

- **City intersection bottlenecks**

  An intersection is a junction where several streets or avenues cross each other. As packets are routed along streets in an UAHD network, intersections naturally become bottlenecks for packet forwarding, unicasting or broadcasting. Besides, sensors located around intersections typically have more neighbors than sensors on streets. The problems of channel contention and packet collision are more critical at intersections of a city.

- **Power efficiency**

  Wireless sensors are powered with limited power supply. The huge number of sensors deployed in an urban area makes replacement of sensors or replenishment of their power resources expensive and unaffordable. Moreover, sensor nodes in urban sensor networks are especially vulnerable to power loss due to the requirements of high accuracy of monitoring sophisticated areas and time consuming tasks. Hence power conservation and power management are critical issues in urban area sensor network design.

- **Emergency response**

  Detecting emergencies and reporting accidents within a short response time are also very important issues. The emergency response time can be defined as the time between the occurrence of an alarm and the time the base station receives the information of this event. A well-designed scheduling algorithm for urban sensor networks should minimize the response time network-wide.

There has been much work in the literature on power aware protocols for traditional wireless sensor networks . Based on their functions, they can be roughly categorized into

topology control protocols, power aware MAC layer protocols and power aware routing algorithms. Some topology control protocols, such as [86], construct a virtual backbone from a connected dominating set (CDS) of a wireless sensor network. A backbone is a connected subset of sensors such that all sensors in the network are at most one hop away from the backbone. Other topology control protocols focus on maintaining network connectivity for a sensor network with difference transmission ranges [77]. Among MAC protocols, the most widely adopted power aware MAC layer protocol is the Sensor-MAC (S-MAC) [84]. The S-MAC lets neighboring nodes form virtual clusters to set up a common sleep schedule. Sensors are put into sleeping mode periodically. If a sensor belongs to two different clusters it follows both sleeping schedules simultaneously. Power aware routing algorithms find and maintain efficient routes for data communications in the network. For example, *Most Forward within Radius (MFR)* [18] chooses the next routing hop farthest away within the communication distance, thus finds a minimum hop-count routing path, while *Nearest with Forward Progress (NFP)* [19] attempts to choose the nearest node as the next forwarding node to minimize the energy required per routing task.

Although the above algorithms and protocols can address power scheduling issues reasonably well in traditional sensor networks, they are not quite suitable for UAHD sensor networks. Directly implementing them in UAHD networks would incur performance penalty due to high densities and special terrain features of UAHD networks. In this chapter we propose a novel cross-layer power aware scheduling (CLPAS) scheme for UAHD sensor networks. The key idea of CLPAS scheme is to divide an UAHD sensor network into overlapped layers with each layer having a bounded low density. It should be mentioned that CLPAS is not introduced to replace existing routing protocols or MAC layer protocols. Instead, its goal is to divide an UAHD network into low density sub-networks,

where existing routing and MAC layer protocols can be feasibly implemented.

In the rest of the chapter, we first discuss the characteristics of the UAHD networks. We then propose a novel approach which partitions an UAHD network into subnetworks with each subnetwork having a k-bounded node degree. We present the CLPAS scheme, which consists of three parts: Spanning Tree Partition (STP) topology control algorithm, Intersection MAC (I-MAC) protocol and Urban Emergency Service (UES) algorithm. CLPAS adopts a distributed STP algorithm as the topology control algorithm to partition an UAHD sensor network. We compare the performance of the STP with that of the optimal algorithm through simulations, and the results show that the STP achieves very close performance to the optimal algorithm in terms of the size of subnetworks and the number of layers in various scenarios. CLPAS also optimizes the performance of packet routing at urban intersections by providing an intersection MAC layer protocol (I-MAC). We show that adopting the traditional S-MAC at urban intersections incurs longer routing paths and leads to lower power efficiency network-wide. Different from the S-MAC, the I-MAC adopts an adaptive time division schedule to dynamically manage packet traffic based on the observations on recent traffic. Intersection sensors directly allocate time slots to neighbor sensors, which avoids overhearing and idle listening, in turn improves power efficiency. Finally, timely reporting urban emergent events, attacks and accidents is a critical issue in urban security. In CLPAS scheme we introduce Urban Emergency Service (UES) algorithm that provides differentiated services to give urgent packets higher priority. We conduct simulations to evaluate the CLPAS scheme for the Times Square area area in New York City. Fig. 4.1 shows the simulated map that is exactly scaled from the Manhattan map. Our results show that adopting CLPAS significantly improves network lifetime as well as data throughput in various communication scenarios such as unicasting, broadcasting and data collecting.

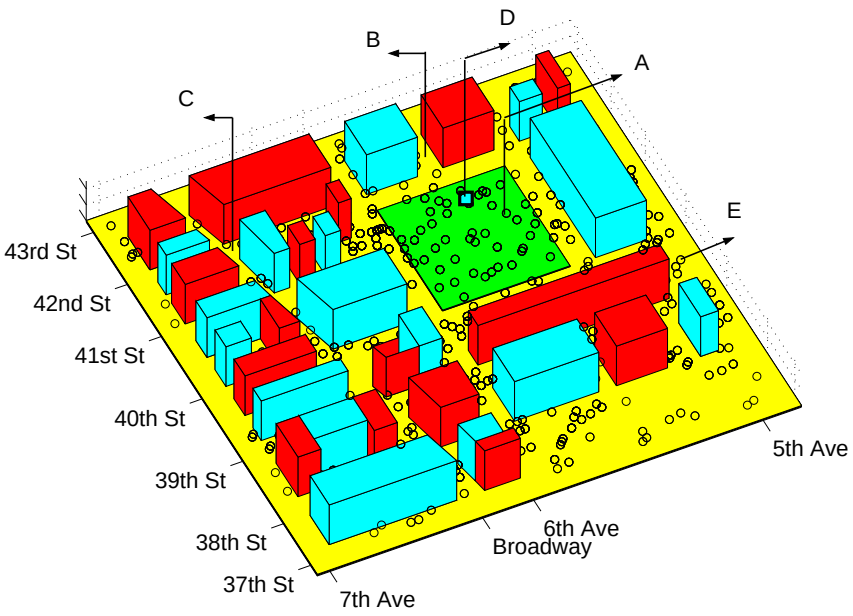


Figure 4.1: An urban area high density sensor network deployed in Times Square area of New York City. It extends north from the 37th street to the 43rd street, and extends east from the 7th avenue to the 5th avenue. Our simulations adopt this map that is exactly scaled from the Manhattan map. A: Open Area, B: Street, C: Intersection, D: Base Station, and E: Wireless Sensor.

4.2 Density Reduction in UAHD Sensor Networks

As wireless sensors find more and more applications, the density of sensor networks deployed in urban areas is dramatically increasing [81, 83]. A typical sensor network could have as high as 20 nodes per m^3 [81]. If considering that sensors typically have a transmission radius of 10 m [43, 44], each sensor in the network has at least 10^2 neighbors. Such a crowded neighborhood tremendously reduces the network power efficiency due to high packet collision, channel contention overhead, and continuous interference over wireless medium [79, 23]. The basic idea of our approach is to partition a sensor network into several overlapped subnetworks with each subnetwork having a low density. Fig. 4.2 (a) shows a connected UAHD network deployed in a 100x100 field with the sensor node degree as high as 12. We build a multi-layer structure in the network by constructing the maximum size k-bounded partition subnetworks. In order to reduce the packet collision and channel contention, in each subnetwork any node has a degree of no more than k, where k (> 1) is a constant defined by users. We first construct such a subnetwork connected with the base station. This subnetwork is referred to as layer 1. Additional subnetworks are constructed in the remaining network to connected with layer 1. These subnetworks are referred to as layer 2. Repeat the above procedure until all sensor nodes belong to a layer. Each subnetwork is an independent set with a bounded low density. Layers may geographically overlap with each other. Each layer adopts a different channel so that data communication among one layer does not interference with other layers. Fig. 4.2 (b) illustrates the structure of the network of Fig. 4.2 (a) after the partition ($k = 6$) in a 3D view.

This multi-layer structure has following advantages over a traditional hierarchical structure. First, in a traditional hierarchical structure, such as cluster based structures or CDS

structures [86], lower layers are controlled by upper layers, which makes upper layers become bottlenecks of data collecting and packet routing. In our structure, each layer is independent of each other in terms of MAC layer resource allocation, data packet routing and decision making. Secondly, in a traditional hierarchical structure the base station is only connected to the top layer, while in our structure the base station can be directly connected to multiple layers. This can efficiently shorten the time to broadcast packets from the base station and reduce the communication overhead among different layers. Thirdly, when there is a need for data communication among layers or when a lower layer is not directly connected to the base station, a layer can transmit packets to its upper layers through gateway nodes. As shown in Fig. 4.2 (b), a gateway node is a sensor that has neighbors in adjacent layers. Since the connectivity of an UAHD network ensures that layer 1 is connected to a base station, packets from any subnetwork can eventually reach the base station through gateway nodes. Note that gateway nodes are less likely to become bottlenecks, because the high network density enables a layer to have a large number of distributed gateways nodes, which greatly reduces the load on each gateway. Finally, multi-layer structure efficiently balances the usage of channel among sensors. The constrain k limits the maximum number of channels used in each layer. It in turn expends the space of reusing same channels in different layers as long as no collisions occur in adjacent layers. In practice k can be flexibly setup based on actual network topology, urban conditions, network traffics, sensor geography and maximum number of sensor channels.

In the next section, we will introduce approaches to partition an UAHD sensor network.

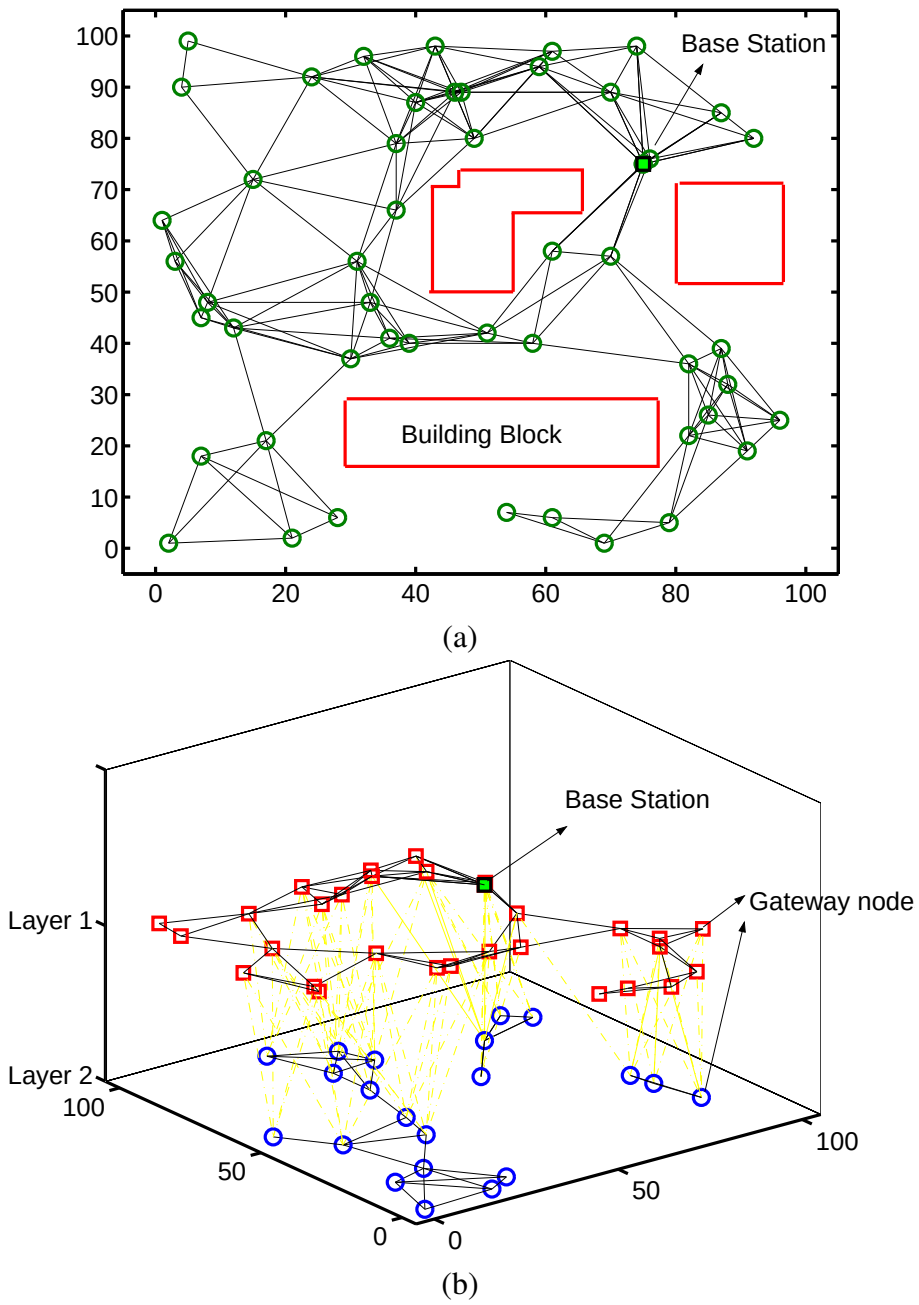


Figure 4.2: Partition a high-density sensor network into overlapped low-density subnetworks. (a) An UAHD sensor network has the sensor degree as high as 12. (b) Optimal partitions of the network in (a). The network is partitioned into two layers with each layer having a maximum sensor degree at most 6. The partitions are illustrated in a 3D view, where we omit the building blocks.

4.3 Cross-Layer Power Aware Scheduling

As discussed in Section 4.2, the optimal algorithm is too expensive to be implemented in power sensitive UAHD sensor networks. In this section we propose an affordable and efficient Cross Layer Power Aware Scheduling (CLPAS) scheme to manage the network as inter-connected low-density subnetworks where traditional protocols can be feasibly implemented. The CLAPS contains three components: Spanning Tree Partition (STP) topology control algorithm, Intersection-MAC (I-MAC) protocol and Urban Emergency Service (UES) algorithm. We will discuss them seperately next.

4.3.1 Spanning Tree Partition Algorithm

In this subsection we present a spanning tree partition (STP) algorithm to partition a network. The STP is a distributed algorithm that has polynomial time complexity. Let n denote the number of sensors in the network and p denote the maximum number of neighbors a sensor may have. In practice, p is a fairly small constant compared to n. The key idea of the STP is to construct spanning trees such that each node in these spanning trees has a node degree no more than constant k.

At the beginning, every node is colored white. A tree is spanning from base stations which are referred to as source nodes. The STP algorithm first colors at most k neighbors of each base station black. The spanning tree keeps growing until there are no more nodes that can be colored black. All these black nodes form a connected subnetwork that is the first layer. To form the second layer, the STP algorithm marks each node in the first layer, including base stations, as source nodes and grows spanning trees from them. The algorithm repeatedly does so until all nodes are assigned to a layer.

Table 1 describes the algorithm in more detail, where three colors are used: white, black

and gray. White nodes are available sensors that have not yet been assigned to a layer. Black nodes are sensors assigned in the layer currently under construction. When running the STP algorithm, to avoid conflicts, we color black nodes in those already constructed layers gray. For example, if we are constructing layer 2, all black nodes in layer 1 are currently colored gray. The purpose is to ensure that the subnetwork under construction does not violate the k degree constraint in this subnetwork. Note that although layer 2 is constructed from layer 1, we do not restrict the number of links among gray nodes and black nodes. For example, a gray may have more than k black nodes. This is because that in our scheme they do not have communication collisions or contentions with each other.

There are two types of *request* packets used in the STP algorithm: R_1 and R_2. R_1 is the request send from a black or gray node y to a white node x. It requests to color x black. On receiving R_1, this white node x needs to ensure that coloring it black will not make itself and all its one hop black nodes violate the k degree constraint. Hence x needs to send another type of request R_2 to all its black neighbors. Only if all its black neighbors agree to color x black, will x reply a *yes* to y.

The STP algorithm contains a partition procedure that takes an input k. This procedure generates spanning trees subject to constraint k from black and gray nodes. Initially, the STP algorithm colors the base stations as black nodes, and it repeatedly calls this procedure to partition the network until all sensor nodes are assigned to a partition. Fig. 4.3 illustrates the scenario that the STP partitions the network with $k = 3$. At this stage, layer 1 has been constructed and its nodes A, B, C and D are colored gray, and the STP is constructing layer 2. E and A are sending requests R_1 to H and G, respectively. To ensure the k degree constraint, H sends R_2 to F. When H receives a *yes* reply from F it replies *yes* to H and colors itself black.

Table 1: Spanning tree partition (STP) Algorithm

Partition Procedure (int k)
begin
Each black or gray node does:
repeat
δ = Number of its black neighbors;
if ($\delta < k$) **then**
Send a request R_1 to a white neighbor x;
if reply from x is yes **then**
Color x black, $\delta = \delta + 1$;
until $\delta = k$ or no neighbor is white;

Each black or gray node does:
On receiving a request R_2 from a white neighbor:
if $\delta > k$ **then** Reply yes; **else** Reply no;

Each white node does:
On receiving a request from a black or gray neighbor y:
$\overline{\delta}$ = Number of its black neighbors;
if ($\overline{\delta} < k$) **then**
Broadcast requests R_2 to all its black neighbors except y;
if all replies are yes **then**
Reply y with yes; Broadcast all its one hop neighbors' δs;
else Reply y with no;
else Reply y with no;
end

The STP Algorithm
begin
Given a connected graph and k;
Color all nodes white;
Color base stations black;
repeat
Call *Partition procedure(k)*;
Color all black nodes gray;
until no node is white;
end

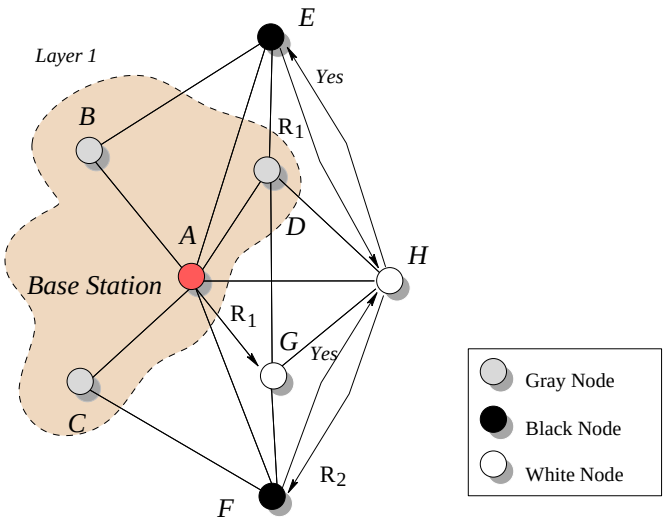


Figure 4.3: An example of constructing partitions by the STP algorithm. At this stage the STP is forming layer 2.

In this algorithm each white node is only colored black once. When it is colored, we only need to consider at most p one hop neighbors. Therefore, the time complexity is $O(pn)$, where n is the number of nodes in the network and p is the maximum node degree. The STP algorithm efficiently partitions a network into overlapped low-density layers. Each node only needs to communicate with its one hop neighbors. Each layer adopts an independent radio channel to avoid packet collisions in the UAHD sensor network. The STP algorithm is also scalable. It is easy to see that the number of layers only depends on the network density rather than the size of the network, which makes the algorithm feasible to be implemented in large cities. We conduct simulations to compare the performance of the STP algorithm and the optimal algorithm. As will be seen in Section 4.4, the STP algorithm not only has low complexity, but also achieves very close performance to that of the optimal algorithm in various scenarios.

4.3.2 Intersection-MAC Protocol

In Section 4.3.1 we proposed the STP algorithm to partition a sensor network. Due to the special characteristics of the urban terrain, in this subsection we will modify the MAC protocol at intersection sensors to further improve the overall network performance.

An intersection is a junction where several streets or avenues cross each other. In an UAHD sensor network, packets are routed along streets. As a result, intersections naturally become bottlenecks for forwarding. Fig. 4.4 shows an example of a city example of a city intersection crossed by Streets X and Y in an urban area. Intersection sensor D is deployed in the intersection. In Fig. 4.4 (a), the STP algorithm is partitioning the network with $k = 3$. At this time intersection sensor D already has three neighbors E, F and G. Sensor C cannot be assigned to layer 1, otherwise it violates the degree constraint. This is a typical problem that particularly occurs in an UAHD sensor network. The urban terrain forces sensors on streets to connect with sensors at the intersection. In our example, if there is no building H that blocks the communication between B and C, we would not have this problem. C is either linked to node D at layer 2 or makes a detour to bypass the entire building block. Either way will reduce the power efficiency of the network.

The idea of intersection optimization is to let intersection sensors act as traffic conductors that actively relay packets. For this purpose we locally increase the value of k at intersection sensors and modify their MAC layer protocols. In the example of Fig. 4.4 (b) we let $k \geq 4$ so that C can be connected to intersection sensor D. To avoid packet collision and channel contention at intersection sensors, we modify their MAC layer protocol. As discussed in Section 4.1, the widely used MAC layer protocol is the S-MAC which locally manages sensor synchronization and lets them follow periodical sleep-listen schedules [84]. In the S-MAC protocol, neighborhood sensors form virtual clusters to set up a

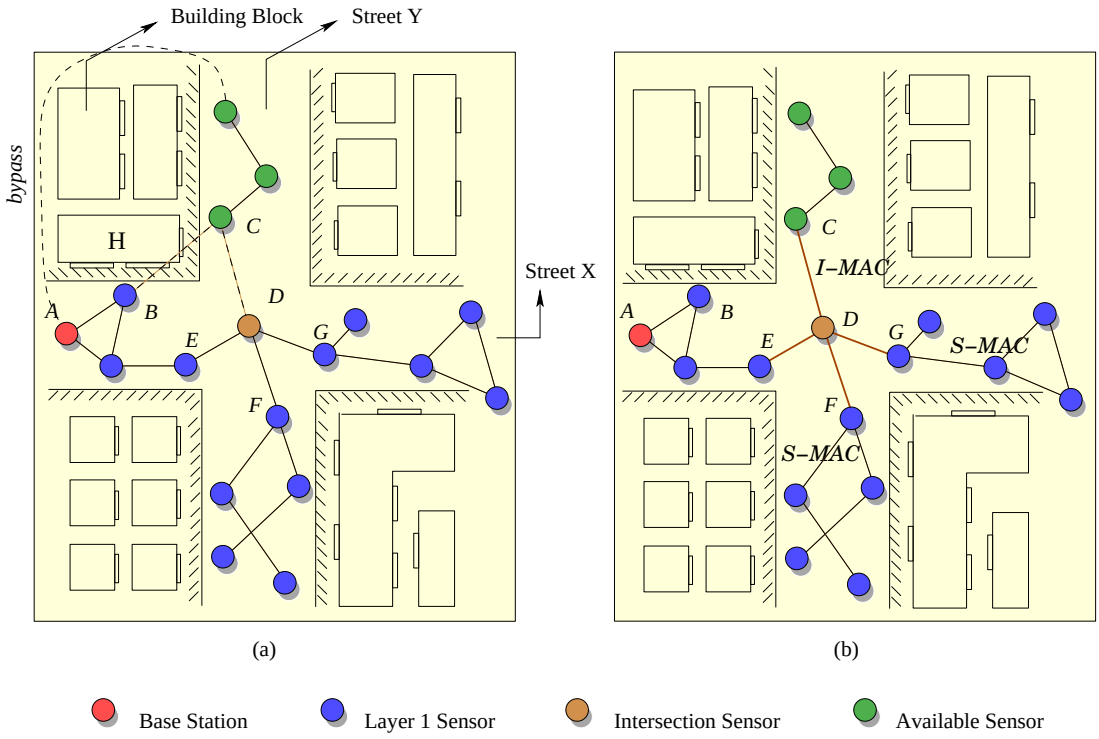


Figure 4.4: Intersections are bottlenecks in an UAHD network. (a) STP is partitioning the network with $k = 3$. Connection between B and C is blocked so packets from C have to either bypass the entire building block to the base station or be sent to intersection sensor D from a lower layer. Neither way is power efficient. (b) Adopting the I-MAC between intersection sensor D and its neighbors C, E, G and F to effectively solve the problem.

common sleep schedule using RTS (request to send) and CTS (clear to send) packets. The S-MAC is feasible to be implemented in traditional sensor networks and achieves good performance. However, it has disadvantages for intersection sensors due to the following reasons. First, in the S-MAC every packet must be acknowledged to prevent the receiving node from losing packets during sleep. Second, sleep and listen periods are predetermined and are constant in the S-MAC. An adaptive schedule that considers real-time traffic will be more feasible for intersection sensors.

We propose the I-MAC (Intersection-MAC) protocol for intersection sensors and their one hop neighbors. The I-MAC has two advantages. (i) Intersection sensors make their own decisions on the receive/sleep schedule. The schedule is broadcast by intersection sensors to all their one-hop neighbors. (ii) The I-MAC adopts an adaptive time division schedule. Time slots are assigned to neighbors of intersection sensors. The length of the time slot and the frequency of a neighbor are adaptive to the recent traffic. An intersection sensor keeps a small table to record the frequency of the packet traffic of its neighbors. The neighbors that recently have more packets to send are dynamically allocated more time slots. This way intersection sensors act more like traffic conductors that actively schedule sensors located at intersections. Since in UAHD sensor networks sensors are less likely to be mobile, keeping track of recent packet traffic of neighbors is feasible for intersection sensors. Note that the I-MAC is only implemented for MAC layer communications between intersection sensors and their one hop neighbors. The S-MAC is still adopted in the rest of the network. Fig. 4.5 gives an example that explains how the protocol works. It shows the MAC layer communication among an intersection sensor I and its two neighbors A and B. In the first phase P_0, I sends a beacon to synchronize its neighbors. (P_1 is the phase to provide emergency service to urgent packets, and we will discuss its details in Section 4.3.3.) Then

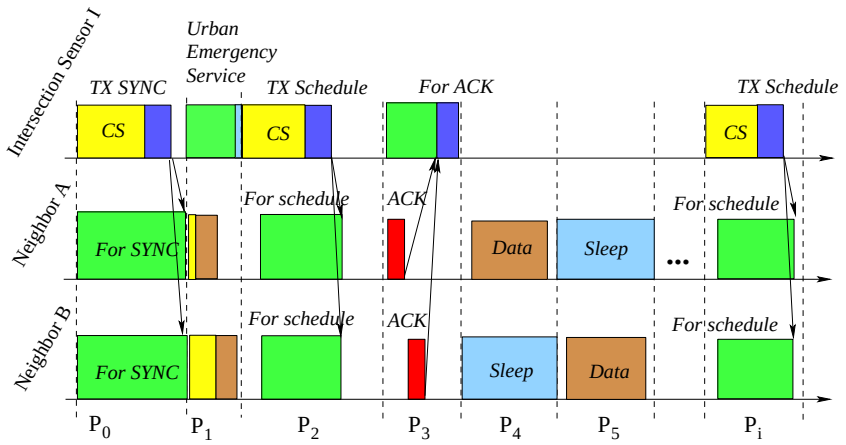


Figure 4.5: An example of the I-MAC protocol. It shows the MAC layer communication among an intersection sensor I and its two neighbors A and B, where CS and SYNC stand for Carrier Sensing and Synchronization, respectively. The details of Urban Emergency Service time slot (P_1) will be illustrated in Fig. 4.6.

a sleep/listen schedule is made by I and broadcast to A and B. To ensure the reception at A and B, I needs acknowledges from all neighbors. In the following phases $P_4, P_5, \ldots, P_{i-1}$, neighbor sensors A and B transmit or receive packets in the time slots allocated to them and turn to the sleep mode for the rest of the time. A sensor needs to turn on its radio transceiver even it has no packet to send or receive. Node I keeps a record of packet transmission rates of all its neighbors. If A has a higher packet transmission rate than B, A will be allocated more time slots in the next sleep/listen schedule. In phase P_i a new schedule is generated according to the observations of recent packet traffic and is broadcast to A and B.

4.3.3 Differentiated Services for Urban Emergencies

Urban emergencies are urgent events, accidents, terrorism attacks, disasters or occurrences that unexpectedly happen in an urban area and demand immediate actions. Detecting and reporting emergencies within a short response time is very important for UAHD

sensor networks. When detecting emergencies, sensor nodes send packets that contain the emergency type, location information, time stamp and sensed physical values such as temperature or humidity. Packets routed in an UAHD network are given different priorities according to how urgent they are. The more urgent an emergency is, the higher priority its packets have, therefore the quicker they should be delivered than packets that have a lower priority. For example, packets transmitted from a building that is vulnerable to collapse will take precedent over other low priority packets.

The method to classify the priorities of packets and provide different delivery services is referred to as a differentiated service algorithm [89]. Several differentiated service algorithms have been proposed in [87, 88]. Most of the algorithms rely on a centralized control and are not quite suitable for large scale networks like UAHD networks. In this subsection we propose an Urban Emergency Service (UES) algorithm to provide differentiated services in UAHD networks. UES algorithm assigns each packet a priority q ($1 \leq q \leq m$), where m indicates the most urgent emergency. Let r ($1 < r \leq m$) be a pre-defined constant. Without loss of generality, we define packets that have priorities no less than r as high priority packets. UES algorithm adopts two methods to provide differentiated services. (i) UES maintains a priority queue at each sensor node. Received packets are sorted in the queue in a descending order. Packets are transmitted in the order of their priorities. (ii) To avoid packets delay at intersections, UES algorithm lets the I-MAC algorithm periodically allocate Urban Emergency Service (UES) time slots. As shown in Fig. 4.5, an UES time slot is inserted as P_1. During P_1, high priority packets with priority $q \geq r$ are transmitted. Lower priority packets with $q < r$ (in the example of Fig. 4.5, they are P_4 and P_5) can be transmitted during the regular time slots later.

Fig. 4.6 gives an example of differentiated services between an intersection sensor I

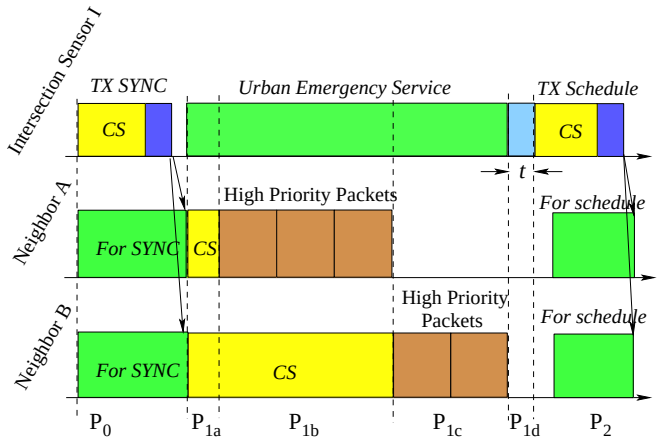


Figure 4.6: An example of differentiated services between an intersection sensor I and its two neighbors A and B. This figure illustrates the details of Urban Emergency Service time slot in Fig. 4.5.

and its two neighbors A and B, where t is a constant time that is used by intersection sensors to manage an UES time slot. As can be seen, during the emergency service time slot $P_{1a},\ldots,P_{1d}$, nodes A and B have two and three high priority packets to transmit, respectively. Each node backs off a random time and senses the carrier. In P_{1b} slot, node A begins to transmit all its high priority packets. When A finishes, B finds the carrier is available and transmits its packets in P_{1c}. In P_{1d} node I detects the carrier has been free for t time and completes the emergency service time slot. Note that UES time slots are periodically inserted so that high priority packets always have a chance to take precedent over lower priority packets. We conduct simulations to evaluate the performance of our differentiated service algorithm. The results in the next section demonstrate that UES algorithm can provide priority-based services in UAHD sensor networks.

4.4 Performance Evaluations

In this section we evaluate the performance of CLPAS scheme in terms of partition size, number of subnetwork layers, network lifetime, data throughput and emergency response time. In subsection 4.4.1 we give the details of how we set up our simulations. In subsection 4.4.2 we compare the performance of Spanning Tree Partition (STP) algorithm with the optimal algorithm in terms of the number of nodes in a partition and the total number of subnetwork layers after partition. Subsection 4.4.3 evaluates the network-wide performance of CLPAS scheme in the scenarios of broadcasting, unicasting and data collecting. Finally in subsection 4.4.4, we study the performance of differentiated services provided by Urban Emergency Service (UES) algorithm.

4.4.1 Simulation Setup

To realistically evaluate the performance of CLPAS scheme for UAHD sensor networks, we setup the simulations for Times Square area in New York City. Fig. 4.7 gives the simulation area shown on the Manhattan map, enclosed by the blue contour. This area extends north from the 37th street to the 43rd street, and extends east from the 7th avenue and Broadway to the 5th avenue. It covers an area of about $1,023,750$ ft^2. There are two reasons why we adopt this area for our simulations. (i) New York City is a metropolitan city that are prone to all types of attacks, accidents and disasters. Simulating CLPAS scheme in this city can directly reflect the performance of our approach in an actual major city. (ii) Times Square is a typical urban area in New York City. This is not only due to its importance in economy and politics, but also because of its special urban terrain. This area consists of 17 blocks, 26 intersections, an open area (the Bryant Park) and roughly 33 buildings. All these urban features impose a great challenge to a sensor network scheduling

Figure 4.7: The simulation area shown on the Manhattan map, enclosed by the blue contour.

scheme.

We scale the area into a 160 $\times$ 180 simulation area. Fig. 4.8 shows its top view with 300 sensor nodes randomly deployed. A base station is located in the Bryant Park. In the following simulations, broadcasting, data collecting and unicasting mean that packets are sent from the base station to all sensors, sensors relay packets to the base station and one sensor sends packets to another sensor in this network, respectively.

4.4.2 STP Performance

In this subsection we compare the performance of the STP algorithm with the optimal algorithm. We consider two performance metrics: partition size and number of subnetwork layers.

Partition Size. Given urban area sensor networks with different number of sensor nodes and transmission radii, we first compare the sizes of maximum partition of STP algorithm and the optimal algorithm. Fig. 4.9 (a), (b) and (c) show the results with different transmission radii. We can see that the longer the radius, the more one-hop neighbors a

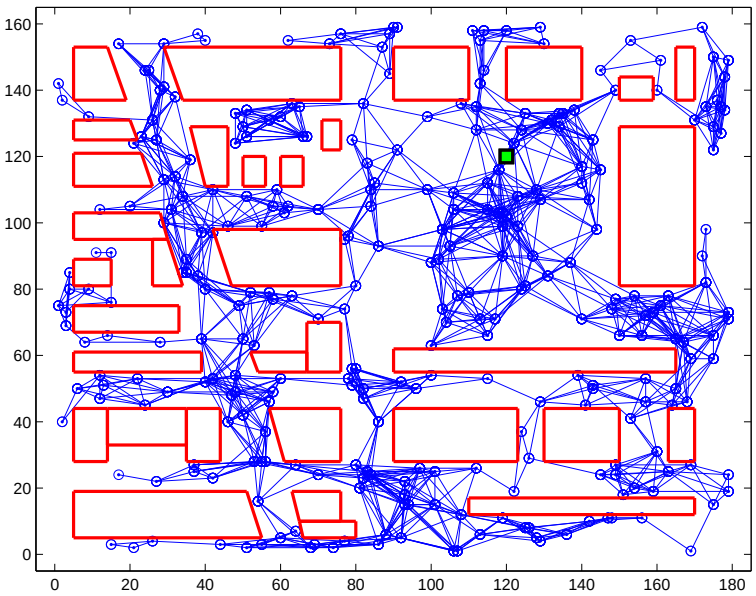
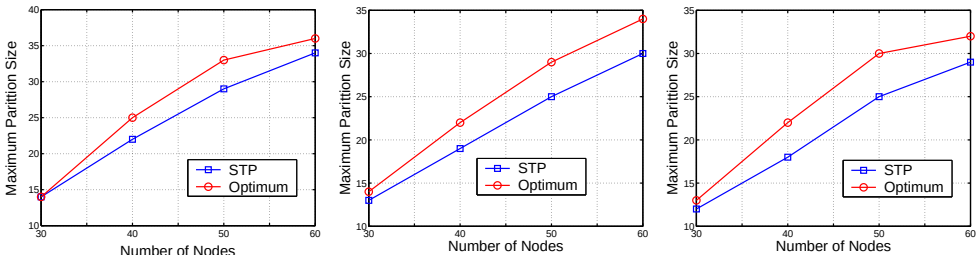


Figure 4.8: The times square area with 300 sensor nodes deployed.

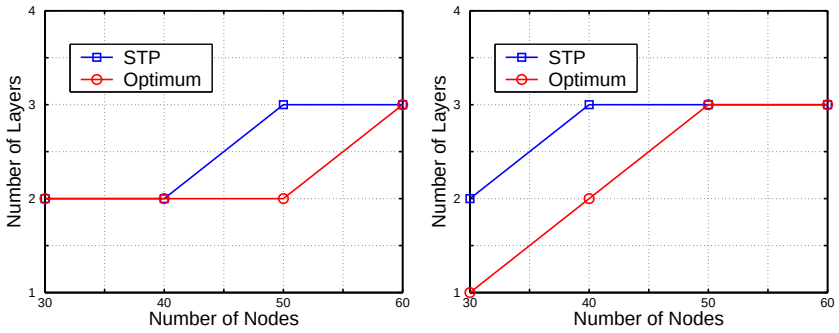
sensor may have. The simulation results illustrate that the STP algorithm achieves very close performance to that of the optimal algorithm. In the worst case, the subnetwork partitioned by the STP algorithm has at most 5 more nodes than that partitioned by the optimal algorithm.

Number of Layers. In partitioning an UAHD network subject to the maximum node degree constraint, the fewer number of layers an algorithm partitions, the better performance it achieves. Fig. 4.10 compares the total number of layers after partitions by the two approaches. We observe that for various network sizes with different densities, the number of layers partitioned by the STP algorithm is at most one more layer than that of the optimal algorithm.



(a) Radius= 10 (b) Radius= 13 (b) Radius= 15

Figure 4.9: We compare the partition size of UAHD networks by STP algorithm and the optimal algorithm. X axis shows the number of nodes in a network, Y axis is the network maximum partition size. (a), (b) and (c) have a sensor transmission radius of 10, 13 and 15, respectively.



(a) Radius= 10 (b) Radius= 15

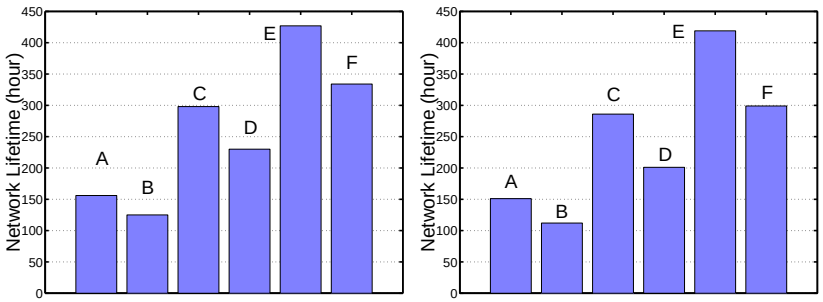
Figure 4.10: We compare the number of layers after partition in UAHD networks by STP algorithm and optimal algorithm. X axis shows the number of nodes in a network, Y axis shows how many layers after partition. (a) and (b) have a sensor transmission radius of 10 and 15, respectively.

4.4.3 Network-wide Performance

In this subsection we evaluate the network-wide performance of the UAHD sensor network with the CLPAS scheme. We consider three communication scenarios: broadcasting, data collecting and unicasting. UAHD networks are partitioned by the STP algorithm. We adopt the I-MAC at intersection sensors and the S-MAC at other sensors. For unicasting we adopt the MFR and NFP as routing protocols. For broadcasting and data collecting we choose flooding as the routing protocol. However, note that CLPAS is not restricted to these routing protocols. Other existing protocols can be easily implemented in our scheme.

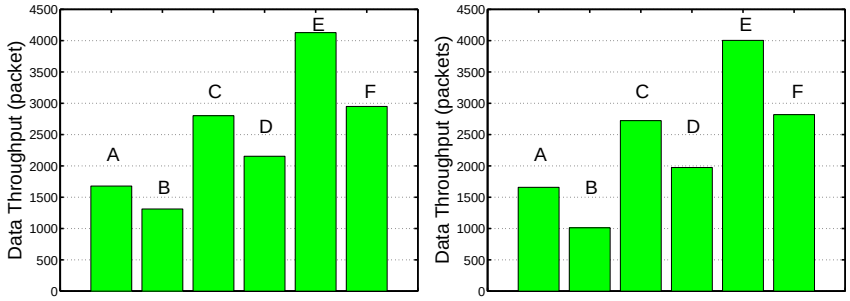
Broadcasting. In broadcasting scenario, we evaluate the network performance in terms of network lifetime and data throughput. Fig. 4.11 (a) and (b) show the network lifetimes of different network sizes with a transmission radius of 10 and 15. We observe that by adopting CLPAS, the network lifetime can be improved by up to 24%. This is because that CLPAS reduces the packet collision and achieves higher power efficiency. We also observe that, without adopting CLPAS, the longer the radius, the higher the network density, hence the shorter the network lifetime due to packet collision. On the other hand, we can see that a higher density has less impact on CLPAS, because it can efficiently partition the network and thus avoid collisions among sensors in high density areas. In our simulations, the base station keeps broadcasting packets. We measure the number of packets successfully received by sensors. As the network lifetime increases the data throughput also increases. We can see from Fig. 4.12 that higher data throughput is also achieved by CLPAS scheme.

Data Collecting. In data collecting, we let sensors randomly generate and propagate packets to the base station. At the beginning, the base station broadcasts a short packet to setup the data collecting path. Fig. 4.13 shows the network lifetimes for various network sizes and transmission radii. We observe that the total network lifetime of data collecting



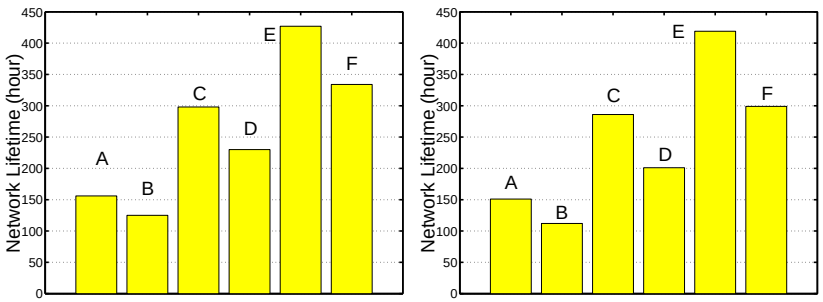
(a) Radius= 10 (b) Radius= 15

Figure 4.11: Network lifetimes in broadcasting scenario. In both (a) and (b): Columns A, C and E show the lifetimes of networks with sizes of 100, 200 and 300 nodes, respectively, with CLPAS scheme. Columns B, D and F show the lifetimes of networks with sizes of 100, 200 and 300 nodes, respectively, without CLPAS scheme. (a) and (b) stand for a transmission radius of 10 and 15, respectively.



(a) Radius= 10 (b) Radius= 15

Figure 4.12: Data throughput in broadcasting scenario. In both (a) and (b): Columns A, C and E show the data throughput of networks with sizes of 100, 200 and 300 nodes, respectively, with CLPAS scheme. Columns B, D and F show the data throughput of networks with sizes of 100, 200 and 300 nodes, respectively, without CLPAS scheme. (a) and (b) stand for a transmission radius of 10 and 15, respectively.



(a) Radius= 10 (b) Radius= 15

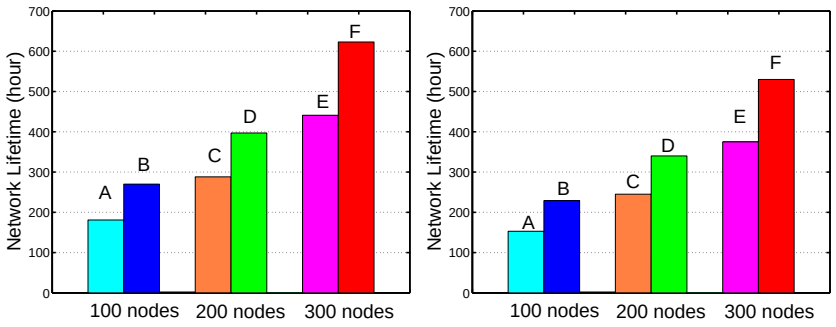
Figure 4.13: Network lifetimes in data collecting scenario. In both (a) and (b): Columns A, C and E show the lifetimes of networks with sizes of 100, 200 and 300 nodes, respectively, with CLPAS scheme. Columns B, D and F show the lifetimes of networks with sizes of 100, 200 and 300 nodes, respectively, without CLPAS scheme. (a) and (b) stand for a transmission radius of 10 and 15, respectively.

is prolonged by up to 29%, due to the reduced packet collisions by CLPAS.

Unicasting. In unicasting, we adopt two traditional protocols MFR and NFP as the routing protocols. We let sensors randomly send packets to destination sensors through multi-hop relay. We evaluate the improvement of network lifetimes by CLPAS scheme. Fig. 4.14 (a) and (b) adopt MFR and NFP, respectively. We can see that CLPAS improves network lifetime by up to 23%. We can also observe that the lifetime achieved in Fig. 4.14 (a) is longer than that in (b), which is due to the nature of MFR and NFP protocols.

4.4.4 UES Performance

In this subsection we evaluate the performance of UES algorithm in UAHD sensor networks. We compare the average packet response time (*rs*) of routing connections with different priorities. There are three different priority levels: high, middle and low in our simulations. Emergent events occur randomly. Sensor nodes choose their priorities according to the emergent event they sense, and they encapsulate the priority information into



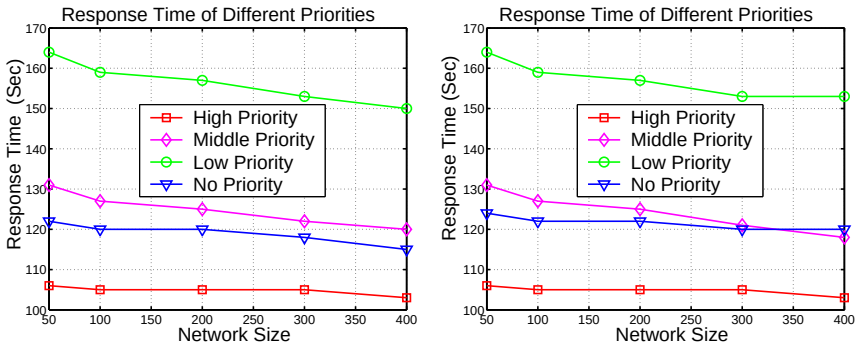
(a) MFR Protocol (b) NFP Protocol

Figure 4.14: Network lifetimes in unicasting scenario. In both (a) and (b): Columns A, C and E show the lifetimes of networks without CLPAS scheme. Columns B, D and F show the lifetimes with CLPAS scheme. (a) and (b) adopt MFR and NFP protocols, respectively. The number of sensors in networks are marked at X axis.

the header of each packet. The *rs* is calculated as the length of the time period between a packet is sent and the time the base station receives it. Fig. 4.15 (a) and (b) adopt MFR and NFP routing protocols, respectively. To make comparisons we also conduct simulations for networks with "no priority" packets and calculate their average *rs*. As can be seen in Fig.4.15, high priority packets have shorter average *rs* than low priority packets, which indicates that UES can provide differentiated services for high priority connections. We also observe that *rs* decreases as the number of sensors increases in the network. This is because that a larger number of sensors enables a routing protocol to set up direct routing paths more easily to destination sensors, therefore reduces the response time.

4.5 Summaries

In this chapter we have proposed a cross-layer solution to provide power efficient scheduling for Urban Area High Density (UAHD) sensor networks. we first discussed the unique



(a) MFR Protocol (b) NFP Protocol

Figure 4.15: Average response time of different priority packets in UAHD sensor networks. (a) and (b) adopt MFR and NFP routing protocols, respectively. Packets are assigned with three different priorities: High, Middle and Low.

characteristics of UAHD sensor networks and gave a method to reduce sensor density in UAHD networks by partitioning a network into multi-layers of overlapped low-density subnetworks. We introduced our CLPAS scheme to efficiently partition and schedule the network. CLPAS contains three parts: the STP topology control algorithm, the I-MAC MAC layer protocol and the UES differentiate service algorithm. We conducted simulations to evaluate the performance of the scheme for Times Square area in New York City. The results demonstrate that CLPAS scheme achieves good performance in terms of network lifetime, data throughput, power efficiency and differentiated services.

Chapter 5

Battery-Aware Hot Spot Covering for WMN

This chapter presents the battery-aware hot spot covering algorithm for WMNs. As more and more outdoor applications require long-lasting, high energy-efficient and continuously-working WMNs with battery-powered mesh routers, it is important to maintain network activities for a long lifetime with high energy efficiency. In this chapter, we introduce a mathematical model on battery discharging duration and lifetime for WMNs. We also present a battery lifetime optimization scheduling algorithm (BLOS) to maximize the lifetime of battery-powered mesh routers. Based on the BLOS algorithm, we further consider the problem of using battery-powered routers to monitor or cover a few hot spots in the network. We refer to this problem as the Spot Covering under BLOS Policy problem (SCBP). We prove that the SCBP problem is NP-hard and give an approximation algorithm called the *Spanning Tree Scheduling* (STS) to dynamically schedule mesh routers. he time complexity of the STS algorithm is $O(r)$ for a network with r mesh routers. Our simulation results show that the STS algorithm can greatly improve the lifetime, data throughput and

energy consumption efficiency of a WMN.

In this chapter we assume mesh routers are equipped with single radio transceiver. We will discuss battery-awareness for multiple radio transceiver mesh routers in next chapter. The rest of this chapter is organized as follows. In Section 5.1 we discuss some background and related work to place our work in context. We study the mathematical battery model for WMN in detail in Section 5.2. Section 5.3 presents the battery lifetime optimization scheduling algorithm (BLOS) to maximize battery lifetime. We introduce the Spot Covering under BLOS Policy problem (SCBP) in Section 5.4, and prove its NP-hardness. Then we give an approximation algorithm to schedule the network in Section 5.5. Finally, we present our simulation results in Section 5.6, and concluding remarks in Section 5.7.

5.1 Related Work

Recently WMNs have emerged as a flexible, low-cost extension to the wired network infrastructure [67, 68, 69]. A WMN is a hybrid network which consists of a mix of fixed routers and mobile clients interconnected via access points [67, 27]. For example, in MIT's roofnet WMN [74], a neighborhood can easily build a community WMN by setting up a few mesh routers with flexible mesh connectivities among houses to support distributed storage, data access, and video streaming. The huge market demand on high-speed and low cost wireless access services has greatly accelerated the development of WMNs. Among the applications of WMNs, one of the most important applications is the outdoor WMNs. Outdoor WMNs are usually setup in Disneyland, outdoor assemblages and stadiums to reduce the cost of setting up Ethernet cables [68]. Outdoor WMNs are also set up to provide connections in urban and country areas. For example, Google proposes to provide a free wireless Internet service for ubiquitous wireless connection in the urban area

of Mountain View, CA [72]. These mesh routers are deployed on street lamps, entrances of buildings, avenue intersections, parks, squares and building tops. The mesh routers together with wireless mesh clients, such as personal wireless devices, form a high density multi-functional urban area wireless communication system [69, 66]. On Long Island, New York, a county-wide WMN is also planned to be deployed by the government in Suffolk County in partnership with New York State Center of Excellence in Wireless and Information Technology (CEWIT) located at Stony Brook University [73]. Since these outdoor WMN applications use battery-powered mesh routers, efficient battery energy consumption is critical in these networks.

In general, a WMN is composed of three components: access points (AP), mesh routers and mesh clients [67, 71]. Fig.5.1 illustrates the architecture of a WMN. Unlike a traditional MANET, which is an isolated wireless network, the WMN architecture introduces a hierarchy with wireless routers communicating between mesh clients and APs [69]. A typical WMN usually has 30 to 100 mesh routers. The fixed APs are wired to connect to the Internet to provide high-bandwidth connections to the Internet backbone. The meshing among wireless routers and APs creates a wireless *backhaul* communication system [67]. The backhaul provides each mobile client with a limited number of entry points connected to the Internet [67]. These entry points, along with the APs, are usually referred to as *Hot Spots*. As the middle layer between the APs and mesh clients, mesh routers must cover all these hot spots. Mesh clients have more varieties of devices compared to mesh routers. These devices can be laptops, tablet PCs, PDAs, IP phones, RFID (Radio Frequency ID) readers, BACnet (Building Automation and Control networks) controllers, and many other types of widely used wireless devices.

Nowadays the batteries on most mesh routers can work for at most a few hours. As an

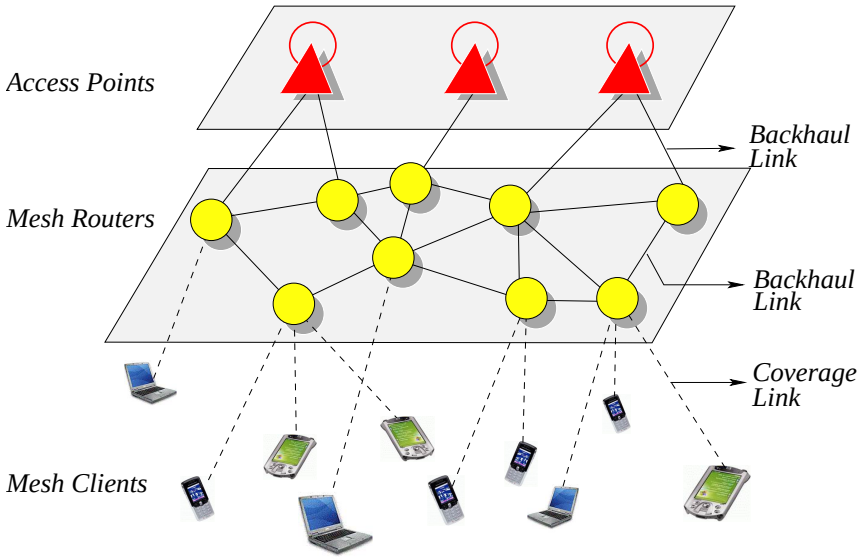


Figure 5.1: Architecture of a WMN.

example, the HotPort [70] series outdoor mesh router can continuously work for about two hours on battery. On the other hand, most outdoor applications, such as Disneyland, require a WMN with a fairly long lifetime. Improving battery performance in mesh routers can greatly improve the overall network communication performance. Thus, carefully scheduling and budgeting battery energy in WMNs has become an urgent and important issue in WMN design.

5.2 Battery Model for Single Transceiver Mesh Router

In this section we first briefly analyze and compare existing battery models. Then we introduce our battery model based on the scenario of epoch time discharging and recovery. We also give a method to simplify the computation of discharging loss in the model. Mathematical battery models were introduced in Chapter 2 and Chapter 3 for MANETs

and WSNs, respectively. In those battery models the battery lifetime is divided into a sequence of discrete time slots with a fixed slot length. This model can effectively capture the effect of battery discharging and recovery. However mesh router need a battery model which considers optimizing the battery lifetime based on the relationships among the discharging time, recovery time and overhead to switch the device from active to idle. In this section, we give a battery model based on continuous epoch time discharging and recovery for optimization in mesh router scheduling.

Consider the scenario where a battery is turned active for δ_i time, and then turned idle for τ_i time ($i = 1, 2, \ldots$). The active-idle period is repeated until the battery dies. Note that by turning a battery into idle, we let it "sleep" with very low current on it. During its idling the battery recovers its over-charged capacity.

In our battery model, battery is discharged in each duration i with length δ_i, where δ_i may not be equal to δ_j if $i \neq j$. α is the initial battery capacity. A duration δ_i is called an epoch. We use I_i, α_i, α_i' and ζ_i to denote the discharging current through the battery, the battery capacity at the beginning of the i_{th} epoch, the battery capacity at the end of the i_{th} epoch, and the discharging loss in the i_{th} epoch, respectively. We use T to denote the entire lifetime of the battery. An epoch δ_i is followed with a recovery period of length τ_i. Without loss of generality, we assume that the discharging current I is a constant in a certain epoch.

The condition of a battery at the i_{th} epoch is measured by its discharging loss at that time. A high discharging loss indicates a "fatigue" battery which needs some recovery, while a battery with low discharging loss is well recovered. Intuitively, an energy-efficient scheduling algorithm should always choose routers with well recovered batteries. Therefore, a good battery model should be able to calculate the discharging loss at any epoch.

The following analytical model can be used to compute the battery discharging loss

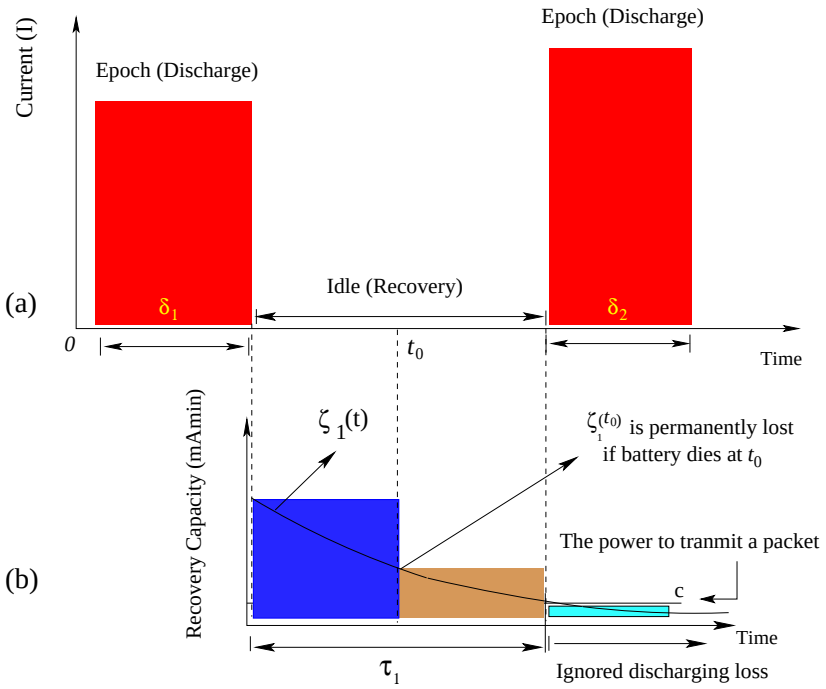


Figure 5.2: Battery discharging in different epochs. (a) Discharging of a battery in δ_1 and δ_2 epochs. Battery is idle between epochs. (b) The capacity $\zeta_1(t)$ is discharged in epoch δ_1 and recovered gradually after that. $\zeta_1(t)$ after time $\delta_1 + \tau_1$ is ignored. If this battery dies at t_0, $\zeta_1(t_0)$ is permanently lost.

at an epoch. The model that computes the energy dissipated by the battery during the i_{th} epoch $[\bar{t}, \bar{t}+\delta_i]$ is

$$\alpha_i - \alpha_i' = I_i \times F(T, \bar{t}, \bar{t}+\delta_i, \beta) \tag{5.1}$$

where

$$F(T, \bar{t}, \bar{t}+\delta_i, \beta) = \delta_i + 2\sum_{m=1}^{\infty}\left[\frac{e^{-\beta^2 m^2 (T-(\bar{t}+\delta_i))} - e^{-\beta^2 m^2 (T-\bar{t})}}{\beta^2 m^2}\right]$$

This model can be interpreted as follows. The dissipated energy during the i_{th} epoch is $\alpha_i' - \alpha_i$ in (5.1). It contains two components: The first term, $I_i \times \delta_i$, is simply the energy consumed in the device during $[\bar{t}, \bar{t}+\delta_i]$. The second term, $2I_i \times \sum_{m=1}^{\infty}\left[\frac{e^{-\beta^2 m^2 (T-(\bar{t}+\delta_i))} - e^{-\beta^2 m^2 (T-\bar{t})}}{\beta^2 m^2}\right]$ is the amount of battery discharging loss in the epoch. It can be seen that the discharging loss decreases as the lifetime T increases. Next we show how the model in (5.1) can be used to calculate the discharging loss at a given epoch.

As defined earlier, ζ_i is consumed in the i_{th} epoch and recovered in the next τ_i time. Clearly,

$$\zeta_i(t) = 2I_i \times \sum_{m=1}^{\infty}\left[\frac{e^{-\beta^2 m^2 (t-(\bar{t}+\delta_i))} - e^{-\beta^2 m^2 (t-\bar{t})}}{\beta^2 m^2}\right] \tag{5.2}$$

where $\zeta_i(t)$ is the residual discharging loss at time t. It should be mentioned that discharging loss $\zeta_i(t)$ is only a potential type of energy. For example, in Fig. 5.2, if the battery dies at time t_0, the battery permanently loses the energy $\zeta_i(t_0)$.

As can be observed, the recovery of $\zeta_i(t)$ continues from $\bar{t}+\delta_i$ to ∞. In practice, we can simplify the computation of ζ_i as follows. Assume c is a fairly small constant, which is the energy to transmit a packet. By observing (5.2), if $\zeta_i(\tau_i)$ is less than c, we can ignore the discharging loss after time $\bar{t}+\delta_i+\tau_i$. Thus, after τ_i time of recovery, the battery can be

considered to be well-recovered.

In summary, in this section we introduce a battery model based on the δ epoch time discharging and τ time recovery scenario. We also give a method to simplify the computation of discharging loss ζ. Next we will apply this model to battery lifetime optimization in WMNs.

5.3 Battery Lifetime Optimization Scheduling (BLOS)

As discussed earlier, given a battery with initial capacity α discharged under current I from 0 to δ_1, the battery capacity drops nonlinearly during time $[0,\delta_1]$, and afterward it gradually recovers the capacity to the value of $\alpha - I \times \delta_1 - \zeta(t)$, where $\zeta(t)$ is the discharging loss at time $\delta_1 + t$. By periodically recovering the battery, we can reduce $\zeta(t)$, and in turn prolong the battery lifetime. Fig. 5.3 gives an example that shows the simulated lifetime prolonged by considering the battery recovery. In this case we assume the battery capacity is $\alpha = 4.5 \times 10^4 mA\text{min}$ and $\beta = 0.4$, which are typical values of a chemical battery [10]. The discharging current is $I_d = 900mA$. Under the greedy mode, the battery is continuously discharged until the battery dies. The total lifetime is 116 minutes. In the battery-aware mode, the battery is discharged in each epoch, and recovered in the next recovery period. The total working time is increased by 14.7%.

Now given a battery, in order to optimize the battery lifetime we need to determine when an epoch should start and how long the epoch should last. We adopt an iterative way to find an optimal scheduling policy for battery-powered mesh routers.

First consider a battery being discharged in $[\bar{t},\bar{t}+\delta_i]$. After that period it takes τ_i to fully recover the discharging loss. From (5.2) we know the length of recovery time τ_i depends only on δ_i for given α, I and β. Therefore, there is a function to calculate τ_i, and we call it

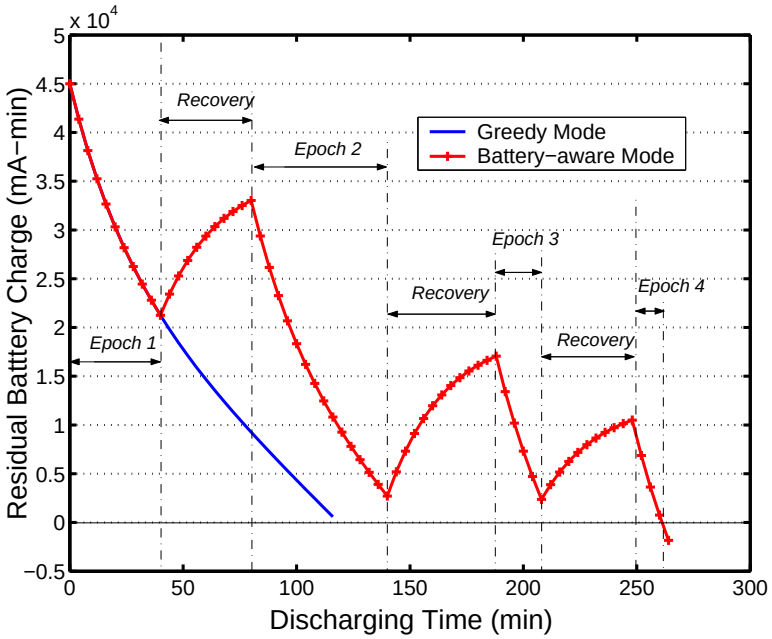


Figure 5.3: Simulated lifetime under the greedy mode and battery-aware mode. The lifetime under the greedy mode is 116 minutes. Under the battery-aware mode, the battery is discharged in each epoch, and recovered in the subsequent recovery time. The total working time is increased by 14.7%.

GetTau. That is,

$$\tau_i = GetTau(\alpha, I, \beta, \delta_i)$$

Under the greedy mode, where the battery is continuously discharged under current I until it dies, we define the total lifetime function as

$$Lifetime(\alpha, \beta, I)$$

This lifetime function can be easily obtained from (5.1). Now let's consider the battery discharging behavior. Our goal is to maximize the total battery discharging time. We assume that the mesh router is turned active n times. Let δ_i be the length of the i_{th} active time, $1 \leq i \leq n$. The problem can be formulated as: Given an initial battery capacity, how to choose the lengths of δ_1, δ_2, ..., δ_n such that the battery has the longest working time.

In mathematical terms, let $\overline{T}_i = \sum_{j=i}^{n} \delta_j$. A policy P is defined as a schedule for a router. P describes the lengths of time $\delta_1, \tau_1, \delta_2, \tau_2, \ldots$ until the battery is used up. An optimal policy is a policy by which $\overline{T}_1 = \sum_{i=1}^{n} \delta_i$ is maximized. Let α_i be the residual charge at the beginning of the i_{th} active time. $\overline{T}_i$ depends on α_i and the policy P. Given α_i, let $\overline{T}_i$ under the optimal policy be

$$\overline{T}_i = P_i(\alpha_i) \tag{5.3}$$

Then what we need to find is $P_1(\alpha_1)$.

Suppose that $P_i(\alpha_i)$ has been found, that is, for any given α_i, we know the optimal lengths of δ_i, δ_{i+1}, ..., δ_n such that $\overline{T}_i$ is maximized. Now we want to find $P_{i-1}(\alpha_{i-1})$.

Define the overhead to switch between active and idle is ε. Note that

$$\overline{T}_{i-1} = \delta_{i-1} + \overline{T}_i \tag{5.4}$$

Also note that if α_{i-1}, δ_{i-1} and τ_{i-1} are given, α_i is determined and can be written as

$$\alpha_i = f(\alpha_{i-1}, I, \delta_{i-1}, \tau_{i-1}) - \varepsilon \tag{5.5}$$

where function $f(\alpha_x, I, \delta_x, \tau_x)$ describes the residual battery energy after being discharged for δ_x time under current I and being recovered for τ_x time.

In this case $\overline{T}_{i-1}$ can be maximized by adopting the optimal policy for the $\overline{T}_i$ we have already found. From (5.4), (5.3) and (5.5), we obtain the maximum value of $\overline{T}_{i-1}$

$$\overline{T}_{i-1} = \delta_{i-1} + P_i(f(\alpha_{i-1}, I, \delta_{i-1}, \tau_{i-1}) - \varepsilon) \tag{5.6}$$

Note that (5.6) is only a function of α_{i-1} and δ_{i-1}. By varying δ_{i-1}, the maximum value of $\overline{T}_{i-1}$ under a given α_{i-1} can be found, which is the $P_{i-1}(\alpha_{i-1})$ we want to find. In practice, we can increase δ_i by a constant $\overline{\delta}$ step by step. Since after each τ_i time recovery, the battery is well-recovered, function f has a very simple form

$$\alpha_i = f(\alpha_{i-1}, I, \delta_{i-1}, \tau_{i-1}) = \alpha_{i-1} - I * \delta_{i-1}$$

Now we are in the position to describe the Battery Lifetime Optimization Scheduling (BLOS) algorithm. As defined earlier, n is the number of epochs during the lifetime of a battery. Clearly, $\overline{T}_1$ is not maximized when $n = 1$, because this is the greedy mode. As n increases, the battery periodically gets recovery, hence $\overline{T}_1$ is also increased. However, this

increasing is not monotonic because the accumulation of overhead ε also increases. The accumulated overhead in turn reduces $\overline{T}_1$. Therefore, there must exist an n such that $\overline{T}_1$ obtains its maximum value. The BLOS algorithm is used to find the optimum $\overline{T}_1$. The algorithm can be described as follows. Initially, we let $n = 1$, and calculate the optimum policy for the given n. Each step we increase n by 1, and calculate the new policy for this n until there is an optimum $\overline{T}_1$ with a peak value.

The BLOS algorithm employs an iterative approach to finding $P_1(\alpha_1)$ for a given n. We call it *GetBlos*. Table 1 gives the details of the *GetBlos* procedure. At the beginning, we find $P_n(\alpha_n)$. In the subsequent steps, $P_i(\alpha_i)$ can be determined according to the results of $P_{i+1}(\alpha_{i+1})$. This way we can finally find $P_1(\alpha_1)$. Since in the last active time the battery works until exhausted and there is no policy involved, it can be simply obtained from the battery model.

Finally, we give the complete BLOS algorithm based on procedures *GetBlos*, *GetTau* and $Lifetime$. Table 2 describes the algorithm in pseudo-codes. From $n = 1, 2, \ldots$, procedure *GetBlos* is called to calculate the optimal lifetime $\overline{T}_1$ for the given n, then n is increased until a peak value of $\overline{T}_1$ is obtained. The policy P at this n is the optimal policy we want. Finally we can use *GetTau* function to specify each τ_i for δ_i, to ensure the battery is well recovered after δ_i time discharging.

5.4 Hot Spot Covering Under BLOS Policy

In Section 5.3 we gave an optimal scheduling algorithm to maximize the lifetime of mesh routers. In this section we consider the problem of using battery-powered mesh routers to monitor or cover hot spots. Our goal is to let mesh routers all follow the optimal battery active-idle policy, while keeping all hot spots covered for as long as possible. We

Table 1: Finding optimal policy P for a given n

```
Procedure GetBlos (α, i)
begin
  if (i = n)
    begin
      δ_n = lifetime(α, β, I);
      return δ_n;
    end
  else
    begin
      T = 0;
      for j = 1 to ⌊α / (I × δ̄)⌋
        begin
          calculate δ_{i+1}, δ_{i+2}, ..., δ_n
            by calling GetBlos (α − j × δ̄ × I − ε, i + 1);
          T_i = Σ_{k=i+1}^{n} δ_k + j × δ̄;
          if T < T_i then T = T_i;
        end
      return T;
    end
end
```

Table 2: Battery lifetime optimization scheduling (BLOS)

```
Procedure BLOS
begin
    n = 0;
    T = 0, T̄_1 = 0;
    repeat
      T = T̄_1;
      n = n + 1;
      calculate T̄_1 by calling GetBlos(α, n);
    until T̄_1 < T;
    calculate τ_i by calling GetTau(α_i, I, β, δ_i),
     for i = 1, 2, ..., n;
end
```

call this problem the *Spot Covering under BLOS Policy* problem, and refer to it as the *SCBP* problem.

Like many other covering problems, the SCBP problem is NP-hard. In the following we give a proof of it. We show that even in the simplest case in which the optimal policy is to "use till exhausted," the decision version of the SCBP problem is NP-hard.

The decision version of the SCOP problem can be formally written as follows. Given a set of mesh routers and a set of spots, where each mesh router may cover several spots and each spot may be covered by several mesh routers. If all mesh routers have the same battery life T, does there exist a schedule by which all spots are covered in $[0,t']$, where t' is a positive constant? Such a schedule is called a *valid* schedule. Note that since all mesh routers, by the optimal policy, must work till the battery is exhausted after turned on, the only thing we can control is when to turn on each mesh router.

Lemma 3 *The decision version of the SCBP problem is equivalent to the* Subset Partition *problem (SP).*

Proof. First we give some properties of a valid schedule. Since all spots must be covered at time 0, some mesh routers must have been turned on at time 0 and these routers must collectively cover all spots. This can be seen without difficulty. Now suppose under a valid schedule, a mesh router is turned on at time t where $0 < t < T$. We can safely delay the turn on time of this router to T, since all spots that it covers must be covered during $[t,T]$ by the mesh routers turned on at time 0, and all spots it covers during $[T,T+t]$ are still covered by itself. As a result of this fact, if there is a valid schedule, there will be a valid schedule by which all mesh routers are turned on at times which are multiples of T. Therefore, the problem reduces to partitioning the mesh routers into groups, where each

group should collectively cover all spots and the number of groups multiplied by T must be larger than t'.

Thus the decision version of the SCBP problem is equivalent to the following problem, which we call the *Subset Partition* problem (SP): Given a whole set (the spots) and some subsets (the mesh routers), can the subsets be partitioned into k groups where each group of subsets covers all the elements in the whole set? ■

Now we have proved that the SCBP problem is equivalent to the SP problem. Next we show that the SP problem is NP-hard, and as a result, the SCBP problem is NP-hard.

Lemma 4 *The SP problem is NP-hard.*

Proof. Consider the *Vertex Covering* problem which is known to be NP-hard [76, 75]: Given a graph and k colors, can each vertex be given a color such that no adjacent vertices have the same color? The SP problem can be shown to be NP-hard as follows.

Given any instance of the VC problem G, construct an instance of the SP problem in 3 steps.

1. For each edge E_i in G, create an element called e_i.

2. For each vertex V_a in G, create a subset called S_a. Subset S_a contains element e_i if V_a is incident to E_i in G. Note that at this time each element is in exactly 2 subsets, i.e., covered by two subsets.

3. Then for each e_i, create $k-2$ identical subsets s_1^i, s_2^i, ..., s_{k-2}^i, where each of these subsets contains only one element which is e_i.

Note that now each element is covered by exactly k subsets. If all the subsets we have created can be partitioned into k groups where each group collectively covers all the

elements, the subsets covering the same element must belong to different groups. Give color C_j to vertex V_a if subset S_a is in the j_{th} group in the partition. Apparently, two vertices V_a and V_b will be given different colors if they are adjacent to each other in graph G. As a result, the partition determines a k-coloring in the VC instance. ■

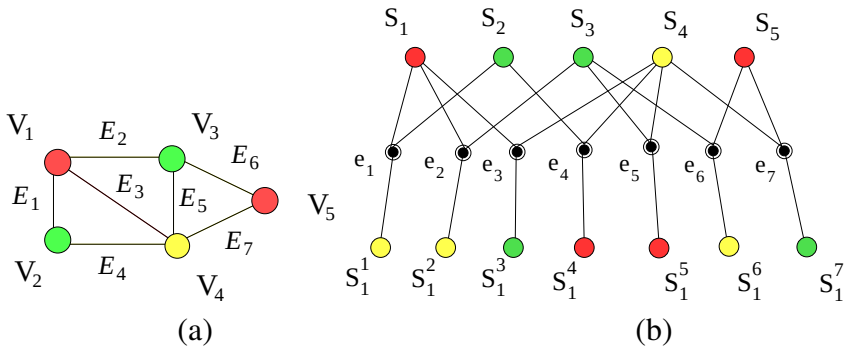


Figure 5.4: An example to show the SP instance for a graph G. (a) Graph G. (b) SP instance for (a).

As an example, the SP instance for the graph shown in Fig. 5.4(a) is given in Fig. 5.4(b). Letting $k = 3$, the subsets in the SP instance can be partitioned into 3 groups where subsets in the same group are shown in the same color: $\{S_1, S_5, s_1^4, s_1^5\}$, $\{S_2, S_3, s_1^3, s_1^7\}$ and $\{S_4, s_1^1, s_1^2, s_1^6\}$. It determines a valid 3-coloring in the VC instance, as shown in Fig. 5.4(a).

Theorem 5 *The SCBP problem is NP-hard.*

Proof. Given the proofs of *Lemma* 3 and *Lemma* 4, we can draw the conclusion that the SCBP problem is NP-hard. ■

Since no optimal SCBP decision can be made in polynomial time, in the next section, we will give an approximation algorithm for the SCBP problem under BLOS scheduling.

5.5 Spanning Tree Mesh Router Scheduling under BLOS Policy

In this section we present an approximation algorithm to ensure wireless mesh routers to cover hot spots under BLOS policy. The key idea of our algorithm is to construct a spanning tree in the WMN to ensure all spots to be covered. The spanning tree is reconstructed periodically according to the BLOS policy. A new tree is constructed to recover the mesh routers in the old tree. We refer to this algorithm as the *Spanning Tree Scheduling* (STS).

We assume that there are r mesh routers in the network. The STS algorithm consists of two steps. First, each mesh router computes its optimal policy by BLOS. Then a spanning tree is constructed from all spots to be covered. $m_1, m_2, \ldots, m_s$ are the s nodes selected in this tree. They are turned to active for δ_1 time, where $\delta_1 = \min\{\delta_1^{m_1}, \delta_1^{m_2}, \ldots, \delta_1^{m_s}\}$. All other nodes are turned into idle during this time. After δ_1 time, a new tree is reconstructed. The STS algorithm turns all nodes not on the spanning tree into idle for recovery. To avoid a router being selected again in the next round, STS gives a weight to each of them. When we select spanning tree nodes, a node with lower weight has higher priority. Table 3 gives the STS algorithm in detail. For a network with r mesh routers, it is easy to see that the time complexity to construct a spanning tree is $O(r)$. In order to verify whether a spanning tree is constructed in the STS algorithm, we let each node broadcast a short packet via the selected routers. Such overhead is very minor considering the number of mesh routers is fairly small $(30-100)$, and the radius of a router is fairly long (300 feet) in a WMN [67, 68]. Fig. 5.5 gives an example of how the STS algorithm works.

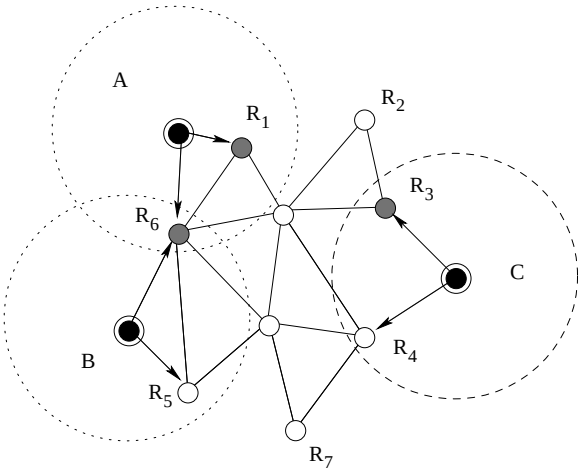


Figure 5.5: An example to show one step of constructing a spanning tree under the STS algorithm. A, B and C are the hot spots to be covered. Weight$(R_1) = 0$, Weight$(R_3) = 0$, Weight$(R_4) = 10$, Weight$(R_5) = 20$, and Weight$(R_6) = 10$. At this step, R_1, R_3 and R_6 are selected based on their weights.

5.6 Performance Evaluations

In this section we evaluate the performance of the STS algorithm under BLOS through simulations. The simulation consists of two parts: stand-alone router performance (Section 5.6.1), and mesh network performance (Section 5.6.2). We evaluate the performance with respect to router lifetime, data throughput, network lifetime and energy consumption.

5.6.1 Stand-Alone Router Performance

In this subsection we evaluate the lifetime of a stand-alone mesh router under BLOS policy. For comparison purpose we also simulated two other battery scheduling algorithms: *Greedy Scheduling* (GS) and *Fixed-Time Scheduling* (FTS), on the same router battery. The GS algorithm lets a battery discharged without recovery during its lifetime. The FTS algorithm switches the battery between active and idle modes in simple fixed time slots. Our

simulation results show that these two algorithms are much less energy-efficient than the BLOS algorithm. To further strengthen our conclusion, we also evaluated battery lifetime performance for different batteries with various initial capacity α, chemical parameter β and discharging current I. Fig. 5.6 gives the simulation results. From the figure, we can observe that by putting router batteries in recovery, both BLOS and FTS improve battery lifetime in all simulated cases compared to GS. We also observe that battery lifetime can be prolonged by up to 21% under BLOS. This is because that BLOS carefully calculates the discharging loss to find the optimal policy. Now let's consider the effects of battery capacity α, chemical parameter β and discharging current I. First, a larger battery capacity usually leads to a longer lifetime. Battery in Fig. 5.6 (b) has the same β and I but larger battery capacity than that in Fig. 5.6 (a). Therefore BLOS achieves longer lifetime in Fig. 5.6 (b) than in Fig. 5.6 (a). Second, the battery in Fig. 5.6 (c) has the same battery capacity and I but a larger β value than that in Fig. 5.6 (a). A larger β value has less discharging loss for BLOS to improve. We can see that, compared to GS, the rate of lifetime improvement by BLOS in Fig. 5.6 (c) is lower compared to that in Fig. 5.6 (a). Finally, the battery in Fig. 5.6 (d) has the same battery capacity and β but higher I than that in Fig. 5.6 (a). According to the battery model, the higher current I, the more discharging loss. Our simulations verify this. Compared to GS, BLOS has longer lifetime in Fig. 5.6 (d) than in Fig. 5.6 (a).

5.6.2 Network Performance

In this subsection we evaluate the performance of the STS algorithm under BLOS in WMNs. We consider the number of alive nodes, network lifetime, energy dissipation and data throughput in the simulation. We assume that the WMN is setup in a 150×150 field. Wireless mesh routers and hot spots are randomly distributed in the field. Fig. 5.7 shows

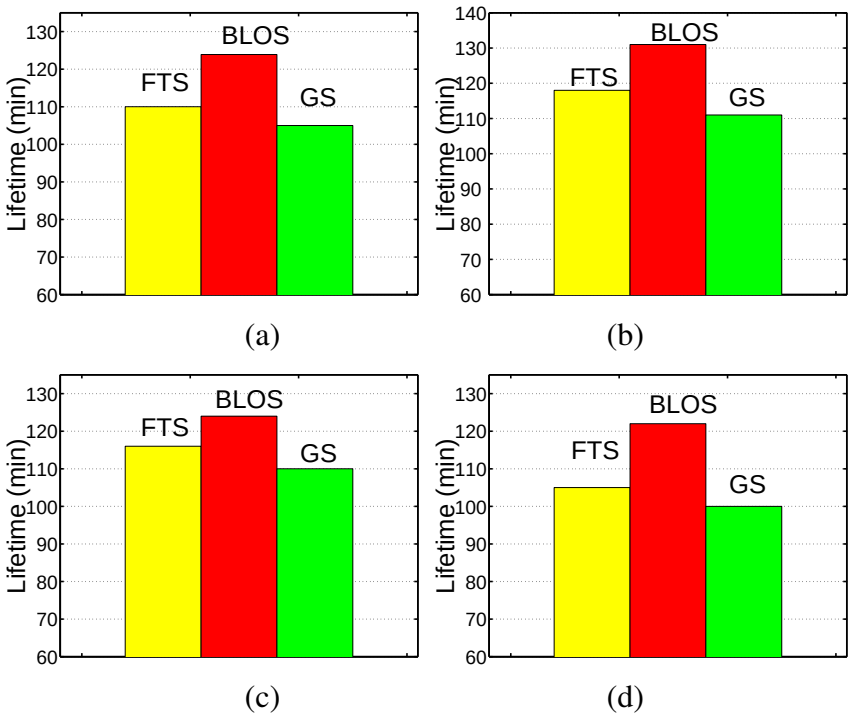


Figure 5.6: Simulated lifetime under BLOS, Greedy Scheduling and Fixed-Time Scheduling for various α, β, and I. (a) $\alpha = 4.5 \times 10^4 mA\text{min}$, $\beta = 0.4$, $I = 900mA$, $\varepsilon = 100mA\text{min}$; (b) $\alpha = 5 \times 10^4 mA\text{min}$, $\beta = 0.4$, $I = 900mA$, $\varepsilon = 100mA\text{min}$; (c) $\alpha = 4.5 \times 10^4 mA\text{min}$, $\beta = 0.5$, $I = 900mA$, $\varepsilon = 100mA\text{min}$; (d) $\alpha = 4.5 \times 10^4 mA\text{min}$, $\beta = 0.4$, $I = 1300mA$, $\varepsilon = 100mA\text{min}$.

an example network with 50 routers and 15 hot spots.

In our simulation we consider several possible WMNs with different numbers of mesh routers and hot spots. To model real-world applications, we also evaluate our algorithm for heterogeneous WMNs, that is, networks with mesh routers having various initial battery capacity α and various β. This may be the case when a WMN is implemented using mesh routers of different brands. The routers from different companies come with very different α and β values. For comparison purpose we implemented two approaches: (i) Greedy Scheduling mode (GS), where all routers are continuously discharged; (ii) Greedy

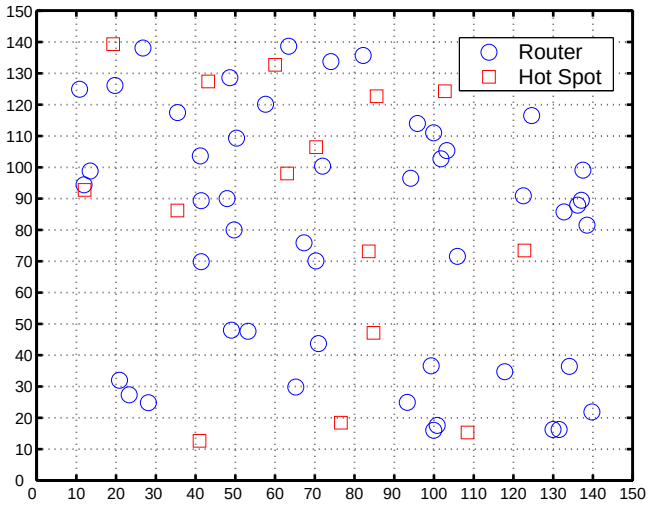


Figure 5.7: An example of a WMN with 50 mesh routers and 15 hot spots distributed.

Spanning Tree mode (GST), where a spanning tree is constructed without considering their battery status. After at least one router in the spanning tree exhausts its energy, a new spanning tree is constructed by alive routers. We evaluated the performance in terms of alive routers, energy dissipation and data throughput.

Network Lifetime. We first consider the number of alive routers during the network lifetime. Since the BLOS algorithm enables mesh routers to use up battery energy gradually, there should be more alive routers. It should be pointed out that the lifetime of a WMN is different from that of a stand-alone router. The lifetime of a stand-alone router is calculated as the total time of its power-on epochs. For example, in Fig. 5.3 the stand-alone router battery lifetime is 116 minutes. Its recovery time does not count for its lifetime. However, in a WMN, mesh routers cooperate with each other for routing: when a few routers are in recovery, the entire network is still working. In the example of Fig. 5.3, if given a sufficiently large number of routers in the network, the total network lifetime could be at least 262 minutes. In this way the total network lifetime can be greatly improved. The

decreasing of the number of alive nodes is shown in Fig. 5.8 for various numbers of routers and hot spots, and α and β values. Fig. 5.8 (a) evaluates two networks with different initial number of routers: network A consists of 50 routers and 15 hot spots, and network B consists of 100 routers and 40 hot spots. The routers in both networks have identical batteries with $\alpha = 4.5 \times 10^4 mA\text{min}$ and $\beta = 0.4$. We observe that STS increases network lifetime for both networks due to its battery-aware scheduling policy. As the number of routers in network B is larger than that in network A, the lifetime of network B is generally longer than that of A. However, although the number of routers in A is only a half of that in B, STS in network A achieves 89.2% of the lifetime of network B, which is longer than the half of lifetime of network B. This is because that as the number of routers increases, the overheads of network routing, such as transmission failures, path contentions and packet collisions, also increase. Thus the lifetime improvement is not linearly propositional to the increase of the number of routers. Fig. 5.8 (b) evaluates two networks with the same number of initial routers and APs. Batteries of routers in the two networks, however, have various α and β values. Network A has identical routers with $\alpha = 4.5 \times 10^4 mA\text{min}$ and $\beta = 0.4$. Network B has router batteries with $\alpha \in [0, 4.5 \times 10^4] mA\text{min}$ and $\beta \in [0, 0.4]$. We first observe that, since the battery-aware scheduling is sensitive to battery status, the decrease in the number of nodes under the STS is slower in both networks. We also note that the improvement rate by STS in network A is higher than that in network B, due to larger battery capacities.

Network Data Throughput. In this simulation, we study the effects of different battery capacities (α) and different battery parameters (β) on network data throughput. We first consider the effects of battery capacity α. We compare two networks, A and B, each containing 50 mesh routers. The routers in the two networks are equipped with batteries

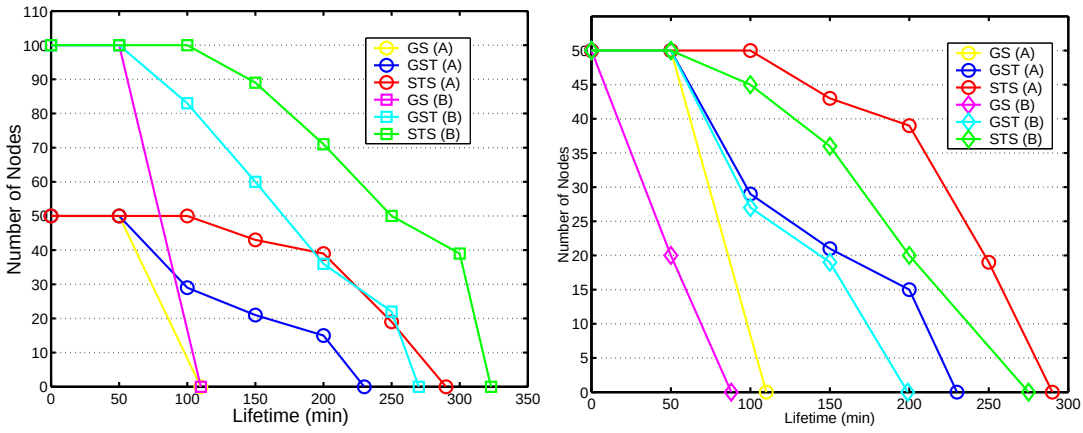


Figure 5.8: Number of alive nodes in the WMN. (a) Simulation results for various numbers of routers and hot spots. Network A has 50 routers and 15 hot spots; Network B has 100 routers and 40 hot spots. (b) Simulation results for various α and β values. Network A has identical routers with $\alpha = 4.5 \times 10^4 mA\text{min}$ and $\beta = 0.4$; Network B is heterogeneous with $\alpha \in [0, 4.5 \times 10^4] mA\text{min}$ and $\beta \in [0, 0.4]$.

with large capacity $\alpha = 4.5 \times 10^4 mA\text{min}$ and small capacity $\alpha = 3.0 \times 10^4 mA\text{min}$, respectively. They have the same β value of 0.5. We evaluate the number of packets successfully routed through routers to APs. The results are plotted in Fig. 5.9 (a). We can see that by adopting battery-awareness, both *STS* and *GST* improve network data throughput with up to 84.3% improvements compared to *GS*. We also observe that network A has higher data throughput than network B under all three scheduling policies. This is because that the larger router battery capacities of network A prolong network lifetime, which in turn improves data throughput.

We then compare two networks, C and D, each containing 50 routers equipped with batteries with large parameter $\beta = 0.6$ and small parameter $\beta = 0.4$, respectively. They have the same α value of $4.5 \times 10^4 mA\text{min}$. The results are shown in Fig. 5.9 (b). It is interesting to observe that *GS* in both networks achieve almost equal lifetimes. This is

because that *GS* does not consider battery-awareness, thus difference β values do not affect battery lifetime under *GS*. *STS* and *GST*, however, consider battery discharging loss to improve data throughput. We can see that the smaller the β, the larger the discharging loss. Thus, batteries with smaller β values give *STS* and *GST* more room to improve their lifetimes. The results of Fig. 5.9 (b) show that, compared to *GS*, data throughput improvement by *STS* is 90.3% in network D and 83.9% in network C.

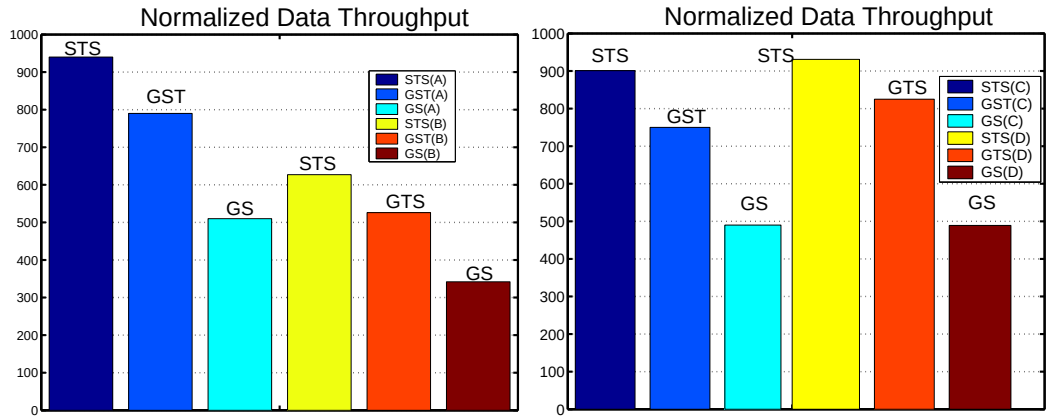


Figure 5.9: Effects of battery capacities (α) and different battery parameters (β) on network data throughput. Networks A, B, C and D each contains 50 mesh routers and 15 APs. (a) The routers in two networks, A and B, are equipped with batteries with large capacity $\alpha = 4.5 \times 10^4 mA\text{min}$ and small capacity $\alpha = 3.0 \times 10^4 mA\text{min}$, respectively. (b) Routers in two networks, C and D, are equipped with batteries with large parameter $\beta = 0.6$ and small parameter $\beta = 0.4$, respectively.

Energy Dissipation. We evaluated the network with various number of heterogeneous routers. Fig. 5.10 shows the energy distribution of the routers in the middle of network transmission (at the $60_{th} min$). The *X* and *Y* axes show the geographic positions of routers in the network. The *Z* axis stands for the residual battery energy of routers. It can be seen that by adopting STS, routers can preserve higher battery energy.

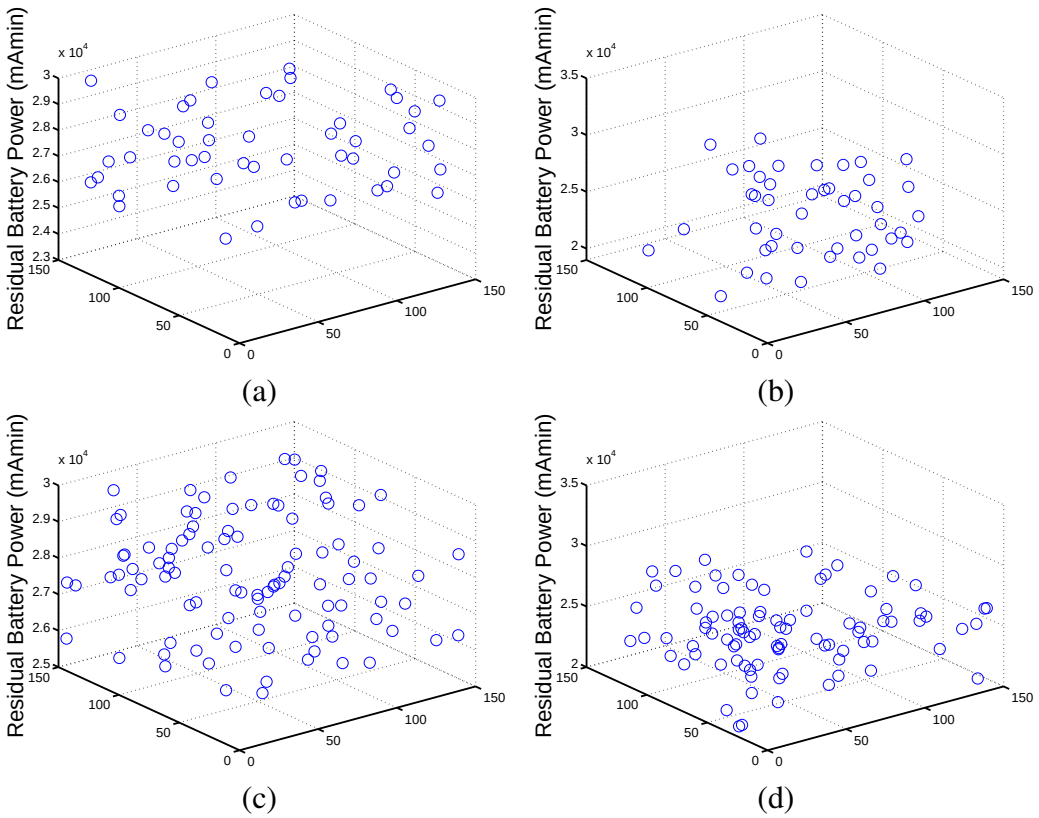


Figure 5.10: Energy dissipation at the 60th minute. (a) 50 routers, STS algorithm; (b) 50 routers, GST algorithm; (c) 100 routers, STS algorithm; (d) 100 routers, GST algorithm; Mesh routers are heterogeneous with random α and β values.

5.7 Summaries

In this chapter, we have considered energy-efficient scheduling in WMNs for hot spot covering. We have studied the relationships between discharging duration and battery lifetime, and introduced a battery lifetime optimization scheduling algorithm (BLOS) to maximize the lifetime of battery-powered mesh router. Based on the BLOS algorithm, we have further considered the SCBP problem for monitoring or covering hot spots under BLOS algorithm. We have proved that the SCBP problem is NP-hard, and given an approximation

algorithm (STS) with time complexity of $O(r)$ for a network with r mesh routers. Our simulation results show that the STS can greatly improve lifetime, data throughput and energy consumption efficiency of WMNs.

Table 3: Spanning tree scheduling algorithm (STS)

Procedure *STS*
begin
Initially each mesh router is assigned a weight;
$\tau = 0, h = 1$;
Each mesh router computes its policy by BLOS;
repeat
Each hot spot is colored as a *black* node;
Each mesh router is colored as a *white* node;
while *the spanning tree is not connected* **do**
begin
Each *black* node k selects a *white* node i,
where Distance(i,k) <Radius(i),
and weight(i)=
min{weight(j)|distance(j,k) <Radius(j)};
i is colored *black*;
end
Obtain a spanning tree with s routers: $m_1, m_2, \ldots, m_s$;
Mesh routers not in spanning tree are
turned idle for recovery;
This spanning tree works for
$\max\{\tau, \min\{\delta_h^{m_1}, \delta_h^{m_2}, \ldots, \delta_h^{m_s}\}\}$ time;
$\tau = \max\{\tau_h^{m_1}, \tau_h^{m_2}, \ldots, \tau_h^{m_s}\}$;
Each node $m_l (l = 1 \ldots s)$ does:
Weight(m_l)=Weight(m_l)+δ_{weight};
$h = h + 1$;
until no spanning tree can be constructed;
end

Chapter 6

Battery-Aware Router Scheduling for MIMO WMN

This chapter presents battery-aware energy-efficient schemes for multiple input multiple output mesh networks. Mesh routers are equipped with multiple radio transceivers that can work simultaneously. This MIMO feature greatly improves data throughput of mesh routers. In this chapter we first study the relationships between various MIMO transceiver parameters and their battery parameters to give an energy model for MIMO transceivers. We then present a multiple current battery model that can accurately describe battery behaviors with multiple current inputs. Based on these two models, we propose a battery-aware MIMO WMN energy scheduling scheme. The scheme consists of two algorithms: the coverage algorithm and the backhaul routing algorithm. The key idea of the coverage algorithm is to let neighboring mesh routers collaboratively adjust their transceiver radii to dynamically recover their over-discharged battery energy. The backhaul routing algorithm adopts the multiple current battery model to calculate battery discharging loss at routers for scheduling mesh backhaul routing. We conducted simulations to evaluate the performance

of the proposed scheme. The results show that network lifetime can be improved by up to 10.3% and 16.1% for homogeneous and heterogeneous WMNs, respectively.

The rest of the chapter is organized as follows. In Section 6.1 we discuss the background and related work to place our work in context. Section 6.2 shows the battery-awareness in radius scheduling on mesh router. In Section 6.3 we study the energy model of mesh radio transceivers. We then present the multiple current battery model in detail in Section 6.4. In Section 6.5 we give the battery-aware MIMO WMN energy scheduling scheme to maximize the battery lifetime. Finally, we present the simulation results in Section 6.6, and give concluding remarks in Section 6.7.

6.1 Related Work

Mesh routers equipped with multiple radio transceivers greatly improve data throughput of WMNs[67]. A MIMO mesh router has at least two radios: coverage radio and backhaul radio. A coverage radio is a transceiver that transmits packets between a mesh router and mesh clients within its radio coverage. A coverage radio is typically an omnidirectional transceiver [65], also known as an isotropic transceiver, that radiates and receives signals equally in all directions. A backhaul radio is the transceiver that connects two mesh routers or a mesh router and an AP. Often a mesh router has more than one backhaul radios. These backhaul radios, together with the coverage radio, work simultaneously at a mesh router with multiple packets going through different radio transceivers. A backhaul radio adopts a directional transceiver [63, 64] to support the MIMO function. A directional transceiver has two main advantages over an omnidirectional transceiver. First, a directional transceiver makes the MIMO function of a mesh router possible. A mesh router can transmit different packets to different neighbors at the same time. It causes less interference to mesh routers

that are not in the direction of the transmission. Second, for a given transmission energy, a directional transceiver can transmit packets over a longer distance in a particular direction compared with an omnidirectional transceiver in an isotropic disk area. This is because that a directional transceiver uses most of its energy in the direction of the transmission, while an omnidirectional transceiver scatters signals in all directions.

In order to schedule the energy consumption of mesh router transceivers in a battery-aware manner, we need an energy model that describes the relationship between the battery parameters and radio transceiver parameters. To serve for this purpose, two models are required. The first model is the energy model for energy dissipated by transceivers. Several previous work, such as [63, 64], gave transceiver energy models that focus on signal gain of a transceiver. In Section 6.3 we will present a transceiver model that describes the relationship among battery current, signal gain and transceiver radius for both directional and omnidirectional transceivers. The second model is the mathematical battery model for MIMO mesh routers. Battery models that can capture the battery discharging behavior have been developed in previous chapters. They achieve good performance, they are not for MIMO mesh routers with multiple currents flowing through the router battery. In Section 6.4 we will present a multiple current battery model that can calculate the battery discharging loss based on multiple current inputs. Battery-aware schemes that adopt battery models to schedule network activities were given in chapter 2 and chapter 5. Chapter 2 proposes a battery-aware routing protocol to locally choose routing hops for MANETs. Chapter 5 designs a spanning tree scheduling scheme to ensure battery-aware hot spot covering for WMNs. However, these protocols were designed based on a single transceiver at each router, thus they are not suitable for MIMO WMNs with multiple transceivers at each router. In Section 6.5 a battery-aware MIMO mesh network power scheduling (BAMPS)

scheme will be proposed to maximize the lifetime of MIMO WMNs.

6.2 Battery-Aware Transceiver Radius Scheduling

In this section we study the issue to achieve battery-awareness by dynamically scheduling the radii of mesh radio transceivers. We let neighboring mesh routers collaboratively adjust their transceiver radii and alternatively recover their battery. Fig. 6.1 gives an example in one-dimension, where a network has two routers A and B and the distance between them is $L = 10m$. Each router has a battery with $1.8 \times 10^4 m\text{A}min$ capacity. A and B collaboratively cover mobile clients on the line between them. We let two routers alternatively use radii R and r, ($L = R + r$ and $R \neq r$), each for a period. As shown in Fig. 6.1(b), in period 1, B uses a shorter radius and has a battery current lower than A, thus it recovers its battery during period 1. In the next period, A adopts a shorter radius to recover its battery. In this way A and B together can minimize the total battery discharging loss and maximize their lifetime. Specifically, we depict the battery discharging loss for router A's radius R in Fig. 6.2(a).

We let A and B alternatively choose R and $L - R$ as their radius for a period of 20 minutes. In this case, the maximum energy consumption occurs when A and B adopt an equal radius 5m. This is because that none of the two routers have a chance to recover their battery discharging loss in this schedule. Fig. 6.2(b) shows the total energy consumption of A and B. We can see that in the example, the network achieves a minimum energy dissipation by letting A and B alternatively adopt radii of $3m$ and $7m$, respectively. Thus, the battery-aware approach in Fig. 6.1(b) can prolong the lifetime by 11.1% than the approach in Fig. 6.1(a). This example shows that we can effectively enhance network performance by taking advantage of the battery behavior. In Section 6.5 we will study the scheduling

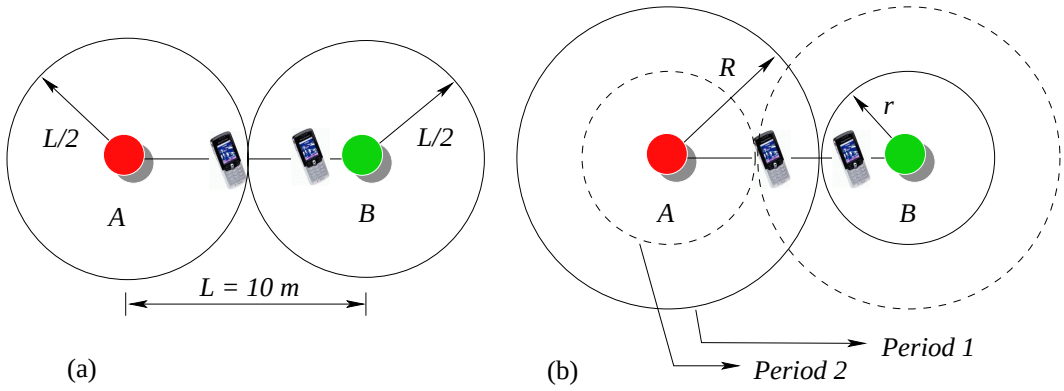


Figure 6.1: Routers collaboratively recover their batteries by dynamically adjust the radii. (a) Two mesh routers use the same radius 5m to cover mesh clients. (b) They alternatively use radii 7m and 3m.

scheme in detail and extend it to two-dimension.

6.3 Energy Models of Mesh Router Radio Transceivers

In this section we study the transceiver energy model for both directional and omnidirectional transceivers. The model describes the relationship among battery current, signal gain and transceiver radius.

Fig. 6.3 shows a radiation pattern of a directional radio transceiver. The area of its radiation is referred to as lobes which reach out from the center. The main lobe is the direction of the maximum radiation or reception. In addition to the main lobe, there are also side lobes and back lobes. These lobes represent the energy loss that a good transceiver attempts to minimize. To measure the radiation of a main lobe, we use $\vec{d} = (\theta, \phi)$ to describe its direction. ϕ ($0 \leq \phi < 2\pi$) is the angle orientation that is defined as the angle measured counter-clockwise from the horizontal axis (0°) to the antenna boresight. θ ($0 \leq \theta < 2\pi$) is the lobe beamwidth. When θ is set to 360°, the radio is an omnidirectional radio.

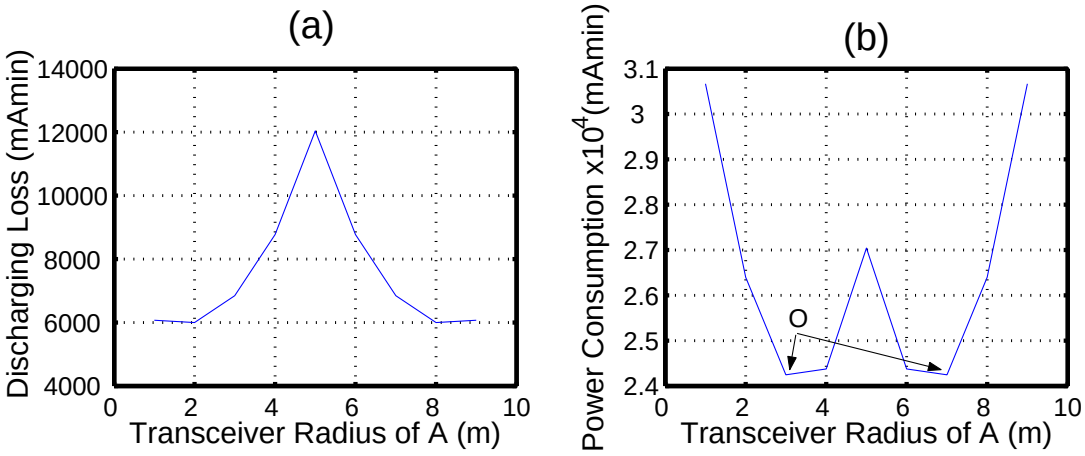


Figure 6.2: Energy consumption vs. transceiver radii of router A in Fig. 6.1. (a) Battery discharging loss of transceiver model. (b) By using battery recovery, minimum energy consumption can be achieved as symbol O indicates.

When using a directional radio for transmitting or receiving, according to the Friss Equation [66] the transmit and receive powers P_T and P_R are related to the transmit and receive gains G_T and G_R as follows.

$$P_R = \frac{P_T G_T G_R}{K r^{\lambda}} \tag{6.1}$$

where term K is a constant that accounts for transmit media such as atmospheric absorption, r is the distance between the transmitter and the receiver, and λ is the path-loss index ($2 \leq \lambda \leq 4$). We define the power density as the transmitter power per unit area [63]. Hence the transmit gain of a directional antenna with a particular direction $\vec{d} = (\theta, \phi)$ is given as

$$G_T = \eta \frac{U(\vec{d})}{U_{avg}} \tag{6.2}$$

where $U(\vec{d})$ is the power density in direction $\vec{d}$, U_{avg} is the average power density over all

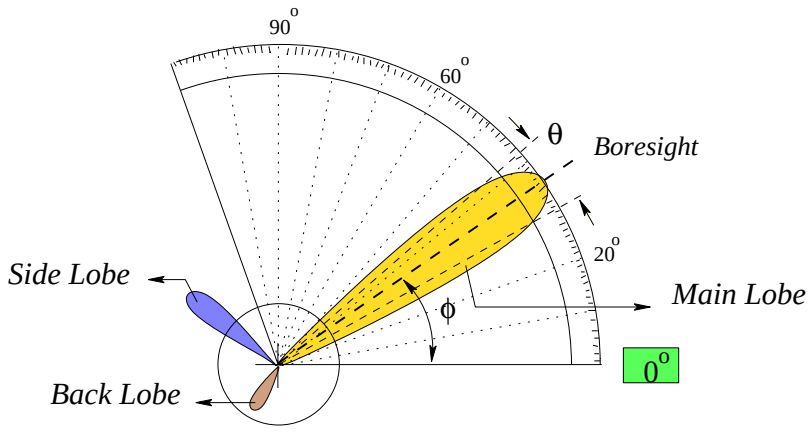


Figure 6.3: Directional radio patten of a directional MIMO mesh router.

directions, and constant η is the efficiency of the antenna and accounts for energy losses. The larger the beamwidth θ, the smaller the value of $U(\vec{d})/U_{avg}$.

From (6.1) and (6.2) we obtain

$$P_R = \frac{\eta U(\vec{d}) P_T G_R}{K r^{\lambda} U_{avg}} \tag{6.3}$$

The electric current I of a transceiver and its power P_T satisfy $P_T = I^2 \times S$, where constant S is the resistance. Hence (6.3) can be written as

$$P_R = \frac{\eta U(\vec{d}) S G_R}{K r^{\lambda} U_{avg}} I^2 \tag{6.4}$$

Observing (6.4) we obtain

$$r = \left(\frac{K P_R}{\eta S G_R} \times \frac{U(\vec{d})}{U_{avg}} I^2 \right)^{\frac{1}{\lambda}} \tag{6.5}$$

Note that the receive gain G_R and power P_R are fixed. K, S, η and λ are constants based on the device, atmospheric condition and communication media. It can be seen that given

a specific beamwidth θ, $U(\vec{d})/U_{avg}$ is fixed, hence the electric current I of a transceiver increases in the order of $I^{\frac{2}{\lambda}}$ as transmit radius r increases. This energy model also indicates that given a fixed r, the wider the beamwidth, the larger the I its battery obtains. For omnidirectional radio transceivers, $U(\vec{d})/U_{avg} = 1$. Their energy model can be written as

$$r = \left(\frac{KP_R}{\eta S G_R} \times I^2 \right)^{\frac{1}{\lambda}} \tag{6.6}$$

Thus, we have obtained the energy model for both directional and omnidirectional transceivers. In the next section we will study the battery model for multiple current inputs.

6.4 Battery Model for Multiple Transceiver Mesh Router

Battery models have been designed in Chapter 2, Chapter 3 and Chapter 5 for MANET, WSN and single transceiver WMN, respectively. The MIMO mesh routers need a battery model that can work efficiently for multiple current inputs. In this section we present the multiple current battery model (or MCBM for short) for mesh routers equipped with multiple radio transceivers. We assume that a router has q transceivers each of which contributes a current I^j $(j = 1, 2, \ldots, q)$ to the circuit. Each I^j periodically turns down to recover the battery. As depicted in Fig. 6.4(b), the battery lifetime is divided into durations. Each duration contains a heavy load period and a recovery period. A heavy load period δ_i is followed by a recovery period of length τ_i. Without loss of the generality, we assume that a current I^j rises to I_i^j for δ_i time, and then turns to $\bar{I}_i^j$ for τ_i time ($i = 1, 2, \ldots$, and $I_i^j > \bar{I}_i^j$). Note that by turning a current I^j to $\bar{I}^j$, we let battery recover its over-charged discharging loss. Fig. 6.4 (a) illustrates the recovered capacity of discharging loss of period δ_1. The active-recovery period is repeated until the battery uses up its energy.

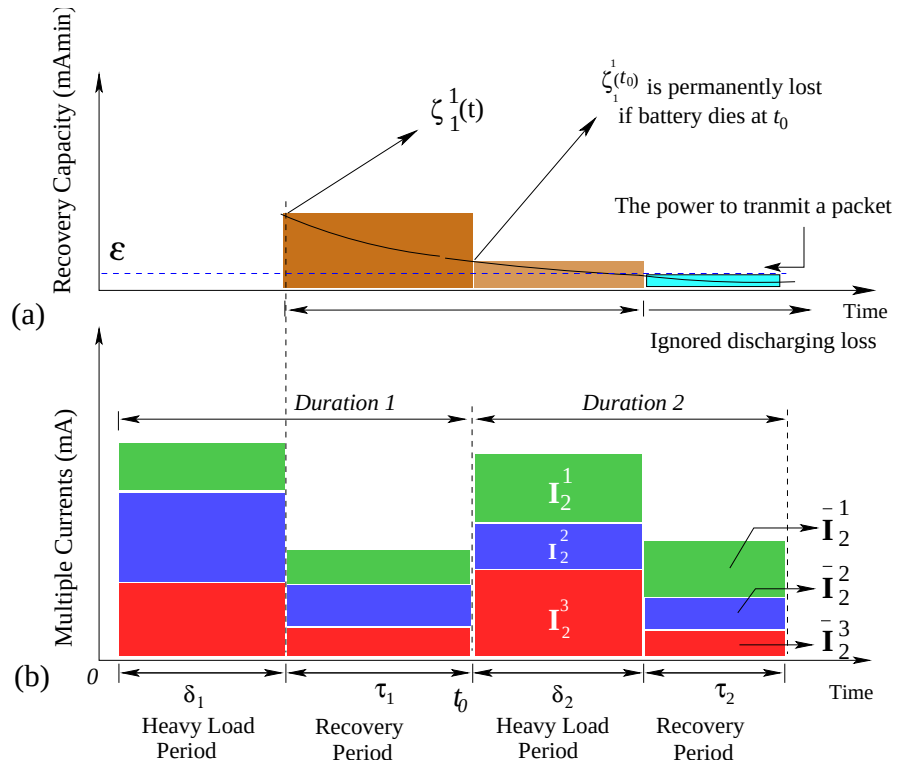


Figure 6.4: The battery discharging and recovering scenario. (a) The capacity $\zeta_1^1(t)$ is discharged in heavy load period δ_1 and recovered gradually after that. $\zeta_1^1(t)$ after time $\delta_1+\tau_1$ is ignored. (b) Discharging of a battery in δ_1 and δ_2 heavy load periods. Battery is recovered by turning down currents. If this battery dies at t_0, $\zeta_1^1(t_0)$ is permanently lost.

In our battery model, the battery is discharged in each discharging period i with length δ_i, where δ_i may not be equal to δ_j if $i \neq j$. Each current I^j in the i_{th} heavy load period consumes certain amount of energy. We use $|\alpha_i^j|$ and ζ_i^j to denote the battery capacity consumed by current I^j during the i_{th} heavy load period and the discharging loss of current I^j in the i_{th} heavy load period, respectively. We use T to denote the entire lifetime of the battery.

The condition of a battery at the i_{th} heavy load period is measured by its discharging loss at that time. A high discharging loss indicates a "fatigue" battery which needs some recovery, while a battery with low discharging loss is well recovered. Intuitively, an energy-efficient scheduling scheme should always choose routers with well recovered batteries to work with. Therefore, a good battery model should be able to calculate the discharging loss for a given time.

The following analytical model can be used to compute the battery discharging loss. Given $\bar{t}$ as the beginning time of the i_{th} heavy load period, the energy dissipated by the battery during the i_{th} heavy load period $[\bar{t}, \bar{t}+\delta_i]$ is

$$|\alpha_i^j| = I_i^j \times F(T, \bar{t}, \bar{t}+\delta_i, \beta) \tag{6.7}$$

where

$$F(T, \bar{t}, \bar{t}+\delta_i, \beta) = \delta_i + 2\sum_{m=1}^{\infty}\left[\frac{e^{-\beta^2 m^2 \bar{t}} - e^{-\beta^2 m^2 (\bar{t}+\delta_i)}}{\beta^2 m^2}\right]$$

This model can be interpreted as follows. The dissipated energy during the i_{th} heavy load period is $|\alpha_i^j|$ in (6.7). It contains two components: The first term, $I_i^j \times \delta_i$, is simply the energy consumed in the device during $[\bar{t}, \bar{t}+\delta_i]$. The second term, $2I_i^j \times \sum_{m=1}^{\infty}\left[\frac{e^{-\beta^2 m^2 \bar{t}} - e^{-\beta^2 m^2 (\bar{t}+\delta_i)}}{\beta^2 m^2}\right]$ is the amount of battery discharging loss in this heavy load period. It can be seen that the

discharging loss decreases as the lifetime T increases. β (> 0) is a constant, which is an experimental chemical parameter and may vary from battery to battery. The larger the β, the faster the battery diffusion rate, hence the less the discharging loss. Next we show how the model in (6.7) can be used to calculate the discharging loss at a given heavy load period.

As defined earlier, ζ_i^j is consumed in the i_{th} heavy load period and recovered in the next τ_i time. Clearly,

$$\zeta_i^j(t) = 2I_i^j \times \sum_{m=1}^{\infty} \left[\frac{e^{-\beta^2 m^2(\bar{t}+t)} - e^{-\beta^2 m^2(\bar{t}+\delta_i+t)}}{\beta^2 m^2} \right] \tag{6.8}$$

where $\zeta_i^j(t)$ is the residual discharging loss at time t.

As can be observed, the recovery of $\zeta_i^j(t)$ continues from $t+\delta_i$ to ∞. In practice, we can simplify the computation of ζ_i^j as follows. Assume ε is a fairly small constant, which is the energy to transmit a packet. By observing (6.8), if $\zeta_i^j(\tau_i)$ is less than ε, we can ignore the discharging loss after time $\bar{t}+\delta_i+\tau_i$. Thus, after τ_i time of recovery, the battery can be considered to be well-recovered.

Given battery parameter β, heavy load time δ_i and $\bar{t}$, we can pre-calculate ζ for various possible current values and put them into a look up table. In the multiple current scenario, when we need to measure how much discharging loss a mesh router has, we can look up ζ_i^j for all currents and sum them up to obtain the total discharging loss. Thus, based on the look up table, the MCBM can provide an efficient approach for calculating battery discharging loss for multiple current inputs. Next we will apply the model to battery lifetime scheduling in wireless MIMO WMNs.

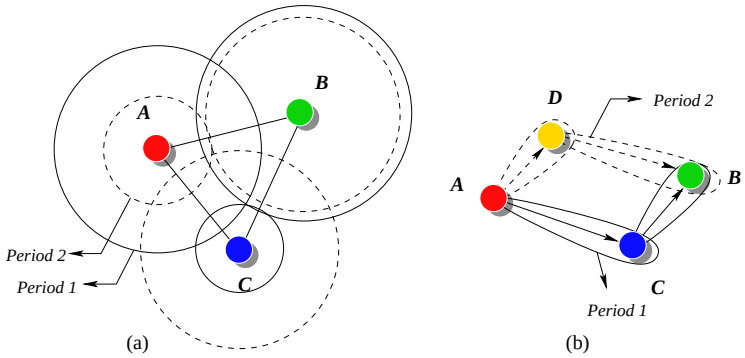


Figure 6.5: BAMPS scheme for wireless mesh routers. (a) The coverage algorithm. (b) The backhaul routing algorithm.

6.5 Battery-Aware MIMO WMN Scheduling

In Section 6.2, we demonstrated that by periodically changing transceiver radii according to battery parameters we can effectively improve network performance in terms of network lifetime and energy dissipation. In this section we study the mesh router scheduling and present a battery-aware MIMO mesh network power scheduling (called BAMPS) scheme. The BAMPS scheme consists of two parts: the coverage algorithm and the backhaul routing algorithm for mesh client coverage and backhaul routing, respectively.

We first consider the coverage algorithm. Section 6.2 gave a scheduling approach for router coverage in one-dimension. We now extend this approach to two-dimension. Assume that there are l mesh routers in the network and each router has at most n neighbors. The idea of the coverage algorithm is that each mesh router periodically calculates its battery discharging loss according to the MCBM model, and broadcasts it to its one hop neighbors. After obtaining all neighbors' discharging loss values, a router calculates a feasible radius according to the transceiver model and the router's battery status. To ensure that the

entire network area is covered, each router chooses the distance between itself and its farthest neighbor for the radius calculation. Fig. 6.5 shows an example network consisting of routers A, B and C. In period 1, C adopts a short radius which leads to the battery recovery for C. In the next period, A has the opportunity to recover its battery, and so on. Table 1 gives an outline of the coverage algorithm. This is a distributed algorithm, and each mesh router only needs to communicate with at most n neighbors. Thus, the time complexity of the algorithm is $O(n)$.

We now consider the backhaul routing in which mesh backhaul nodes communicate through directional radios. Whenever there is a need to transmit data packets to or from APs and neighbor routers, a directional connection is set up from the sender to the receiver. The idea of the backhaul routing algorithm is to relay data packets through different neighbors. Thus it can recover batteries on those routers not currently in use. Since the data traffic transmitted among mesh backhaul nodes is generally quite heavy, relaying data packets through a specific router would soon use up its battery without letting it recover the overcharged energy. Our backhaul routing algorithm lets the sender periodically poll the battery status of its relay routers and assign the relay route on the best recovered routers. As depicted in Fig. 6.5(b), sender A transmits data packets to B through C or D. A polls the network for battery discharging loss. At the beginning, C has lower discharging loss than D. Therefore in period 1, A chooses C as the relay node and leaves D for recovery, and vise verse in period 2. Table 1 also gives an outline of the backhaul routing algorithm. The algorithm is a centralized algorithm. In order to set up a route, a sender polls the network once for battery statuses. Thus, it has time complexity $O(l)$. Note that since a typical WMN has only about 30 to 100 backhaul nodes [67], the $O(l)$ time complexity is quite reasonable.

Table 1: Battery-aware MIMO mesh network power scheduling (BAMPS) algorithm

Coverage Algorithm
  repeat
  A mesh router i does the following:
    Calculates its discharging loss based on the MCBM model;
    Broadcasts discharging loss to one hop neighbors;
    Collects n packets from one hop neighbors;
    Calculates its radius $\overline{d}$ between its farthest neighbor F,
      using F's discharging loss;
    Adopts $\overline{d}$ as the transceiver radius for δ time;
    Adopts $d - \overline{d}$ as the transceiver radius for δ time,
      where d is the distance between i and F;
  until Energy is used up.
Backhaul Routing Algorithm
  A sender i does the following:
  for each δ time **do**
    Broadcasts a polling message to collect discharging loss ζ;
    Generates a threshold th
    repeat
      Chooses the shortest route among nodes with $\zeta > th$;
      if cannot find the route **then** increases th;
    until A route is found;
    Transmits data along this route;

6.6 Performance Evaluations

We conducted simulations to evaluate the performance of the BAMPS scheme. We assume that the WMN is set up in a 100×100 field. Wireless mesh routers and access points (AP) are randomly distributed in the field. APs, as portals of packets routing, are connected to the Internet by cables. Mesh routers collaboratively cover mobile clients in the 100×100 area. To simulate the scenario that mesh clients randomly move and transmit packets to mesh routers, in our simulation we let mesh routers dynamically receive packets from clients. These packets are routed to APs through other routers. The network lifetime is defined as the duration between the network is set up and the moment when its mesh

router layer can no longer cover the entire area. In our simulation we let BAMPS adjust transceiver radii every $\delta = 30$min. Two backhaul transceivers and one coverage transceiver are implemented at each router.

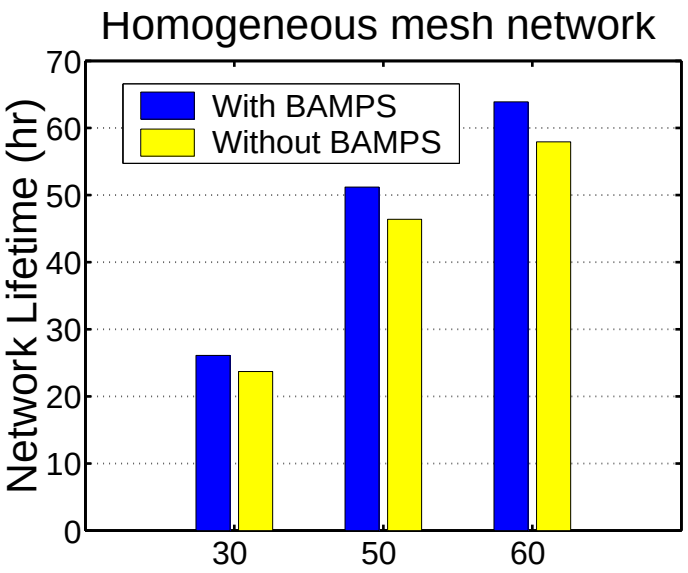


Figure 6.6: Lifetime improvement by BAMPS in homogeneous WMNs.

We consider two types of networks: homogeneous WMN and heterogeneous network. A homogeneous WMN is a network in which mesh routers are equipped with the same batteries. In our case, each battery has $1.8 \times 10^4 m$Ahr capacity with $\beta = 0.5$. This is the popular mesh router currently available on the market [70]. We measured the network lifetime for various sizes of networks: 30, 50 and 60 mesh routers, and the results are plotted in Fig. 6.6. From the figure, we can observe that the network lifetime is improved by up to 10.3% when we implement the BAMPS scheme in homogeneous WMNs. This improvement is due to the battery capacity recovered from two sources: (i) The coverage radio is alternatively adjusting radii to recover batteries; (ii) Packets are routed through well-recovered mesh backhaul nodes so that fatigue nodes can switch themselves into recovery

status.

In some applications, a WMN may consist of mesh routers from different manufacturers. These different mesh routers with various battery capacities form a heterogeneous WMN. A robust scheduling scheme, therefore, should also work in heterogeneous networks. Routers in a heterogeneous WMN usually are less energy-efficient, because some of them may be manufactured a while ago. Therefore we set mesh routers in our simulation with battery capacities ranging from $1.0 \times 10^4 m$Ahr to $2.5 \times 10^4 m$Ahr and β values from 0.5 to 0.1. The simulation results are shown in Fig. 6.7. From the figure, we can observe that the BAMPS scheme prolongs the lifetime of heterogeneous WMNs by up to 16.1%. This is because that a heterogeneous network consists of more routers with less battery capacity that are prone to die soon. BAMPS prolongs lifetimes of these routers so that they can remain in the network to maintain the network connectivity. Thus the lifetime improvement of heterogeneous networks is larger. We also observe that the entire lifetime of homogeneous networks is generally longer than that of heterogeneous networks. This is due to the fact that homogeneous networks have mesh routers with higher battery efficiency initially.

6.7 Summaries

In this chapter we have proposed a battery-aware scheme to schedule the MIMO mesh routers. We first studied the unique features of the energy model for different MIMO mesh transceivers. We then gave a multiple current battery model to describe the battery behavior for multiple transceiver inputs. Based on the energy model and the battery model, we presented an approach to dynamically scheduling radii of mesh radio transceivers in which neighboring routers can collaboratively adjust their transceivers radii to alternatively

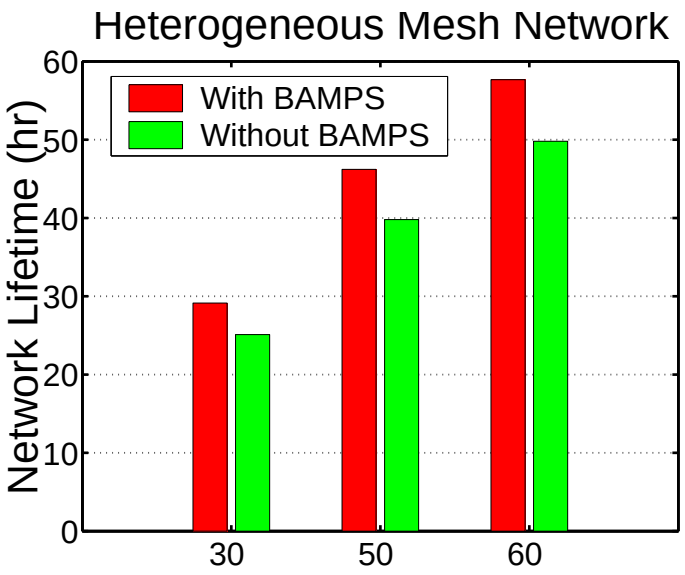


Figure 6.7: Lifetime improvement by BAMPS in heterogeneous WMNs.

recover batteries. We then gave the BAMPS scheme for network-wide mesh client coverage and backhaul routing. We have conducted simulations to evaluate the performance of the BAMPS scheme. The results demonstrate that the BAMPS achieves up to 10.3% and 16.1% longer network lifetime for homogeneous and heterogeneous WMNs, respectively.

Chapter 7

Battery-Aware Client Driven Scheduling for WMN

This chapter presents a cross-layer client driven battery-aware (CDBA) scheduling scheme to efficiently schedule mesh network coverage. CDBA scheme consists of two parts: CDBA coverage algorithm and CDBA MAC algorithm. The key idea of CDBA coverage algorithm is to let neighboring mesh routers collaboratively adjust their transceiver radii based on positions of mesh clients. In this way routers are able to recover their over-discharged battery power dynamically. This algorithm is a distributed algorithm with $O(n)$ time complexity where n is the maximum number of neighbors of a router in the network. To further jointly improve system performance among layers, we design the CDBA MAC algorithm to provide a seamless and fast service for handoff among mesh routers. We conduct simulations to evaluate the performance of the proposed CDBA scheme. The results show that network lifetime and data throughput can be improved by up to 27.27% and 30.54% in WMNs, respectively.

The rest of the chapter is organized as follows. In Section 7.1 we discuss related work

to place our work in context. Then we show that network performance can be greatly improved by adopting client driven battery-aware scheduling in Section 7.2. We present the client driven battery-aware (CDBA) scheduling scheme in Section 7.3. Finally, we show simulation results in Section 7.4, and give concluding remarks in Section 7.5.

7.1 Related Work

It is vital to maintain the WMN coverage for a long lifetime with high energy efficiency. Chapter 5 and Chapter 6 proposed mesh router driven, network layer scheduling schemes. A spanning tree scheduling scheme to ensure battery-aware hot spot covering in WMNs was designed in Chapter 5. A backhaul routing and router covering approach with directional antenna modulation and MIMO router scheduling was introduced in Chapter 6. However, these protocols did not take client positions into the scheduling decision, thus are not client driven.

In this chapter we will consider client driven, cross-layer battery-aware scheduling scheme, for on the following reasons: First, in a WMN, packets are transmitted from mesh clients to mesh routers through physical layer, MAC layer and routing layer. To jointly minimize the cost among different layers, a cross-layer scheduling protocol is desirable for power efficient scheduling in WMNs. Second, unlike a general wireless ad hoc network, in which network nodes are mobile, a WMN has routers geographically fixed and static [67]. This feature allows a scheduling protocol to dramatically reduce power consumption with only minimum overhead. Finally, the purpose of setting up a mesh router is to provide high performance service to mesh clients. A power scheduling protocol should be aware of the activities and mobilities of the clients. Therefore, an efficient scheduling scheme should be client driven.

As previous chapters revealed, batteries tend to discharge more power than needed, and reimburse the over-discharged power later if they have sufficiently long recovery time. Based on this observation, in this chapter we study the relationship between mesh routers and mesh clients by analyzing the client driven coverage scheduling with battery-awareness. We show that the network performance can be greatly improved by adopting client driven battery-aware scheduling. We adopt battery model designed in Chapter 6 for mesh router battery capacity computation. We design a client driven battery-aware (CDBA) scheduling scheme in this chapter. CDBA consists of two parts: CDBA coverage algorithm and CDBA MAC algorithm. CDBA coverage algorithm is a distributed algorithm that organizes mesh routers to collaboratively adjust their CDBA MAC algorithm provides MAC service for seamless and fast handoff among neighboring mesh routers. A handoff [67] refers to the moving of a mesh client from one mesh router to another in the middle of the client's data transmission. The main objective of the CDBA MAC algorithm is to reduce packet loss or traffic burst during handoff. We conducted simulations to evaluate the performance of the proposed scheme. The results demonstrate that CDBA scheme can improve lifetime and data throughput of WMNs by up to 27.27% and 30.54%, respectively.

7.2 Battery Recovery and Mesh Router Scheduling

In this section we will introduce a scheme for client driven battery-aware router scheduling. The key idea of battery-awareness in WMNs is to dynamically schedule the radii of mesh routers. Neighboring routers can collaboratively adjust their radii to alternatively recover their battery. We give an example in one-dimension in Fig. 7.1, where a network has two routers A and B and the distance between them is $L = 10m$. Each router has a battery with $1.8 \times 10^4 m\text{A}min$ capacity. A and B collaboratively cover three clients C_1, C_2 and C_3

positioned between them. The distances from C_1, C_2 and C_3 to router A are 3m, 4m and 6m, respectively. The radii of A and B are R and r ($R, r \leq L$), respectively. In the traditional router scheduling, radii of A and B are both set to $L/2$ equally, as shown in Fig. 7.1 (a). To take advantage of battery behaviors, a more power efficient solution is depicted in Fig. 7.1 (b): two routers alternatively use radii 7m and 3m, each for a period. In period 1, B uses a shorter radius and has a battery current lower than A, thus it recovers its battery during period 1. In the next period, A adopts a shorter radius to recover its battery. In this way A and B together can reduce the total battery discharging loss to extend their lifetime. However, as we observe from this example, when B adopts radius $r_2 = 3m$ in period 2, there is actually no client for it to cover. In other words, B can save more energy by turning off its radio in period 2. Similarly, the radius of A in period 2 and the radius of B in period 1 can be reduced to $6m$, to achieve a better power efficiency. The scheduling is illustrated in Fig. 7.1 (c), where mesh client positions are considered in router coverage scheduling.

In our example, A and B can choose from four pairs of radii to cover clients: $(R = 0, r = 7)$, $(R = 3, r = 6)$, $(R = 4, r = 4)$, and $(R = 6, r = 0)$. We simulated the network coverage by letting A and B adopt a pair of radii in a period of 20 minutes and adopt another pair in the next 20 minutes. We calculated the total power consumption of A and B as shown in Fig. 7.2. X axis and Y axis each stands for the pair of radii that A and B adopt in period 1 and in period 2, respectively. Z indicates the power consumption. Battery discharging loss is also considered in calculating power consumption. To ensure fairness among routers, we assume that their radii are different in adjacent periods. From Fig. 7.2, we can see that the network achieves a minimum power dissipation by letting A and B alternatively adopt radii of $0m$ and $7m$ in period 1, and $6m$ and $0m$ in period 2, respectively. In this way, the power consumption is reduced by 37.02% compared with the most power consuming

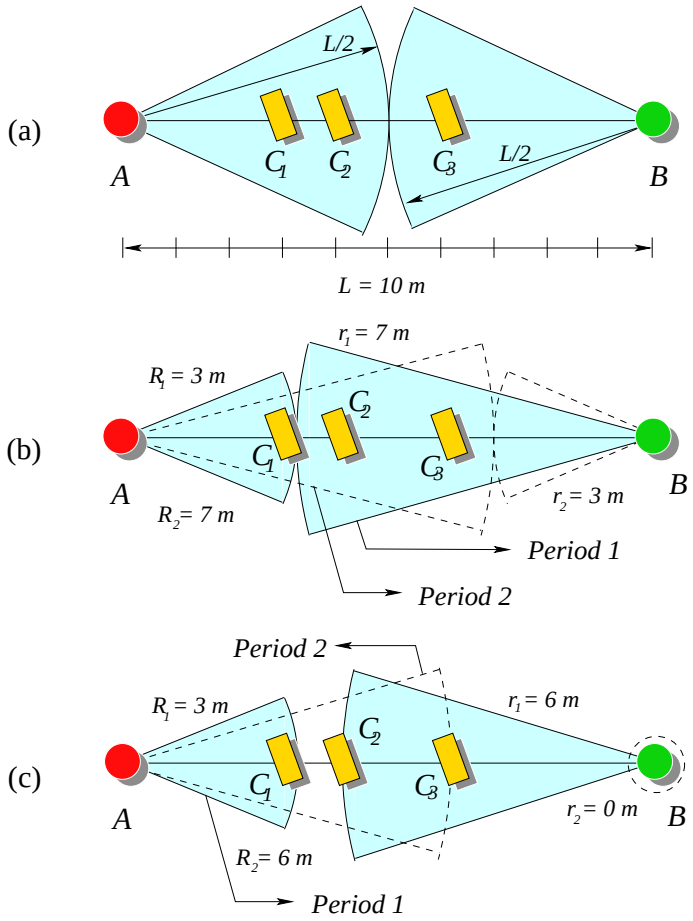


Figure 7.1: Mesh clients C_1, C_2 and C_3 are covered by two routers A and B in the network. (a) Routers adopt the same radius 5m to cover clients. (b) Routers alternatively adopt radii 7m and 3m to cover clients. (c) Two routers reduce their radii to avoid extra power dissipation based on client positions.

scenario where A and B adopt radii of $3m$ and $6m$ in period 1, and $4m$ and $4m$ in period 2, respectively. This example shows that we can effectively enhance network performance by taking advantage of client positions and router battery status. In Section 7.3 we will study the scheduling scheme in detail.

We will present the client driven battery-aware (CDBA) scheme to maximize the lifetime of WMNs in Section 7.3. A MAC layer algorithm will also be given to handle mesh

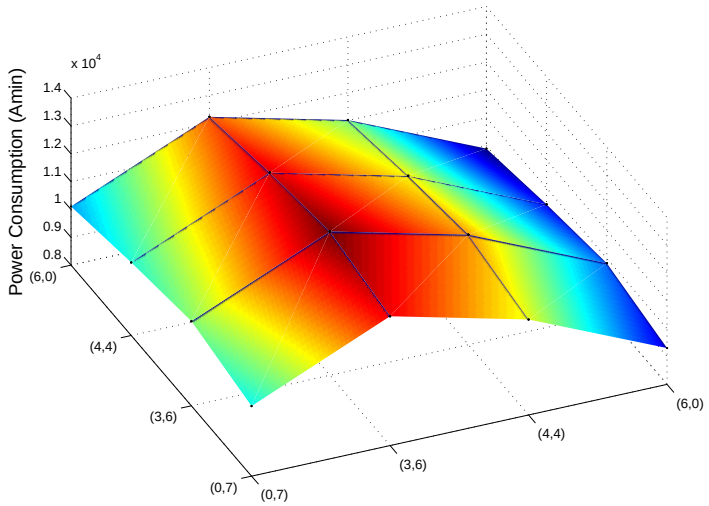


Figure 7.2: Power consumption of various radius pairs of routers *A* and *B* in Fig. 7.1.

client handoff quickly and seamlessly.

7.3 Client Driven Battery-Aware Scheme

In Section 7.2, we demonstrated that network performance can be effectively improved by periodically changing transceiver radii according to battery parameters and client positions. In this section we study the mesh router scheduling and present a client driven battery-aware power scheduling (called CDBA) scheme. The CDBA scheme consists of two parts: CDB MAC algorithm and CDBA coverage algorithm.

Handoff often occurs when mesh clients move through the areas covered by different routers. Specifically, the handoff refers to the moving of a mesh client from one mesh router to another router in the middle of the client's data session which is the transmission of an ongoing packet stream. Consider the scenarios in Fig. 7.3: a mesh client was transmitting a data session through router *A* in the first place but accidentally switches to another router

B before its data session completes. We summarize the mesh client handoff scenarios into the following categories. (i) A client *C* is communicating with router *A*. As it moves out of the current radius of *A*, *C* is still within the maximum coverage of *A* as depicted in Fig. 7.3(a). This is a very common scenario of mesh client handoff. In this situation, *A* can simply increase its radius to cover *C*. (ii) A client *C* moves out of the maximum coverage of *A* during a data session as shown in Fig. 7.3(b). To avoid loss or interruption of service, router *B* must seamlessly and quickly resume *A*'s data connection with *C* as soon as a MAC layer request from *C* is received by *B*. In this scenario there are two ways for *A* to detect that *C* is no longer in its coverage. One way is that *A* waits for a fairly long time threshold to find out that *C* does not reply its MAC requests. However, it is very time consuming and power consuming. In our MAC layer algorithm we adopt another way: *C* constantly reports its GPS positions in the packet head to let *A* monitor the leaving of *C*. We will discuss the details later. (iii) In scenarios (i) and (ii), the handoff of client *C* occurs in the middle of a data session. In these scenarios it is easier for router *A* to detect the leaving of clients as GPS positions are constantly received with packets in data sessions. Fig. 7.3(c), however, illustrates a scenario in which a client *C* completes its data session with *A*, moves to router *B* and starts a new data session. Though this is not an actual handoff as the last session is completed, a MAC layer algorithm should also carefully handle this situation. Since *C* was previously registered with router *A*, router *B* must notify *A* to remove *C*'s registration from it. Hence *C* should provide its information of its previous router in its MAC requests. (iv) In Fig. 7.3(d), *C* moves away from *A* to an area which none of *A* and *B* can cover. In this scenario *A* has to notify its neighbors to locate *C*. Since the network area can be covered by all routers with their maximum radii, it is guaranteed that *C* can be found and covered by at least one neighbors of *A*.

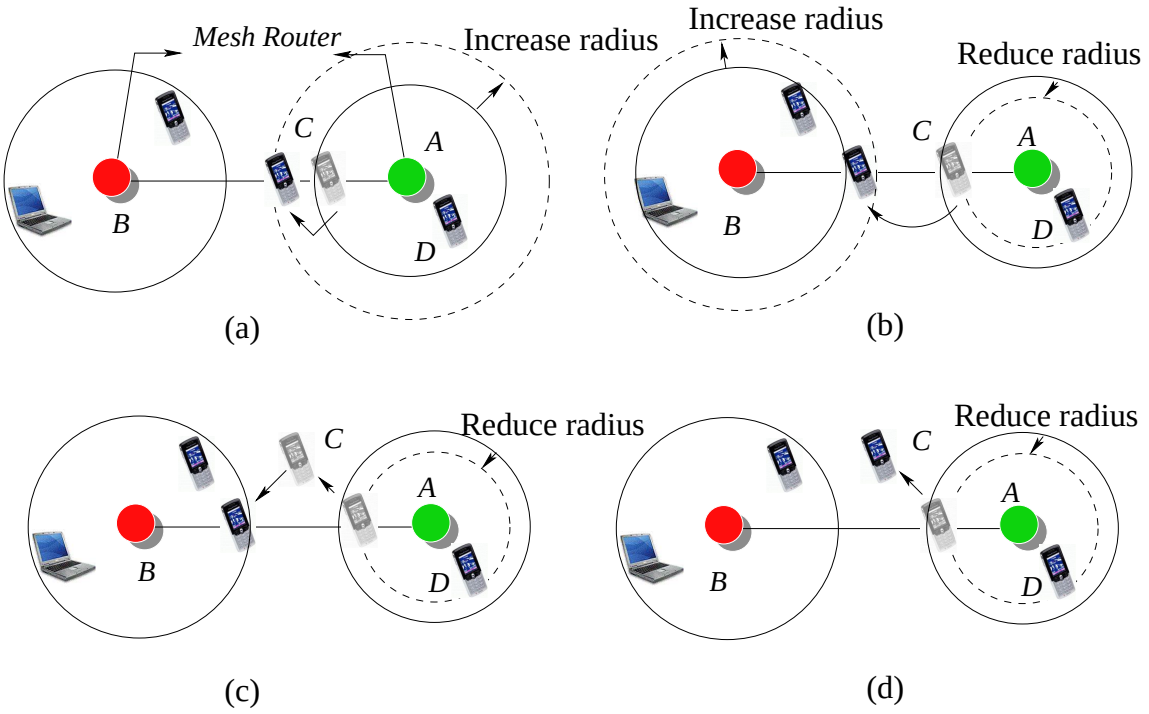


Figure 7.3: Scenarios of mesh client handoff. (a) Client C moves within the maximum coverage of A. (b) Client C moves to B's coverage. (c) Client C moves to B after its last data session is completed with A. (d) Client C moves to a currently unattended area.

We now describe the CDBA scheme for various scenarios in WMNs. The key idea of CDMA scheme is the following. First, a mesh client always encapsulates its GPS positions in the packet head during a data session. Mesh routers can be aware of the leaving of a client based on the changing of its positions. A mesh router calculates the speed and direction of its clients to dynamically reduce or increase its radius. For example, in Fig. 7.3(a) A can increase its radius to cover C. Second, while the router detects that a client is moving out of its maximum radius, it notifies its neighboring routers in that moving direction. Upon receiving such a notification, routers temporarily extend their radii to their maximum radii. As soon as a mesh router locates this client, all other routers reduce their radii back to the

original ones. For example, in Fig. 7.3(b) B increases its radius to cover C and A reduces its radius to cover D. Third, in scenarios that a client completes its data session and moves away as in Fig. 7.3(c), we let the mesh client C report to its new mesh router B to clear its registration with the previous router A. In this case A dynamically reduces its radius upon receiving the notification from B. Finally, routers can extend radii to maximum to detects clients in an unattended area.

CDBA scheme also provides a MAC layer algorithm for efficient handoff of mesh clients. Fig. 7.4 shows the MAC layer communication by CDBA MAC algorithm between mesh router B and its client C. At the beginning, client sets up a connection with router B using RTS (request to send) and CTS (clear to send). C registers with the router and reports its previous router's information at duration P_1. After ACK is confirmed at duration P_2, a data session begins in duration P_3. GPS information is continuously updated to the router during the data session. This process will repeat in the next data session in duration P_i.

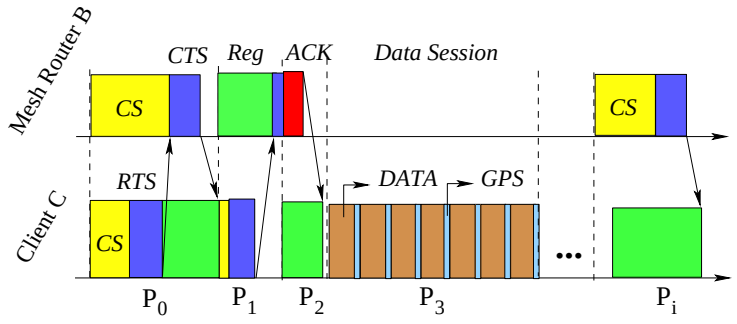


Figure 7.4: MAC layer communication between a mesh router B and its client C.

The CDBA coverage algorithm is summarized in Table 1. The algorithm contains two procedures to be called by a main program. Procedure *SetUpRadius* calculates a radius according to router's battery status and client positions. After obtaining all neighbors'

discharging loss values, a router calculates a feasible radius according to the router's battery status. To ensure that the entire network area is covered, each router chooses the distance between itself and its farthest neighbor for the radius calculation. *SetUpRadius* procedure calls procedure *ClientDriven* to further reduce the router radius based on client positions. The main program periodically sets radii by calling *SetUpRadius* so that each router is able to recover its battery without affecting the function of covering clients. This is a distributed algorithm, in which each mesh router only needs to communicate with at most n neighbors. Thus, the time complexity of the algorithm is $O(n)$.

Table 1: Client driven battery-aware coverage algorithm

Procedure: *ClientDriven*(*float* x){
  receive GPS positions from all covered clients;
  reduce x to x',
    where x' is the distance between i and its farthest client;
  return x';}
Procedure: *SetUpRadius*(){
    Calculate discharging loss of router i based on the battery model;
    Broadcast discharging loss to one hop neighbors;
    Collect n packets from one hop neighbors;
    Calculate its radius $\overline{d}$ based on discharging loss of F,
      where F is its farthest neighbor;
    Adopt $y = ClientDriven(\overline{d})$ as the transceiver radius for δ time;
    Adopt $y = ClientDriven(d - \overline{d})$ as the transceiver radius for δ time,
      where d is the distance between i and F;}
A mesh router i does the following:
  repeat
    Call *SetUpRadius*();
    Monitor x' during the 2δ time,
      where x' is the distance between i and its farthest client:
      if $(x' \leq \overline{d})$ *then* $y = x'$ *else* send a notification packet to neighbors;
    On receiving a notification from neighbors: Call *SetUpRadius*();
  until Power is used up;

7.4 Performance Evaluations

We conducted simulations to evaluate the performance of the CDBA scheme. We assume that the WMN is set up in a $100 \times 100m^2$ field. Wireless mesh routers and access points (AP) are randomly distributed in the field. APs, as portals of packets routing, are connected to the Internet by cables. 30 mesh routers collaboratively cover mobile clients in the area. There are many mesh clients randomly roaming in the area with different speeds from $1m/min$ to $5m/min$. Clients randomly begin their data sessions with routers. To simulate different data services, such as on-line games, video conferences, web service, TCP and UDP connections, we let the lengths of a data session dynamically vary from 10 minutes to 100 minutes. Data packets are routed to APs through other routers. The network lifetime is defined as the duration between the network is set up and the moment when its mesh router layer can no longer cover the entire area. In our simulation we let CDBA adjust transceiver radii every $\delta = 30$min. Mesh router each carries a $1.8 \times 10^4 m$Ahr capacity battery with $\beta = 0.5$.

We compare three protocols in our evaluations: the scheme without battery-aware coverage scheduling (Greedy), the scheme with battery-aware coverage but not client driven (BA) and CDBA scheme. We consider network lifetime and data throughput in our simulations. First we measure the network lifetime for 30 and 50 mesh clients, and the results are plotted in Fig. 7.5 (a). From the figure, we observe that CDBA scheme improves the network lifetime by up to 27.27% compared to the Greedy protocol. We also observe that CDBA scheme achieves up to 12% longer lifetime than the BA protocol. This is due to the power saved by reducing radii based on client positions.

We also evaluate the normalized gross data throughput of the network under three scheduling schemes. The results are shown in Fig. 7.5(b). As can be seen, the CDBA

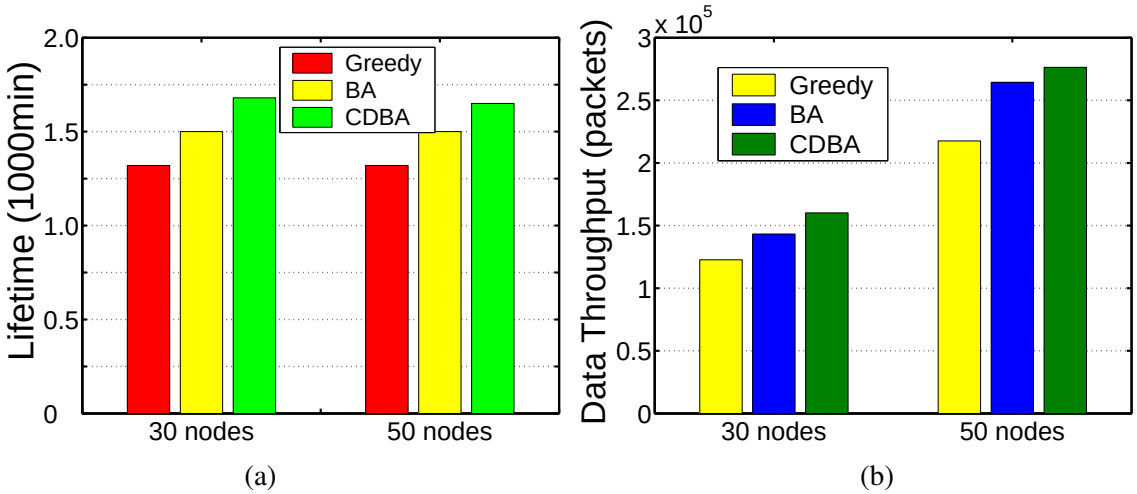


Figure 7.5: Evaluations of CDBA scheme in WMNs. We compare two networks with different numbers of mesh clients: 30 clients and 50 clients. (a) Network lifetime. (b) Data throughput.

improves the total data throughput by up to 30.54% compared to the Greedy protocol. This is because that as network lifetime is prolonged, the gross data throughput increases, thus the overall performance of the WMN is improved. We can also see that compared with the BA protocol, CDBA scheme improves data throughput by up to 11.78%. This is due to the fact that although the BA protocol considers battery status of routers, many routers still waste power on covering no client area. Also, as the number of clients increases from 30 to 50, the gross data throughput naturally increases.

7.5 Summaries

In this chapter we have proposed a client driven battery-aware scheme to schedule mesh routers. We presented an approach by which neighboring routers can collaboratively adjust

their transceivers radii to alternatively recover batteries. Client positions are also considered to efficiently minimize the power consumption at routers. We then presented the CDBA scheme for network-wide mesh client coverage and handoff scheduling. We have conducted simulations to evaluate the performance of the CDBA scheme. The results demonstrate that the CDBA achieves up to 27.27% longer network lifetime and 30.54% more data throughput for WMNs.

Chapter 8

Conclusions

This thesis studied battery-aware and energy-efficient algorithms in wireless mobile ad hoc networks, wireless sensor networks and wireless mesh networks. A suite of battery-aware and energy-efficient algorithms and schemes for routing and scheduling in these networks were presented. First, on-line computable, discrete time mathematical battery models provided approaches to accurately calculate battery discharging loss and battery residual capacity in an energy-efficient way. The calculation of battery discharging loss in the models was simplified. It required low computation complexity and little memory. Secondly, a battery-aware power metric was introduced for battery-aware routing in MANETs. Based on the metric battery-aware routing schemes and prioritized battery-aware routing schemes were proposed to improve energy efficiency of packet routing in MANETs. This thesis also studied battery-awareness for WSNs. A virtual backbone scheduling scheme for data propagation and distribution among sensors was proposed based on the mathematical battery model. The scheme constructed a battery-aware connected dominating set to let fatigue sensors recover batteries and prolong the lifetime of WSNs. A energy-efficient cross-Layer Scheduling for Urban Area WSN was designed for UAHD sensor

networks. A router-level battery lifetime optimization scheduling algorithm was proposed to maximize the lifetime of battery-powered mesh routers. A network-wide spanning tree scheduling algorithm was designed to improve lifetime of WMNs with battery-awareness. A battery-aware MIMO mesh network energy scheduling scheme was proposed to dynamically schedule mesh routers' radii based on battery behaviors. Finally we designed a cross-layer battery-aware client driven for scheduling mesh routers in WMNs.

The performance evaluations of our proposed battery-aware and energy-efficient algorithms demonstrate that they achieve better performance compared with previous algorithms. This research combines protocol design, algorithm design, analytical, probabilistic and simulation techniques to conduct comprehensive studies on the above issues. The research will have a significant impact on fundamental design principles and infrastructures for the development of future wireless networks. The outcome of this project will be applicable to a wide spectrum of applications, including space, military, environmental, health care, home and other commercial areas.

Bibliography

[1] N. Li and J. Hou, "Topology control in heterogeneous wireless networks: problems and solutions," *IEEE Infocom,* 2004.

[2] H. Zhou, C. Yeh and H. Mouftah, "A power efficient medium access control protocol for heterogeneous wireless networks," *IEEE Vehicular Technology Conference (VTC),* 2004.

[3] S. Drew and B. Liang, "Mobility-aware web prefetching over heterogeneous wireless networks," *IEEE Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC),* 2004.

[4] M. Mauve and J. Widmer, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network,* vol. 15 no. 6, pp. 30-39, Nov/Dec 2001.

[5] C.F. Chiasserini and R.R. Rao, "Energy efficient battery management," *IEEE Journal on Selected Areas in Comm.,* vol. 19, no. 7, pp. 1235-45, 2001.

[6] L. Benini, G. Castelli, A. Macii and R. Scarsi, "Battery-driven dynamic power management," *IEEE Design and Test of Computers,* vol. 18, no. 2, pp. 53-60, 2001.

[7] M. Doyle, T. F. Fuller and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *Journal of Electrochemical Society,* vol. 140, no. 6, pp. 1526-33, 1993.

[8] D. Panigrahi, C. Chiasserini, S. Dey and R. Rao "Battery life time estimation of mobile embedded systems," *14th International Conference on VLSI Design,* pp. 57-63, 2001.

[9] D. N. Rakhmatov and S.B.K. Vrudhula, "An analytical high-level battery model for use in energy management of portable electronic systems," *2001 IEEE/ACM International Conference on Computer-Aided Design (ICCAD),* pp. 488-93, 2001.

[10] D. Rakhmatov and S. Vrudhula, "Energy management for battery-powered embedded systems," *ACM Transaction in Embedded Computing Systems (TECS),* vol. 2, no. 3, pp. 277-324, 2003.

[11] L.D. Paulson, "Will fuel cells replace batteries in mobile devices?" *IEEE Computer,* vol. 11, pp. 10-12, 2003.

[12] R. Rao, S. Vrudbula and D.N. Rakbmatov, "Battery modeling for energy-aware system design," *IEEE Computer,* vol. 36, pp. 77-87, 2003.

[13] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA),* pp. 90-100, 1999.

[14] C. Perkins, E. Royer and S. Das, "Ad hoc on demand distance vector (AODV) routing," *IETF Draft,* http://www.ietf.org/internet-drafts/draft-ietf-manetaodv -10.txt, 2002.

[15] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM,* pp. 234-244, 1994.

[16] E.D. Kaplan and C. Hegarty, editors, "Understanding GPS - principles and applications, " *Artech House,* Second Edition, 2005.

[17] T. Melodia, D. Pompili and I. Akyildiz, "Optimal local topology knowledge for energy efficiency geographical routing in sensor networks," *IEEE Infocom,* 2004.

[18] H. Takagi and L. Kleinrock, "Optimal transmission ranges for randomly distributed packet radio terminals," *IEEE Transaction on Communications,* vol.32, no. 3, pp. 246-57, 1984.

[19] T.-C. Hou and V.O.K. Li, "Transmission range control in multihop packet radio networks," *IEEE Transaction on Communications,* vol. 34, no. 1, pp. 38-44, 1986.

[20] T.S. Rappaport and T. Rappaport, "Wireless communications: principles and practice," *Prentice-Hall,* Second Edition, 2001.

[21] Y. Yang and C. Ma, "Battery-aware routing in wireless ad hoc networks – part I: energy model," *19th International Teletraffic Congress (ITC-19),* 2005.

[22] C. Ma and Y. Yang, "Battery-aware routing in wireless ad hoc networks – part II: battery-aware routing," *19th International Teletraffic Congress (ITC-19),* 2005.

[23] C. Ma and Y. Yang, "Battery-aware routing for streaming data transmissions in wireless sensor networks," *ACM/Kluwer Mobile Networks (MONET),* 2006.

[24] C. Ma, Y. Yang and Z. Zhang, "Constructing battery-aware virtual backbones in sensor networks," *International Conference on Parallel Processing (ICPP '05),* pp. 203-210, Oslo, Norway, June 2005.

[25] C. Ma, Z. Zhang and Y. Yang, "Battery-aware router scheduling in wireless mesh networks," *20th IEEE International Parallel and Distributed Processing Symposium (IPDPS),* 2006.

[26] C. Ma and Y. Yang, " A prioritized battery-aware routing protocol for wireless ad hoc networks," *8th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM),* 2005.

[27] C. Ma, M. Ma and Y. Yang, "A battery-aware scheme for energy efficient coverage and routing in wireless MIMO mesh networks," *IEEE Wireless Communications and Networking Conference (WCNC),* Hong Kong, 2007.

[28] C. Ma, M. Ma and Y. Yang, "Data-centric energy-efficient scheduling in densely deployed sensor networks," *IEEE International Conference on Communications (ICC),* pp. 3652-3656, Paris, France, 2004.

[29] C. Ma and Y. Yang, "A cross-layer client driven battery aware scheme in wireless mesh networks," to appear *50th IEEE Global Telecommunications Conference (GLOBECOM),* Washington DC, USA, November, 2007

[30] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine,* pp. 102-114, August 2002.

[31] K.M. Alzoubi, P.-J. Wan and O. Frieder, "Distributed heuristics for connected dominating sets in wireless ad hoc networks," *Journal of Communications and Networks,* vol. 4, no. 1, March 2002.

[32] J. Blum, M. Ding, A. Thaeler and X. Cheng,"Connected Dominating Set in Sensor Networks and MANETs" *Handbook of Combinatorial Optimization*, Kluwer Academic Publishers, pp. 329-369, 2004.

[33] K.M. Alzoubi, P.-J. Wan and O. Frieder, "Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks," *MOBIHOC*, EPFL Lausanne, Switzerland, 2002.

[34] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks". *ACM MobiCom*, July 2001.

[35] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating sets," *ACM/Baltzer J. Wireless Networks*, vol. 1, pp. 255-65, 1995.

[36] A. Ephremides, J. Wieselthier and D. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," *Proceedings of the IEEE*, vol. 75, pp. 56-73, 1987.

[37] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, "Wireless sensor networks for habitat monitoring," *ACM Wireless Sensor Networks and Applications (WSNA),* September 2002.

[38] K. Sohrabi, J. Gao, V. Ailawadhi and G.J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communication,* October 2000.

[39] S. Butenko, X. Cheng, D.-Z.Du and P.M. Pardalos,"On the construction of virtual backbone for ad hoc wireless networks," *Cooperative Control: Models, Applications and Algorithms,* pp.43-54, Kluwer Publishers, 2003.

[40] T. W. Haynes, S. T. Hedetniemi and P. J. Slater, "Fundamentals of domination in graphs (Pure and Applied Mathematics)," *Marcel Dekker,* 1998.

[41] T. S. Rappaport, "Wireless communications," *Prentice Hall,* 1996.

[42] B.N. Clark, C.J. Colbourn and D.S. Johnson, "Unit disk graphs," *Discrete Mathematics,* vol. 86, pp. 165-177, 1990.

[43] Telos Sensor Mote, http://www.moteiv.com/products-tmotesky.php

[44] UCLA AWAIRS I., http://www.janet.ucla.edu/awairs/

[45] V. Raghunathan, C. Schurgers, S. Park and M. Srivastave,"Energy-aware wireless microsensor networks," *IEEE Signal Processing,* pp. 40-49, March 2002.

[46] M. Srivastava, A. Chandrakasan and R. Brodersen, "Predictive system shutdown and other architecture techniques for energy efficient programmable computation," *IEEE Transaction on VLSI Systems,* vol. 4, no. 1, pp. 42-55, 1996.

[47] 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, August 2002.

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

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

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

[51] http://robotics.eecs.berkeley.edu/ pister/SmartDust/, 2004.

[52] http://www-mtl.mit.edu/research/icsystems/uamps/, 2004.

[53] G. Asada, T. Dong, F. Lin, G. Pottie, W. Kaiser and H. Marcy, "Wireless integrated network sensors: low power systems on a chip," *European Solid State Circuits Conference,* The Hague, Netherlands, October 1998.

[54] "Running experiments on ISI wireless sensor network testbed," http://www.isi.edu/scadds/pc104testbed/guideline.html, 2004.

[55] "Crossbow wireless sensor nodes," http://www.xbow.com/

[56] R. Lec, "Piezoelectric biosensors: recent advances and applications," *IEEE International Frequency Control Symposium and PDA Exhibition,* June 2001.

[57] A. Sinha and A. Chandrakasan, "Operating system and algorithmic techniques for energy scalable wireless sensor networks," *2nd International Conference on Mobile Data Management,* January 2001.

[58] "Habitat monitoring on great duck island," http://www.greatduckisland.net/technology.php

[59] E. Shih, S. Cho, N. Lckes, R. Min, A. Sinha, A. Wang and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," *ACM MobiCom,* July 2001.

[60] G.J. Pottie, W.J. Kaiser, "Wireless integrated network sensors," *ACM Communications,* vol. 43, no. 5, pp. 551-558, May 2000.

[61] P.-J. Wan, K.M. Alzoubi and O. Frieder,"Distributed construction of connected dominating sets in wireless ad hoc networks," *IEEE Infocom,* June, 2002.

[62] R. Raz and S. Safra, "A sub-constant error-probability low-degree test and a sub-constant error-probability PCP characterization of NP, *29th ACM Symposium on Theory of Computing (STOC),* pp.475-484, 1997.

[63] A. K. Saha and D. Johnson, "Routing improvements using directional antennas in mobile ad hoc networks," *IEEE Globecom,* 2004.

[64] S. Guo and O. Yang, "Antenna orientation optimization for minimum-energy multicast tree construction in wireless ad hoc networks with directional antennas," *ACM MobiHoc,* 2004.

[65] S. Yi, Y. Pei and S. Kalyanaraman, "On the capacity improvement of ad hoc wireless networks using directional antennas," *ACM MobiHoc,* 2003.

[66] J. Barry, E. Lee and D. Messerschmitt, "Digital communication," *Springer,* 3rd edition, September, 2003.

[67] I. Akyildiz, X. Wang and W. Wang, "Wireless mesh networks: a survey," *Computer Networks Journal (Elsevier),* vol. 47, no. 4, pp. 445–487, March 2005.

[68] R. Bruno, M. Conti and E. Gregori, "Mesh networks: commodity multihop ad hoc networks," *IEEE Communications,* pp. 123-131, 2005.

[69] R. Draves, J. Padhye and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," *ACM MobiCom,* pp. 114-128, 2004.

[70] Firetide Networks Inc., http://www.firetide.com/

[71] K.-D. Lee and V.C.M. Leung, "Low-latency broadcast in multirate wireless mesh networks," *IEEE Journal on Selected Areas in Communications,* vol. 24, no. 11, pp. 2051-2060, November 2006.

[72] "Google wifi," *Google Inc.,* http://wifi.google.com/.

[73] "Wireless broadband initiative," *Suffolk County Government,* http://www.co.suffolk.ny.us/wireless/pdfs/LIWirelessBroadbandInitRFP.pdf.

[74] D. Aguayo, J. Bicket, S. Biswas, D.S.J. De Couto and R. Morris, "MIT roofnet implementation," http://pdos.lcs.mit.edu/roofnet/design/

[75] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* Freeman, 1978.

[76] T. Cormen, C. Leiserson, R. Rivest and C. Stein, *Introduction to Algorithms,* 2nd edition, The MIT Press, Sept. 2001.

[77] X. Li, W. Song and Y. Wang, "Efficient topology control for wireless ad hoc networks with non-uniform transmission ranges," *ACM Wireless Networks,* vol. 11, no. 3, May 2005.

[78] R. G. Little, "A holistic strategy for urban security," *Journal of Infrastructure Systems,* vol. 10, pp. 52-59, June 2004.

[79] C. Gamage, K. Bicakci, B. Crispo and A. S. Tanenbaum, "Security for the mythical air-dropped sensor network," *11th IEEE Symposium on Computers and Communications (ISCC),* 2006.

[80] E. H. Callaway and Jr. E. H. Callaway, "Wireless sensor networks: architectures and protocols," *AUERBACH Press,* pp. 9, Aug. 2003.

[81] J. Deng, Y. S. Han, W. B. Heinzelman and P. K. Varshney, "Scheduling sleeping nodes in high density cluster-based sensor networks," *ACM MONET, Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks,* 2005.

[82] E. Yoneki, "Evolution of ubiquitous computing with sensor networks in urban environments," *Ubicomp - Workshop on Metapolis and Urban Life,* 2005.

[83] C. Intanagonwiwat, D. Estrin, R. Govindan and J. Heidemann, "Impact of network density on data aggregation in wireless sensor networks," *IEEE International Conference on. Distributed Computing Systems (ICDCS),* July 2002.

[84] W. Ye, J. Heidemann and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transaction on Networking,* vol. 12, no. 3, pp. 493-506, June 2004.

[85] M. Yannakakis, "Node-and edge deletion NP-complete problems," *ACM Symposium on Theory of Computing (STOC),* 1978.

[86] X. Hou and D. Tipper, "Gossip-based sleep protocol (GSP) for energy efficient routing in wireless ad hoc networks," *IEEE Wireless Communications and Networking Conference (WCNC),* 2004.

[87] X. Pallot and L.E. Miller, "Implementing message priority policies over an 802.11 based mobile ad hoc network," *Military Communications (MILCOM),* October 2001.

[88] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in ad hoc networks," *IEEE Journal of Selected Areas Communications,* vol. 17, no. 8, pp1488-505, August 1999.

[89] D. Wu and R. Negi, "Utilizing multiuser diversity for efficient support of quality of service over a fading channel," *IEEE Transactions on Vehicular Technology,* vol. 54, no. 3, pp. 1198-1206, May 2005.