Report Number: WUCSE-2006-59

2006-01-01

Unified Power Management in Wireless Sensor Networks, Doctoral Dissertation, August 2006

Guoliang Xing

Radio power management is of paramount concern in wireless sensor networks (WSNs) that must achieve long lifetimes on scarce amount of energy. Previous work has treated communication and sensing separately, which is insufficient for a common class of sensor networks that must satisfy both sensing and communication requirements. Furthermore, previous approaches focused on reducing energy consumption in individual radio states resulting in suboptimal solutions. Finally, existing power management protocols often assume simplistic models that cannot accurately reflect the sensing and communication properties of real-world WSNs. We develop a unified power management approach to address these issues. We first analyze... **Read complete abstract on page 2.**

Unified Power Management in Wireless Sensor Networks, Doctoral Dissertation, August 2006

Guoliang Xing

Complete Abstract:

Radio power management is of paramount concern in wireless sensor networks (WSNs) that must achieve long lifetimes on scarce amount of energy. Previous work has treated communication and sensing separately, which is insufficient for a common class of sensor networks that must satisfy both sensing and communication requirements. Furthermore, previous approaches focused on reducing energy consumption in individual radio states resulting in suboptimal solutions. Finally, existing power management protocols often assume simplistic models that cannot accurately reflect the sensing and communication properties of real-world WSNs. We develop a unified power management approach to address these issues. We first analyze the relationship between sensing and communication performance of WSNs. We show that sensing coverage often leads to good network connectivity and geographic routing performance, which provides insights into unified power management under both sensing and communication performance requirements. We then develop a novel approach called Minimum Power Configuration that ingegrates the power consumption in different radio states into a unified optimization framework. Finally, we develop two power management protocols that account for realistic communication and sensing properties of WSNs. Configurable Topology Control can configure a network topology to achieve desired path quality in presence of asymmetric and lossy links. Co-Grid is a coverage maintenance protocol that adopts a probabilistic sensing model. Co-Grid can satisfy desirable sensing QoS requirements (i.e., detection probability and false alarm rate) based on a distributed data fusion model.

2006-59

Unified Power Management in Wireless Sensor Networks, Doctoral Dissertation, August 2006

Authors: Guoliang Xing

Corresponding Author: xing@wustl.edu

Abstract: Radio power management is of paramount concern in wireless sensor networks (WSNs) that must achieve long lifetimes on scarce amount of energy. Previous work has treated communication and sensing separately, which is insufficient for a common class of sensor networks that must satisfy both sensing and communication requirements. Furthermore, previous approaches focused on reducing energy consumption in individual radio states resulting in suboptimal solutions. Finally, existing power management protocols often assume simplistic models that cannot accurately reflect the sensing and communication properties of real-world WSNs.

We develop a unified power management approach to address these issues. We first analyze the relationship between sensing and communication performance of WSNs. We show that sensing coverage often leads to good network connectivity and geographic routing performance, which provides insights into unified power management under both sensing and communication performance requirements. We then develop a novel approach called Minimum Power Configuration that ingegrates the power consumption in different radio states into a unified optimization framework. Finally, we develop two power management protocols that account for

Type of Report: Other

WASHINGTON UNIVERSITY

THE HENRY EDWIN SEVER GRADUATE SCHOOL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIFIED POWER MANAGEMENT IN WIRELESS SENSOR NETWORKS

by

Guoliang Xing, B.E., M.E., M.S.

Prepared under the direction of Professor Chenyang Lu

A dissertation presented to the Henry Edwin Sever Graduate School of
Washington University in partial fulfillment of the
requirements for the degree of
DOCTOR OF SCIENCE

August 2006

Saint Louis, Missouri

WASHINGTON UNIVERSITY
THE HENRY EDWIN SEVER GRADUATE SCHOOL
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ABSTRACT

UNIFIED POWER MANAGEMENT IN WIRELESS SENSOR NETWORKS

by

Guoliang Xing

ADVISOR: Professor Chenyang Lu

August 2006

Saint Louis, Missouri

Radio power management is of paramount concern in wireless sensor networks (WSNs) that must achieve long lifetimes on scarce amount of energy. Previous work has treated communication and sensing separately, which is insufficient for a common class of sensor networks that must satisfy both sensing and communication requirements. Furthermore, previous approaches focused on reducing energy consumption in individual radio states resulting in suboptimal solutions. Finally, existing power management protocols often assume simplistic models that cannot accurately reflect the sensing and communication properties of real-world WSNs.

We develop a unified power management approach to address these issues. We first analyze the relationship between sensing and communication performance of WSNs. We show that sensing coverage often leads to good network connectivity and geographic routing performance, which provides insights into unified power management under both sensing and communication performance requirements. We then develop

a novel approach called Minimum Power Configuration that integrates the power consumption in different radio states into a unified optimization framework. Finally, we develop two power management protocols that account for realistic communication and sensing properties of WSNs. Configurable Topology Control can configure a network topology to achieve desired path quality in presence of asymmetric and lossy links. Co-Grid is a coverage maintenance protocol that adopts a probabilistic sensing model. Co-Grid can satisfy desirable sensing QoS requirements (i.e., detection probability and false alarm rate) based on a distributed data fusion model.

To Yanni

Contents

List of Tables

List of Figures

Acknowledgments

First, I would like to thank my advisor, Dr. Chenyang Lu, for his inspiration, his enthusiasm, and his great efforts to help me grow as a researcher. He has made my doctoral study at Washington University a journey full of fun and excitements. His guidance and our numerous scientific discussions have greatly improved this work.

I wish to thank Dr. Robert Pless for his constant support and guidance. He gave many insightful and constructive suggestions on this work, and helped me improve my analytical skills. I am very grateful to Dr. Gruia-Catalin Roman who always was available when I needed his advices throughout my D.Sc period.

Part of the thesis work in Chapter 5 was done while I was a research intern at the Palo Alto Research Center Inc. It is a pleasure to thank Dr. Qingfeng Huang, Dr. Ying Zhang, Dr. Markus P.J. Fromherz and all other members in the Embedded Reasoning Area (ERA) group for their hospitality and collaborations.

I wish to thank my wife and best friend, Yanni, for accompanying me through the difficult times, and for her love, understanding and support during my D.Sc period.

Thanks to my fellow students for their collaborations and friendship. I am especially grateful to Sangeeta Bhattacharya, Octav Chipara, Chien-Liang Fok, Kevin Klues, Venkita Subramonian, Xiaorui Wang, and Yuanfang Zhang.

Lastly but not least, I wish to thank my parents for their love and continuous support. I am grateful to my father for my early interest in science and technology.

Guoliang Xing

Washington University in Saint Louis
August 2006

Chapter 1

Introduction

1.1 Motivation

The advances in Micro-Electro-Mechanical Systems (MEMS), low-cost communication and sensing technologies have enabled the deployments of a large number of *sensor nodes* to collect the information about the physical environments. These nodes are typically equipped with CPU, memory, wireless communication interfaces and various sensors. Due to the lack of *a priori* infrastructure support in deployment environments, the nodes must self-organize into networks, referred to as *wireless sensor networks (WSNs)*, and collaboratively sense the useful information from their surrounding areas and deliver to the base stations.

Typical WSN applications include habitat monitoring in remote biological environments [42], structural health monitoring [72], object detection and tracking [35], and so on. Due to the high deployment cost, a WSN must remain operational for a long lifetime (from several months to years) on batteries or limited power supplies like small solar panels. For instance, due to the high cost for embedding sensors in civil infrastructure such as buildings and bridges, a WSN deployed for structural health

monitoring must be able to continuously operate for more than ten years to be economically feasible. Therefore, energy conservation is crucial for WSNs to become viable in real world deployments. In this thesis, we focus on reducing the radio power consumption as radio is the major source of power dissipation in many WSNs [54].

While power management has the potential of significantly extending the system lifetime of WSNs, it may degrade the Quality of Service (QoS) received by users. For example, reducing radio transmission power of nodes, although can lower the energy consumed in packet transmissions, may potentially result in partitioned network or longer communication delay due to reduced radio range. Therefore, a major challenge of power management in WSNs is to achieve satisfactory QoS required by applications while minimizing the total energy consumption of the network.

In sharp contrast to traditional ad hoc networks only concerned with communication performance, WSNs must satisfy both sensing and communication QoS required by applications simultaneously. Such combined performance requirements underlie the integration of multi-hop wireless communication and sensing capabilities on WSN platforms.

- **Sensing QoS requirements:** Since the primary purpose of WSNs is to monitor the environment, they must maintain sufficient quality of sensing even when operating in an energy conservation mode. An important sensing QoS required by many WSN applications is *sensing coverage*. Sensing coverage characterizes the monitoring quality provided by a sensor network in a designated region. Different applications require different degrees of sensing coverage. For example, distributed detection based on data fusion [64] requires every location to

be monitored by multiple nodes, and distributed tracking and classification [35] requires even higher degrees of coverage.

- **Communication QoS requirements:** At the same time, both data fusion among multiple sensors and data services for end-users may have QoS requirements on the communication network. The minimum requirement is that the active nodes must guarantee network connectivity among different nodes whenever they need to communicate. Single connectivity is not sufficient for many sensor networks because a single failure could disconnect the network. Higher connectivity may also be necessary to maintain good throughput by avoiding communication bottlenecks. In addition, many sensing applications also involve real-time observation or interaction with the physical environment and the underlying sensor networks need to deliver sensor data under timing constraints in form of end-to-end deadlines. For example, a surveillance sensor network may need to deliver the information of an evolving wild fire to firefighters in 5 seconds. Otherwise, the firefighters may fail to locate the frontier of the quickly evolving fire. The end-to-end delay is related to the number of hops between source and sink and one-hop communication delay.

Several radio power management approaches can achieve certain communication QoS required by users. *Topology control* [50, 48, 45, 30, 36, 3, 38] aims at preserving network connectivity or low node degree while reducing the total transmission power consumption of a network. *Power-aware routing* [56, 18, 19, 10, 51] can reduce transmission energy consumption of a packet by choosing appropriate routes. *Sleep scheduling* [78, 83, 63] can configure nodes to run in sleep/wakeup duty cycles to reduce the idle listening energy consumption of a network while providing sufficient effective network bandwidth. *Connectivity maintenance* protocols [7, 11, 74] maintain a small set

of active nodes that preserve network connectivity while scheduling all other nodes to sleep.

Recently, several power management approaches have been proposed to achieve sensing QoS requirements. Several centralized solutions [43, 13, 9] can select the minimal set of active nodes to maintain sensing coverage based on the global information of a network. Due to requirements for scalability and fault-tolerance, localized algorithms are more suitable and robust for large-scale WSNs that operate in dynamic environments. Recently, several distributed protocols [61, 75, 6] are proposed to maintain sensing coverage and schedule redundant nodes to sleep.

Although a multitude of solutions have been proposed, existing power management techniques often share the following important limitations.

- **Separate treatment of sensing and communication:** The existing power management approaches have treated sensing and communication QoS in an isolated fashion. Unfortunately, providing sensing or communication QoS alone is not sufficient for a sensor network to satisfy users' requirements. For example, although a connectivity maintenance protocol can guarantee continuous connection of the network through a communication backbone composed of a small number of active nodes, it may lower the network's capability of sensing asynchronous targets due to the reduced density of active nodes. The combination of sensing and communication is a special requirement introduced by sensor networks that integrate multi-hop wireless communication and sensing capabilities into a single platform. On one hand, without enough sensing coverage, the network cannot monitor the environment with sufficient accuracy or may even suffer from "sensing voids" – locations where no sensing can occur.

On the other hand, when the network connectivity is low or the communication latency is high, nodes may not be able to coordinate effectively in distributed sensing tasks or transmit sensor data back to base stations in time. However, to our best knowledge, no unified framework exists to date for power management under both sensing and communication QoS requirements. Moreover, the fundamental relationship between sensing and communication has not been studied. For example, the sensing coverage required by applications determines the minimum number of active nodes and their distribution, which in turns impacts the communication performance of the network. Understanding the relationship between sensing and communication performance requirements will yield key insights into minimizing the total network energy consumption.

- **Power conservation in individual radio states:** The existing power management approaches only aim at reducing the power consumed in a particular radio state. As a result, they are only effective for certain radio platforms and network conditions. Topology control and power-aware routing protocols only reduce the transmission power of radio, and hence are not suitable for the applications with low workload or the radio platforms with high idle power consumption. Backbone maintenance and sleep scheduling protocols, on the other hand, only reduce the idle power consumption, and hence are not effective when the network workload is high or the idle power consumption of radio is low. Clearly, a WSN needs to reduce the energy consumed in each of the radios states (i.e., transmission, reception, and idle) in order to minimize its total energy consumption, which requires effective application all the above approaches.

- **Unrealistic communication and sensing models:** Recent empirical studies [81, 8, 84] revealed that lossy and asymmetric links are common in WSNs. For example, communication over a wireless link may experience a highly probabilistic reception performance. These findings contradicted the widely adopted distance based radio models used in the existing power management protocols. Similarly, the deterministic sensing models (such as the disc model) used by the existing coverage maintenance protocols [9, 61, 65, 75, 76] cannot characterize the stochastic nature of real physical signals. Furthermore, these protocols assume that each sensor performs sensing independently. This assumption is invalidated by many sensor network applications which rely on data fusion to improve sensing performance.

In this thesis work, we propose a *unified* approach for power management in WSNs. We first conduct systematic studies on the fundamental relationship between sensing and communication QoS requirements. Our results show that sensing coverage often leads to good network connectivity. Moreover, simple greedy geographic routing algorithms are shown to perform well in many sensing-covered WSNs. These results provide important insights into unified power management under both sensing and communication QoS requirements. We then propose a new radio power management scheme called *minimum power configuration (MPC)*. In contrast to the existing approaches that treat different radio states (transmission/reception/idle) in isolation, MPC integrates them into a unified optimization framework. We design four approximation algorithms with provable performance bounds and two practical routing protocols that significantly outperform existing power-aware routing and topology control protocols. Finally, to account for realistic properties of WSNs like probabilistic sensing range and communication link quality, we develop two novel power

management protocols. Configurable Topology Control (CTC) is a localized topology control protocol that can configure a network topology to achieve desired path quality bounds in presence of lossy links. Co-Grid is a distributed coverage maintenance protocol that adopts a probabilistic sensing model. Co-Grid can satisfy desirable sensing QoS requirements (i.e., detection probability and false alarm rate) based on a distributed data fusion model.

1.2 Contributions

Specifically, the main contributions of this thesis work are as follows:

- **Analysis of the relationship between coverage and connectivity:** We provide a geometric analysis of the fundamental relationship between coverage and connectivity. Specifically, our analytical results (1) show that sensing coverage implies network connectivity when the sensing range is no more than half of the communication range; and (2) quantify the relationship between the degree of coverage and connectivity. These results give underlying insights for treating coverage and connectivity in a unified power management framework. This is in sharp contrast to several existing works that address the two requirements separately.

- **Analysis of the impact of coverage on geographic routing:** We conduct theoretical analysis on the impact of sensing coverage on the performance geographic routing algorithms. Our results demonstrate that an existing geographic routing algorithm, greedy forwarding (GF), although often fails on random network topologies, can successfully find short routing paths based on

local states in sensing-covered networks. In particular, we derive theoretical upper bounds on the network dilation of sensing-covered networks under GF. We also propose a new greedy geographic routing algorithm called Bounded Voronoi Greedy Forwarding (BVGF) that achieves path dilation lower than 4.62 in sensing-covered networks as long as the communication range is at least twice the sensing range. Furthermore, we extend GF and BVGF to achieve provable performance bounds in terms of total number of transmissions and reliability in lossy networks. Our studies prove that simple greedy geographic routing algorithms like GF and BVGF yield good performance in presence of sensing coverage.

- **Minimum power configuration:** We propose a new power management approach called *Minimum power configuration* (MPC). In contrast to existing approaches that treat different radio states (transmission/reception/idle) in isolation, MPC integrates them in a unified optimization model that considers both the set of active nodes and their transmission power. We have designed a set of approximation algorithms with provable performance bounds, and two practical distributed protocols. In particular, the *Minimum Power Configuration Protocol* (MPCP) can flexibly adapt to a wide range of radio platforms and applications by taking into consideration the power characteristics of the radio and the current workload of the network.

- **Configurable transmission power control:** We propose a new formulation of topology control problem for lossy WSNs based on a new metric called dilation of transmission count (DTC). DTC accounts for lossy links and quantifies the worst-case path quality of a network topology. We develop a set of novel,

localized configurable topology control (CTC) algorithms that can achieve different DTC bounds. CTC has two salient features. It can provide path quality assurance over lossy and asymmetric links in WSNs. Furthermore, it enables applications to achieve desired tradeoff between transmission power and path quality based on their specific requirements.

- **Probabilistic coverage maintenance:** We develop a new coverage maintenance protocol called Co-Grid. In contrast to the existing solutions based on the simple disc sensing model, Co-Grid adopts a probabilistic sensing model that incorporates realistic properties of physical signals like propagation decay. Coverage in Co-Grid is defined based on detection probability and system false alarm rate thresholds that are consistent to the QoS requirements of many sensing applications like event monitoring. Moreover, Co-Grid integrates a distributed detection model that allows for efficient data fusion from multiple nodes.

The rest of the thesis is organized as follows. Related work is reviewed in Section 2. The analyses of the relationship between coverage and connectivity, and the impact of sensing coverage on geographic routing are presented in Section 3 and 4, respectively. We present the minimum power configuration approach to power management in Section 5. Two novel power management schemes that account for realistic communication and sensing properties of WSNs, Configurable Topology Control (CTC) and Co-Grid, are presented in Section 6 and 7, respectively. We conclude the thesis in Section 8.

Chapter 2

Related Work

As one of the most fundamental issues in WSNs, power management has been extensively studied in the literature. In this section, we review different approaches to the power management of WSNs. We first review the existing work on sensing coverage maintenance, followed by a survey on different power management approaches under communication QoS requirements, including topology control, connectivity maintenance, power-aware routing, and sleep scheduling.

2.1 Sensing Coverage Maintenance

Several node deployment strategies are investigated in [13, 9] to provide sufficient coverage for distributed detection based on global information of a network. Due to requirements for scalability and fault-tolerance, localized algorithms are more suitable and robust for large-scale WSNs that operate in dynamic environments. Tian and Georganas [61] proposed a local coverage maintenance algorithm in which each node decides to be active or not based on the geometric calculation of the sponsored sectors from its neighbors. The differentiated surveillance protocol proposed

in [75] is designed to achieve different degrees of coverage by dynamically scheduling nodes' duty cycles. For WSNs to operate successfully, the active nodes must maintain both sensing coverage and satisfactory communication performance. Unfortunately, none of the above coverage maintenance protocols is concerned with communication performance of the network.

Our earlier work on Coverage Configuration Protocol [71] is the first work that aims to provide both provable degree of sensing coverage and network connectivity. Ye et al. [76] proposed the PEAS protocol that also addresses both coverage and connectivity in a configurable fashion. However, nodes in PEAS base their status (active or asleep) on probabilistic probing to its neighbors, and hence does not provide analytical guarantees on the degree of coverage or connectivity. The problem of maintaining both sensing coverage and network connectivity is also studied in [79].

As these works provide initial promise of maintaining both sensing coverage and network connectivity, the fundamental implication of sensing coverage on the communication performance of the network including both network connectivity and routing performance has not been addressed, and is our focus of discussion in Section 4.

2.2 Power-aware Routing

Power-aware routing aims at minimizing the per packet transmission power in multihop routing. Singh et al. proposed five power-aware routing metrics to reduce energy consumption and extend system lifetime [56]. The implementation of a minimum energy routing protocol based on DSR was discussed in [18, 19]. An online power-aware routing scheme is proposed to optimize system lifetime of WSNs in [39]. Chang and

Tassiulas studied the problem of maximizing the lifetime of a network with known data rates [10]. Chang et al. formulated the problem of choosing routes and transmission power of each node to maximize the system lifetime as a linear programming problem and discussed two centralized algorithms [10]. Sankar et al. formulated maximum lifetime routing as a maximum concurrent flow problem and proposed a distributed algorithm [51]. More recently, Dong et al. [17] studied the problem of minimum transmission energy routing in the presence of unreliable communication links.

The major limitation of power-aware routing is that it only minimizes the transmission power of nodes and ignores the power consumption in other radio states. As a result, it is only effective for the radio platforms with high transmission power or the networks with high workload where nodes operate in transmission state in most of time.

2.3 Topology Control

Topology control aims to preserve the desirable properties of a wireless network (e.g., K-connectivity) through reduced nodal transmission power. A comprehensive survey on existing topology control schemes can be found in [52]. Most of the existing algorithms aim at maintaining (strong) network connectivity and several other secondary network properties such as low node degree and bidirectional link connectivity [50, 48, 45, 30, 36, 3]. In the scheme proposed in [50], a node chooses to relay through other nodes only when less power is used. The network is shown to be strongly connected if every node only keeps the links with the nodes in its "enclosure" defined by the relay regions. Ramanathan and Hain [48] proposed two centralized algorithms to

minimize the maximal power used per node while maintaining the (bi)connectivity of the network. Two distributed heuristics were also proposed for mobile networks in [48], although they may not necessarily preserve the network connectivity. Two algorithms are proposed in [45, 30] to maintain network connectivity using the minimal transmission power. CBTC [36] preserves the network connectivity using the minimum power that can reach some node in every cone smaller than $5\pi/6$. A local topology called Localized Delaunay Triangulation (LDT) is shown to have a constant stretch factor with respect to the original network [3]. Li et al. [38] proposed LMST – a MST-based topology control scheme which preserves the network connectivity and has bounded node degree. Li and Hou [37] proposed a localized algorithm that builds a K-vertex connected topology based on the extended Kruskal's MST algorithm [32]. Li et al. [40] proposed a localized algorithm that preserves K-connectivity by having each node choose K closest neighbors in each of the $p \geq 6$ cones. Hajiaghayi et al. [25] proposed three approximate algorithms (two centralized and one distributed) that build the K-connected topology.

Most of the above works aim at maintaining connectivity based metrics of a network through reduced transmission power. However, network connectivity does *not* suffice to provide a satisfactory communication performance when the underlying links among nodes are lossy. Recent empirical studies [81, 1] revealed that lossy communication links are common in wireless sensor networks and 801.11 mesh networks. The simple network model (e.g., the unit disk model) adopted by existing schemes fails to capture the characteristics of such lossy links in real deployments. Furthermore, similar to power-aware routing, topology control only minimizes the transmission power of the network.

2.4 Connectivity maintenance

An effective approach of reducing idle listening power consumption of radios is to maintain a communication backbone composed of a small number of active nodes and schedule other nodes to sleep. ASCENT [7] and SPAN [11] are two adaptive connectivity maintenance protocols in which each node assesses its local connectivity with neighbors and decides whether to join the communication backbone of the network. AFECA [73] is a routing scheme in which each node decides to be active or asleep based on application-level information as well as network density. Xu et al. [74] proposed a connectivity maintenance protocol called GAF that utilizes geographic information to ensure every node in the network is covered by the communication range of at least one active node.

The connectivity maintenance approach has two major limitations: (1) As connectivity maintenance only reduces the idle listening power of nodes, it is only effective when the idle power of the radio is high or the network workload is low. (2) It is only suitable for the networks that are dense enough such that extraneous nodes can be turned off without impairing the network performance.

2.5 Sleep Scheduling

In sleep scheduling, each node operates in a duty cycle composed of wake-up/sleep intervals. A sleep schedule with fixed duty cycle is adopted in 802.11 Power Saving Mode (PSM) [27]. Ye et al. propose a MAC protocol called S-MAC for WSNs [78]. Each node in S-MAC operates in synchronous adaptive duty cycles that can

be extended based on network activity. T-MAC [63] is a MAC protocol that can mitigate the impact of low node duty cycle on network throughput by an adaptive sleep schedule adjustable based on workload variation. Zheng and Kravets [83] proposed a similar on-demand sleep scheduling scheme adaptive to the network workload. The above approaches are based on synchronous sleep schedules that are inherently subject to significant communication delay when packets arrive during sleep time of nodes. Although the adaptive sleep scheduling schemes adopted in [78, 83, 63] can improve the responsiveness of nodes, they are based on heuristics and hence cannot provide guarantee on the worst-case communication delay.

Recently, asynchronous sleep scheduling schemes have been proposed to achieve better trade-offs between energy consumption and communication delay. Zheng et al. [82] proposed an asynchronous sleep scheduling that allows bounded wake-up delay in neighbor discovery. Lu et al. [41] showed that minimizing the end-to-end communication delay for a general topology under asynchronous duty cycle is NP-hard. The authors studied the optimal solutions for several special topologies like trees and rings. Polastre et al. [47] proposed a MAC protocol called B-MAC for WSNs that supports asynchronous sleep scheduling and adopts a low power listening scheme to wake up nodes. The major limitation of asynchronous sleep scheduling is its high overhead in neighbor discovery as each node in the neighborhood has different sleep schedules. Similar to connectivity maintenance, sleep scheduling only reduces the idle listening power of the network.

Chapter 3

Relationship between Coverage and Connectivity

Sensing coverage maintenance has recently received much attention [61, 75, 70, 76, 79]. The basic idea is to choose a small set of active nodes that satisfy the coverage requirement of the application while scheduling all other nodes to operate in sleeping or power saving mode. For a WSN to operate successfully, the active nodes must maintain both sensing coverage and satisfactory communication performance. In this chapter, we analyze the fundamental relationship between sensing coverage and network connectivity. This analysis will provide important insights for treating sensing and communication in a unified power management framework. Our results show that many networks with sensing coverage exhibit desirable network connectivity.

3.1 Preliminaries

We first define the simple sensing and communication models assumed in this section. We then introduce a geometric graph structure called *Voronoi Diagram* that is used

in our analysis. We present our analytical results based on the simple sensing and communication models in Section 3.2, and then extend them to more realistic models in Section 3.3.

3.1.1 Sensing and Communication Models

We now introduce the following simplistic network model that is useful for our initial analysis. Every node v has a circular sensing radius R_s. Any point within R_s of v is *covered* by v. Any two nodes u and v can directly communicate with each other if their Euclidian distance is less than a communication range R_c. Although this simplistic model only represents an limited approximation to the communication/sensing region in real WSN deployments, it allows us to quantify the impact of sensing coverage and develop important insights into the relationship between sensing coverage and communication. We will discuss how our results are extended when these assumptions are relaxed.

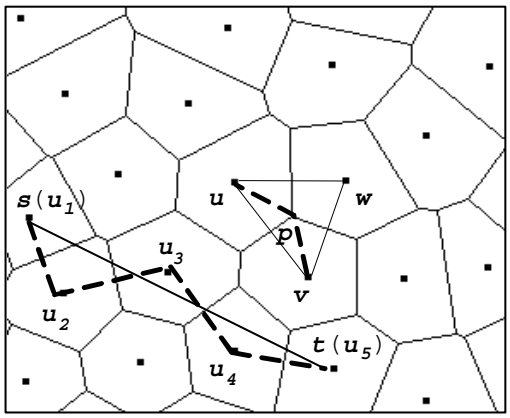


Figure 3.1: The Voronoi diagram of the nodes that 1-cover a region.

3.1.2 Voronoi Diagram

Voronoi diagram is one of the most fundamental structures in computational geometry and has found applications in a variety of fields [4]. For a set of n nodes V in 2D space, the Voronoi diagram of V is the partition of the plane into n *Voronoi regions*, one for each node in V. The Voronoi region of node i ($i \in V$) is denoted by *Vor(i)*. Fig. 3.1 shows a Voronoi diagram of a set of nodes. A point in the plane lies in *Vor(i)* if and only if i is the closest node to the point. The boundary between two contiguous Voronoi regions is called a *Voronoi edge*. A Voronoi edge is on the perpendicular bisector of the segment connecting two adjacent nodes. A *Voronoi vertex* is the intersection of Voronoi edges. As shown in Fig. 3.1, point p is a Voronoi vertex of three contiguous Voronoi regions: *Vor(u)*, *Vor(v)* and *Vor(w)*. We assume that all nodes are in *general positions* (i.e., no four nodes are co-circular).

In the dual graph of Voronoi diagram, Delaunay Triangulation (denoted by $DT(V)$), there is an edge between nodes u and v in $DT(V)$ if and only if the Voronoi regions of nodes u and v share a boundary. $DT(V)$ consists of *Delaunay triangles*. Fig. 3.1 shows a Delaunay triangle uvw. $DT(V)$ is planar, i.e., no two edges cross.

3.2 Coverage vs. Connectivity

We now analyze the relationship between coverage and connectivity. We note that connectivity only requires that the location of any active node to be within the communication range of one or more active nodes such that all active nodes can form a connected communication backbone, while coverage requires all locations in the coverage region to be within the sensing range of at least one active node. Intuitively, the

relationship between connectivity and coverage depends on the ratio of the communication range to the sensing range. However, it is easily seen that a connected network may not guarantee its coverage regardless of the ranges. This is because coverage is concerned with whether any location is uncovered while connectivity only requires that the locations of all active nodes are connected. Hence we focus on analyzing the sufficient condition for a covered network to guarantee connectivity in the rest of this section. A network is K_s covered (or has a K_s coverage) if any point within the deployment region of the network is covered by at least K_s nodes. A network is K_c connected (or has a K_c connectivity) if removing any $K_c - 1$ nodes would not disconnect any two nodes in the network. We have the following theorem:

Theorem 1. *If $R_c \geq 2R_s$, any network that at least 1-covers a convex region is connected.*

Proof. We prove the statement using the Voronoi diagram of the nodes in the network, as shown in Fig. 3.1. Let $Vor(u)$ represent the Voronoi cell of node u. We first prove that any two nodes whose Voronoi cells are adjacent can communicate with each other if $R_c \geq 2R_s$. As illustrated in Fig. 3.1, p is the Voronoi vertex of three adjacent Voronoi cells $Vor(u)$, $Vor(v)$ and $Vor(w)$. According to the definition of Voronoi diagram, u, v and w are equally distant from p and are closest to p among all nodes. Hence, p must be covered by u, v and w, otherwise it will not be covered by any nodes. According to the triangle inequality, we have: $|uv| \leq |pu| + |pv| < 2R_s \leq R_c$.

We now prove the network is connected by showing that there is a communication path between any two nodes s and t in the network. Suppose line segment uv intersects consecutive Voronoi cells $Vor(s) = Vor(u_1), Vor(u_2) \cdots Vor(u_n) = Vor(t)$. For any two consecutive nodes in the series u_1 to u_n, since their Voronoi cells are adjacent, they

can communicate with each other according to the discussion earlier. Hence nodes u_1 to u_n constitute a communication path from s to t. The dotted path between s and t in Fig. 3.1 illustrates such a path. □

Theorem 1 establishes a sufficient condition for a 1-covered network to guarantee 1-connectivity. The coverage requirement for a sensor network also depends on the number of faults that must be tolerated. A network with a higher degree of coverage can maintain acceptable coverage in face of higher rates of node failures. We now extend this result to quantify the relationship between the degree of coverage and connectivity in the following theorem. The proof of the theorem can be found in [71].

Theorem 2. *If $R_c \geq 2R_s$, a network that K_s-covers a convex region A has a connectivity of K_s. The interior nodes whose sensing range does not intersect with the boundary of A have a connectivity of $2K_s$.*

3.3 Extensions to More Realistic Models

We now relax our assumptions made at the beginning of this section and extend our results to more realistic cases. We assume that nodes may have non-uniform and irregular (i.e., possibly non-circular) communication and sensing regions. This assumption is more consistent with WSNs in practice where the communication and sensing range of a node is highly dependent on the environment around the node [81]. We define the following notation. The minimum communication range (MCR) of node v, $R_{cmin}(v)$, is the minimum distance between the location of node v and the boundary of its communication region. The maximum sensing range (MSR) of node v, $R_{smax}(v)$, is the maximal distance between the location of node v and the

boundary of its sensing region. R_{cmin} and R_{smax} represent the minimum MCR and the maximum MSR of all nodes in the network, respectively. $SN(u)$ represents the set of nodes whose sensing region intersect node u's sending region. We have the following theorem.

Theorem 3. *Theorems 1 and 2 hold when* $R_{cmin} \geq 2R_{smax}$.

Proof. Since region A is K_s-covered by the nodes and the actual sensing range of every node is upper-bounded by R_{smax}, A is K_s-covered by the circles that are centered at the nodes and have a radius R_{smax}. Hence Theorems 1 and 2 hold if the communication range of every node is R_{cmin}. From the definition of R_{cmin}, the actual communication range of every node is lower-bounded by R_{cmin}. Hence the results on the network connectivity proved in Theorems 1 and 2 still hold. □

Theorem 3 depends on the knowledge of two global network properties, R_{smax} and R_{cmin}, which may not be easily available in a large-scale sensor network. Furthermore, from Theorem 3, the sufficient condition to guarantee the network connectivity becomes $R_{smax} \geq R_{cmin}$, which may be too conservative for heterogeneous sensor networks where nodes may have different types of network interfaces and/or node modalities. The proof of sufficient condition for network connectivity in Theorem 1 depends on the fact that, when $R_c \geq 2R_s$, any two sensing neighbors can communicate directly. This observation allows us to extend Theorem 1 to the case where nodes have different communication and sensing ranges.

Theorem 4. *Theorem 1 holds when the following condition is satisfied for every node* u*:* $\forall v \in SN(u),\ R_{cmin}(u) \geq R_{smax}(v) + R_{smax}(u)$.

Proof. Let node v be a sensing neighbor of node u. Since the sensing regions of u and v are contained by the circles $C(u, R_{smax}(u))$ and $C(v, R_{smax}(v))$, respectively, the two circles intersect. Hence $|uv| < R_{smax}(u) + R_{smax}(v)$. From the assumption, $R_{cmin}(u) > |uv|$, i.e., node v is within the communication range of node u. Similarly, it can be shown that node u is within the communication range of node v. That is, any two sensing neighbors are connected in the communication graph. For any two nodes i and j, similar to the proof Theorem 1 , it can be shown that a communication path can be constructed along the line segment joining i and j, since any two sensing neighbors whose sensing regions are intersected by line ij can communicate with each other. □

For a sensing-covered network, Theorem 4 gives a sufficient condition for connectivity based on the communication and sensing ranges of sensing neighbors. This condition is less pessimistic than Theorem 3 in heterogeneous network platforms. It also allows a sensing-covered network to determine whether it needs explicit connectivity configuration based on local states.

3.4 Integrated Coverage and Connectivity Maintenance

Our analyses (Theorems 1-4) suggest that many networks with coverage have good connectivity. This result has important implication on integrating coverage and connectivity maintenance into a unified power management framework. Under the condition that $R_c \geq 2R_s$, the active nodes that cover the network deployment region can

also guarantee the network connectivity, which justifies the approach of many coverage maintenance protocols that does not consider the network connectivity. However, when $R_c < 2R_s$, a coverage maintenance protocol alone fails to maintain both sensing coverage and network connectivity. In our earlier work [71], we proposed to integrate our coverage maintenance protocol CCP with a connectivity maintenance protocol SPAN [11] to achieve both requirements.

Chapter 4

Impact of Coverage on Greedy Geographic Routing

We have shown in Chapter 3 that sensing coverage can guarantee network connectivity when communication range is at least twice sensing range. In this chapter, we present our analytical results on the routing quality of the networks with sensing coverage.

We focus our analysis on geographic routing. Geographic routing is a suitable routing scheme for WSNs as communication in WSNs is often addressed by physical locations. For example, instead of querying a sensor with a particular ID, a user often queries a geographic region. Furthermore, geographic routing makes efficient routing decisions based on local states (e.g., locations of one-hop neighbors), which enables it to scale to large distributed applications. As the simplest form of geographic routing, greedy forwarding (GF) is particularly attractive for WSNs. In GF, a node always forwards a packet to the neighbor closest to the destination. Due to the low overhead, GF can be easily implemented on resource constrained sensor network platforms. However, GF fails when a node cannot find a better neighbor than itself. Earlier research has shown such routing void is common on random network topologies. In this section,

we present geometric analysis that demonstrates GF is a viable and effective routing scheme in sensing-covered networks. We also propose a new greedy geographic routing algorithm called Bounded Voronoi Greedy Forwarding (BVGF) that outperforms GF in sensing-covered networks.

4.1 Network Model and Performance Metric

The ratio between the communication range, R_c, and the sensing range, R_s, has a significant impact on the routing quality of the network. Intuitively, as the ratio increases, a sensing-covered network becomes denser, resulting in better connectivity and routing performance. The ratio varies across a wide range due to the heterogeneity of sensor networks. As a starting point for the analysis, we focus on those networks with the *double range property*, i.e., $R_c/R_s \geq 2$. This assumption is motivated by the geometric analysis in Section 3, which showed that a sensing-covered network is always connected if it has the double range property. As shown in [68], the double range property is applicable to a number of representative sensing applications. In the rest of this Chapter, we assume a sensing-covered network always satisfies the double range property.

We use *network dilation* to quantify the performance of a routing algorithm on a WSN. It quantifies the performance of the algorithm relative to the *ideal* case in which the path between any two nodes u and v has $\left\lceil \frac{|uv|}{R_c} \right\rceil$ hops. Formally, the network dilation of network $G(V, E)$ under the routing algorithm R is defined as follows.

$$D_n = \max_{u,v \in V} \frac{\tau_G(u,v)}{\left\lceil \frac{|uv|}{R_c} \right\rceil} \tag{4.1}$$

where $\tau_G(u,v)$ represents the shortest path (in terms of hop count) between u and v found by the routing algorithm R.

4.2 Greedy Forwarding

Our first analysis is concerned with the performance of GF on sensing-covered networks. Greedy forwarding (GF) is an efficient, localized ad hoc routing scheme employed in many existing geographic routing algorithms [29, 60, 21]. Under GF a node makes routing decisions only based on the locations of its (one-hop) neighbors, thereby avoiding the overhead of maintaining global topology information. In each step a node forwards a packet to the neighbor with the shortest Euclidean distance to the destination [29, 21]. An alternative greedy forwarding scheme [60] chooses the neighbor with the shortest projected distance to the destination on the straight line joining the current node and the destination.

However, a routing node might encounter a *routing void* when it cannot find a neighbor that is closer (in term of Euclidean or projected distance) to the destination than itself. In such a case, the routing node must drop the packet or enter a more complex recovery modes [29, 33, 58] to route the packet around the routing void. In this section we prove GF always succeeds in sensing-covered networks when the double-range property is satisfied. We further derive the upper bound on the network dilation of sensing-covered networks under GF.

Theorem 5. *In a sensing-covered network, GF can always find a routing path between any two nodes. Furthermore, in each step (other than the last step arriving at the destination), a node can always find a next-hop node that is more than $R_c - 2R_s$ closer (in terms of both Euclidean and projected distance) to the destination than itself.*

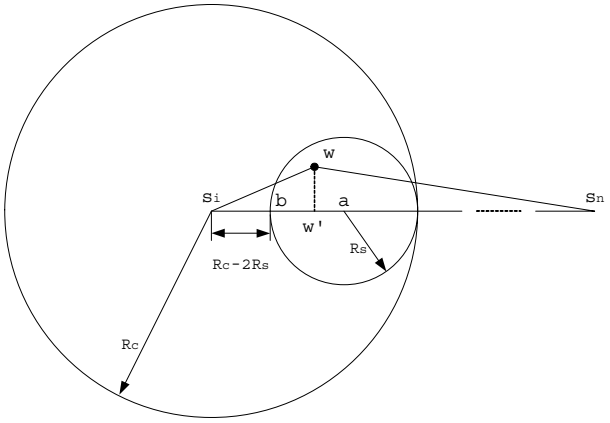


Figure 4.1: GF always finds a next-hop node in sensing-covered networks.

Proof. Let s_n be the destination, and s_i be either the source or an intermediate node on the GF routing path, as shown in Fig. 4.1. If $|s_i s_n| \leq R_c$, the destination is reached in one hop. If $|s_i s_n| > R_c$, we find point a on $\overline{s_i s_n}$ such that $|s_i a| = R_c - R_s$. Since $R_c \geq 2R_s$, point a must be outside of the sensing circle of s_i. Since a is covered, there must be at least one node, say w, inside the circle $C(a, R_s)$.

We now prove the progress toward destination s_n (in terms of both Euclidean and projected distance) is more than $R_c - 2R_s$ by choosing w as the next hop of s_i. Let point b be the intersection between $\overline{s_i s_n}$ and $C(a, R_s)$ that is closest to s_i. Since circle $C(a, R_s)$ is internally tangent with the communication circle of node s_i, $|s_i b| = R_c - 2R_s$. Clearly, the maximal distance between s_n and any point on or inside circle $C(a, R_s)$ is $|s_n b|$. Suppose w' is the projection of node w on line segment $\overline{s_i s_n}$. We

have:

$$|s_n s_i| - |s_n w'| \geq |s_n s_i| - |s_n w| > |s_i b| = R_c - 2R_s \geq 0$$

From above relation, we can see that both the projected distance and the Euclidean distance in one hop (other than the last hop arriving at the destination) of a BVGF routing path is more than $R_c - 2R_s$. Thus GF always can find a routing path between any two nodes. □

Theorem 5 establishes that the progress toward the destination in each step of a GF routing path is lower-bounded by $R_c - 2R_s$. Therefore, the network length of a GF routing path between a source and a destination is upper-bounded.

Theorem 6. *In a sensing-covered network, GF can always find a routing path between source u and destination v no longer than $\left\lfloor \frac{|uv|}{R_c - 2R_s} \right\rfloor + 1$ hops.*

Proof. Let N be the network length of the GF routing path between u and v. The nodes on the path are $s_0(u)$,$s_1 \cdots s_{n-1}$,$s_n(v)$. From Theorem 5, we have

$$\begin{aligned} |s_0 s_n| - |s_1 s_n| &> R_c - 2R_s \\ |s_1 s_n| - |s_2 s_n| &> R_c - 2R_s \\ &\vdots \\ |s_{n-2} s_n| - |s_{n-1} s_n| &> R_c - 2R_s \end{aligned}$$

Summing all the equations above, we have:

$$|s_0 s_n| - |s_{n-1} s_n| > (N-1)(R_c - 2R_s)$$

Given $|s_0 s_n| = |uv|$, we have:

$$\begin{aligned} N &< \frac{|uv| - |s_{n-1}s_n|}{R_c - 2R_s} + 1 \\ &< \frac{|uv|}{R_c - 2R_s} + 1 \end{aligned} \tag{4.2}$$

Hence $N \leq \left\lfloor \frac{|uv|}{R_c - 2R_s} \right\rfloor + 1$ □

From Theorem 6 and (4.1), the network dilation of a sensing-covered network $G(V, E)$ under GF satisfies:

$$D_n(GF) \leq \max_{u,v \in V} \left(\frac{\left\lfloor \frac{|uv|}{R_c - 2R_s} \right\rfloor + 1}{\left\lceil \frac{|uv|}{R_c} \right\rceil} \right) \tag{4.3}$$

The *asymptotic* network dilation bound of sensing-covered networks under GF can be computed by ignoring the rounding and the constant term 1 in (4.3).

Corollary 1. *The asymptotic network dilation of sensing-covered networks under GF satisfies*

$$\tilde{D}_n(GF) \leq \frac{R_c}{R_c - 2R_s} \tag{4.4}$$

From (4.4), the dilation upper bound monotonically decreases when R_c/R_s increases. It becomes lower than 2 when $R_c/R_s > 4$, and approaches 1 when R_c/R_s becomes

very large. This result confirms our intuition that a sensing-covered network approaches an ideal network in terms of network length when the communication range is significantly longer than the sensing range.

However, the GF dilation bound in (4.4) increases quickly to infinity when R_c/R_s approaches 2. In the proof of Theorem 5, when R_c approaches $2R_s$, a forwarding node s_i may be infinitely close to the intersection point between $C(a, R_s)$ and $\overline{s_i s_n}$. Consequently, s_i may choose a neighbor inside $C(a, R_s)$ that makes an infinitely small progress toward the destination and hence result in a long routing path. Similar to the proof of Theorem 5.1 in [22], it can be shown that the network length of a GF routing path between source u and destination v is bounded by $O((\frac{|uv|}{R_c})^2)$. From (4.1), we can see that this result cannot lead to a constant upper bound on the network dilation for a given range ratio. Whether GF has a tighter analytical network dilation bound when R_c/R_s is close to two is an open research question left for future work.

4.3 Bounded Voronoi Greedy Forwarding (BVGF)

From Sections 4.2, we note that although GF has a satisfactory network dilation bound on sensing-covered networks when $R_c/R_s \gg 2$, the bound becomes very large when R_c/R_s is close to two. This result motivates us to develop a new routing algorithm, Bounded Voronoi Greedy Forwarding (BVGF), that has a satisfactory analytical dilation bound for any $R_c/R_s > 2$.

4.3.1 The BVGF Algorithm

Similar to GF, BVGF is a localized algorithm that makes greedy routing decisions based on one-hop neighbor locations. When node i needs to forward a packet, a neighbor j is eligible as the next hop only if the line segment joining the source and the destination intersects *Vor(j)* or coincides with one of the boundaries of *Vor(j)*. BVGF chooses as the next hop the neighbor that has the shortest Euclidean distance to the destination among all eligible neighbors. When there are multiple eligible neighbors that are equally closest to the destination, the routing node randomly chooses one as the next hop. Fig. 4.2 illustrates four consecutive nodes ($s_i \sim s_{i+3}$) on the BVGF routing path from source u to destination v. The communication circle of each node is also shown in the figure. We can see that a node's next hop on a routing path might not be adjacent with it in the Voronoi diagram (e.g., node s_i does not share a Voronoi edge with node s_{i+1}). When $R_c \gg R_s$, this greedy forwarding scheme allows BVGF to achieve a tighter dilation bound than the DT bound that only considers DT edges and does not vary with the range ratio.

The key difference between GF and BVGF is that BVGF only considers the neighbors whose Voronoi regions are intersected by the line joining the source and the destination. As we will show later in this section, this feature allows BVGF to achieve a tighter upper-bound on the network dilation in sensing-covered networks.

In BVGF, each node maintains a neighborhood table. For each one-hop neighbor j, the neighborhood table includes $j's$ location and the locations of the vertices of $Vor(j)$. For example, as illustrated in Fig. 4.2, for one-hop neighbor s_i, node s_{i+1} includes in its neighborhood table the locations of s_i and the vertices of $Vor(s_i)$ (denoted by crosses in the figure). To maintain the neighborhood table, each node periodically

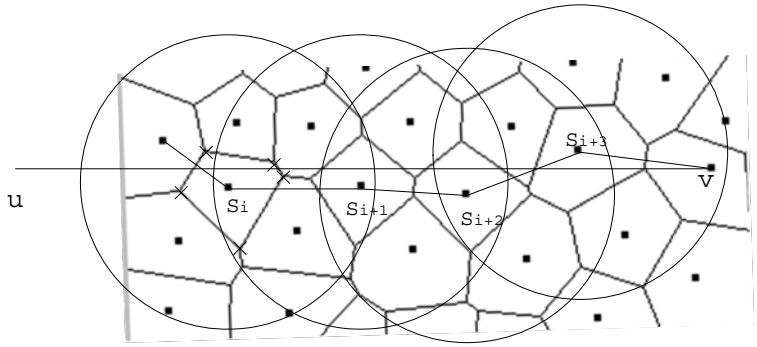


Figure 4.2: A routing path of BVGF

broadcasts a beacon message that includes the locations of itself and the vertices of its Voronoi region. Note each node can compute its own Voronoi vertices based on its neighbor locations because all Voronoi neighbors are within its communication range (as proved in Theorem 1).

Assume the number of neighbors within a node's communication range is bounded by $O(n)$. The complexity incured by a node to compute the Voronoi diagram of all its one-hop neighbors is $O(n \log n)$ [4]. Since the number of vertices of the Voronoi region of a node is bounded by $O(n)$ [4], the total storage complexity of a node's neighborhood table is $O(n^2)$.

4.3.2 Network Dilation of BVGF

In this section, we analyze the network dilation of BVGF. We first prove that BVGF can always find a routing path between any two nodes in a sensing-covered network (Theorem 7). We next show that a BVGF routing path always lies in a *Voronoi forwarding rectangle*. We then derive lower bounds on the projected progress in every step of a BVGF routing path (Lemma 3). Since this lower bound is not tight when

R_c/R_s is close to two, we derive the tighter lower bounds on the projected progress in two and four consecutive steps on a BVGF routing path (Lemmas 6 and 7). Finally we establish the asymptotic bounds of the network dilation of sensing-covered networks under BVGF in Theorem 9.

In the rest of this section, to simplify our discussion on the routing path from source u to destination v, we assume node u is the origin and the straight line joining u and v is the *x-axis*. The *Voronoi forwarding rectangle* of nodes u and v refers to the rectangle defined by the points $(0, R_s)$, $(0, -R_s)$, $(|uv|, -R_s)$ and $(|uv|, R_s)$. Let $x(a)$ and $y(a)$ denote the x-coordinate and y-coordinate of a point a, respectively. The projected progress between two nodes is defined as the difference between their x-coordinates.

Theorem 7. *In a sensing-covered network, BVGF can always successfully find a routing path between any two nodes. Furthermore, the projected progress in each step of a BVGF routing path is positive.*

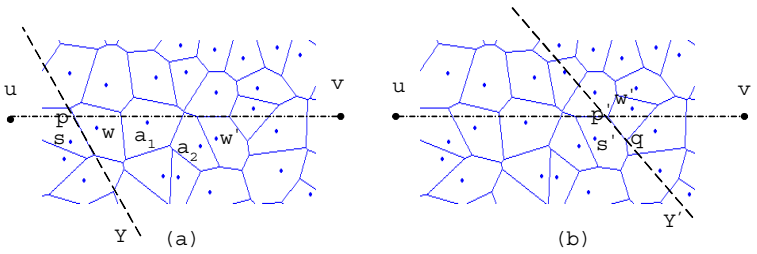


Figure 4.3: BVGF always finds a next-hop node

Proof. As illustrated in Fig. 4.3, node s_i is an intermediate node on the BVGF routing path from source u to destination v. *x-axis* intersects $Vor(s_i)$ or coincides with one of the boundaries of $Vor(s_i)$. Let point p be the intersection between $Vor(s_i)$ and the

x-axis that is closer to v (if *x-axis* coincides with one of the boundaries of $Vor(s_i)$, we choose the vertex of $Vor(s_i)$ that is cloest to v as point p). There must exist a node w such that $Vor(s_i)$ and $Vor(w)$ share the Voronoi edge that hosts p and intersects the *x-axis*. The straight line (denoted as dotted line in Fig. 4.3) where the Vornoi edge lies on defines two half-planes P_i and P_{i+1}, and $s_i \in P_i, w \in P_{i+1}$. From the definition of Voronoi diagram, any point in half-plane P_{i+1} has a shorter distance to w than to s_i. Since $v \in P_{i+1}$, $|wv| < |s_iv|$. In addition, since $|s_iw| < 2R_s \leq R_c$ (see Theorem 1) and line segment $\overline{uv}$ intersects $Vor(w)$ (or coincides with one of the boundaries of $Vor(w)$), w is eligible to be the next hop of s_i. That is, s_i can find at least one neighbor (w) closer to the destination. This holds for every node other than the destination and hence BVGF can always find a routing path between the source and the destination.

We now prove the projected progress in each step of a BVGF routing path is positive. We discuss two cases. 1) If s_i chooses w as the next hop on the BVGF routing path, from the definition of Voronoi diagram, s_i and w lies to the left and the right of the perpendicular bisector of line segment $\overline{s_iw}$, respectively. Therefore, $x(s_i) < x(p) < x(w)$ and hence the projected progress between s_i and w is positive. 2) If s_i chooses node s_{i+1} (which is different from w) as the next hop, we can construct a consecutive path (along the *x-axis*) consisting of the nodes s_i, $a_0(w)$, $a_1 \cdots a_m$, s_{i+1} such that any two adjacent nodes on the path share a Voronoi edge that intersects the *x-axis*, as illustrated in Fig. 4.3. Similar to case 1), we can prove:

$$x(s_i) < x(a_0) < \cdots < x(a_m) < x(s_{i+1})$$

Hence the projected progress between the consecutive nodes s_i and s_{i+1} on the BVGF routing path is positive. □

In a sensing-covered convex region, any point is covered by the node closest to it. This simple observation lead to the the following Lemma.

Lemma 1 (Coverage Lemma)**.** *A convex region A is covered by a set of nodes V if and only if each node can cover its Voronoi region (including the bounary).*

Proof. The nodes partition the convex region A into a number of Voronoi regions in the Voronoi diagram. Clearly, if each Voronoi region (including the boundary) is covered by the node within it, region A is sensing-covered. On the other hand, if region A is covered, any point in region A must be covered by the *closest* node(s) to it. In the Voronoi diagram, all the points in a Voronoi region share the same closest node. Thus every node can cover all the points in its Voronoi region. Any point on the boundary of two Voronoi regions $Vor(i)$ and $Vor(j)$ has the same distance from i and j and is covered by both of them. □

BVGF always forwards a packet to a node whose Voronoi region is intersected by the *x-axis*. From Lemma 1, every Voronoi region in a sensing-covered network is within a sensing circle. Therefore, the nodes on a BVGF routing path lie in a bounded region. Specifically, we have the following Lemma.

Lemma 2. *The BVGF routing path from node u to node v lies in the Voronoi forwarding rectangle of nodes u and v.*

Proof. As illustrated in Fig. 4.4, s_i is an intermediate node on the BVGF routing path between u and v. Let point w be one of the intersections between the *x-axis* and

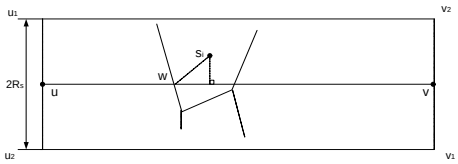


Figure 4.4: Voronoi forwarding rectangle.

Vor(s_i) (if *x-axis* coincides with one of the boundaries of *Vor*(s_i), choose a vertex on the boundary as point w). From Lemma 1, node s_i covers point w, and hence $|s_i w| < R_s$. We have $|y(s_i)| \leq |s_i w| < R_s$. Furthermore, from Theorem 7, $0 < |x(s_i)| < |uv|$. Thus, s_i lies in the Voronoi forwarding rectangle of nodes u and v. □

In a sensing-covered network, the greedy nature of BVGF ensures that a node chooses a next hop that has the shortest distance to the destination among all eligible neighbors. On the other hand, according to Lemma 2, the next-hop node must fall in the Voronoi forwarding rectangle. These results allow us to derive a lower bound on the progress of every step on a BVGF routing path.

Lemma 3 (One-step Advance Lemma)**.** *In a sensing-covered network, the projected progress in each step of a BVGF routing path is more than* Δ_1*, where* $\Delta_1 = \max(0, \sqrt{R_c^2 - 2R_c R_s} - R_s)$.

Proof. As illustrated in Fig. 4.5, s_i is an intermediate node on the BVGF routing path between source u and destination v. Let point s_i' be the projection of s_i on the *x-axis*. From Lemma 2, $s_i s_i' < R_s$. Let point d be the point on the *x-axis* such that $|s_i d| = R_c - R_s$. According to Lemma 1, there must exist a node, w, which covers point d and $d \in$*Vor(w)*. Clearly w lies in circle $C(d, R_s)$. Since d is on the *x-axis* and $d \in$*Vor(w)*, *x-axis* intersects *Vor(w)*. Furthermore, since circle $C(d, R_s)$

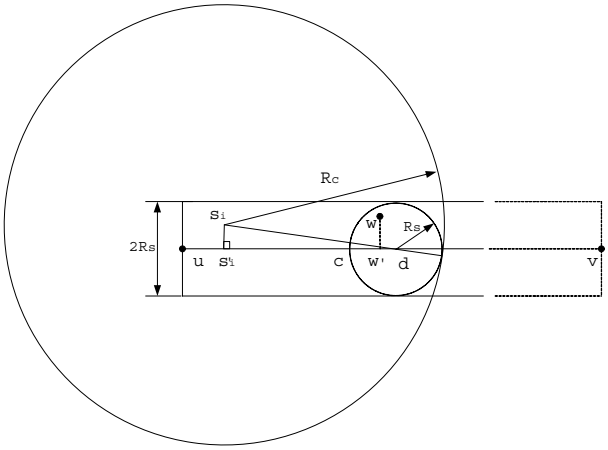


Figure 4.5: One-step projected progress of BVGF

is internally tangent with the communication circle of node s_i, node w is within the communication range of node s_i. Therefore node s_i can at least choose node w as the next hop. Let c be the intersection between $C(d, R_s)$ and *x-axis* that is closest to u. Let w' be the projection of w on the *x-axis*. The projected progress between s_i and w is:

$$\begin{aligned} |s_i'w'| > |s_i'c| &= |s_i'd| - R_s \\ &= \sqrt{|s_id|^2 - |s_is_i'|^2} - R_s \\ &> \sqrt{(R_c - R_s)^2 - R_s^2} - R_s \\ &= \sqrt{R_c^2 - 2R_cR_s} - R_s \end{aligned}$$

$|s_i'w'| \leq 0$ whe $R_c/R_s \leq 1+\sqrt{2}$. From Theorem 7, projected progress made by BVGF in each step is positive. Therefore, the lower bound on the projected progress in each step is $\max(0, \sqrt{R_c^2 - 2R_cR_s} - R_s)$. □

From Lemma 3, we can see that the lower bound on the projected progress between any two nodes on a BVGF routing path approaches zero when $R_c/R_s \leq 1 + \sqrt{2}$. We ask the question whether there is a tighter lower bound in such a case. Consider two non-adjacent nodes i and j on a BVGF routing path. The Euclidean distance between them must be longer than R_c because otherwise BVGF would have chosen j as the next hop of i which contradicts the assumption that i and j are non-adjacent on the routing path. We refer to this property of BVGF as the *non-adjacent advancing property*[1]. We have the following Lemma (the proof can be found in [69]).

Lemma 4 (Non-adjacent Advancing Property)**.** *In a sensing-covered network, the Euclidean distance between any two non-adjacent nodes on a BVGF routing path is longer than R_c.*

The non-adjacent advancing property, combined with the fact that a BVGF routing path always lies in the Voronoi forwarding rectangle, leads to the intuition that the projected progress toward the destination made by BVGF in two consecutive steps is lower-bounded. Specifically, we have the following Lemma that establishes a tighter bound on the projected progress of BVGF than Lemma 3 when R_c/R_s is small.

Lemma 5. *In a sensing-covered network, the projected progress between any two non-adjacent nodes i and j on a BVGF routing path is more than:*

$$\begin{cases} \sqrt{R_c^2 - R_s^2} & \text{if } i, j \text{ on the same side of the x-axis} \\ \sqrt{R_c^2 - 4R_s^2} & \text{if } i, j \text{ on different sides of the x-axis} \end{cases}$$

[1]Similarly, GF also can be shown to have this property.

Proof. Let $s_0(u)$,$s_1 \cdots s_{n-1}$,$s_n(v)$ be the consecutive nodes on the BVGF routing path between source u and destination v. From Lemma 4, $|s_i s_{i+k}| > R_c$ $(k > 1)$. Fig. 4.6(a) and (b) illustrate the two cases where s_i and s_{i+k} are on the same or different sides of the *x-axis*, respectively. Both s_i and s_{i+k} lie in the Voronoi forwarding rectangle of nodes u and v (dotted box in the figure). When s_i and s_{i+k} are on the same side of the *x-axis*, we have

$$|y(s_{i+k}) - y(s_i)| < R_s$$

The projected progress between s_{i+k} and s_i satisfies:

$$\begin{aligned} x(s_{i+k}) - x(s_i) &= \sqrt{|s_i s_{i+k}|^2 - (y(s_{i+k}) - y(s_i))^2} \\ &> \sqrt{R_c^2 - R_s^2} \end{aligned}$$

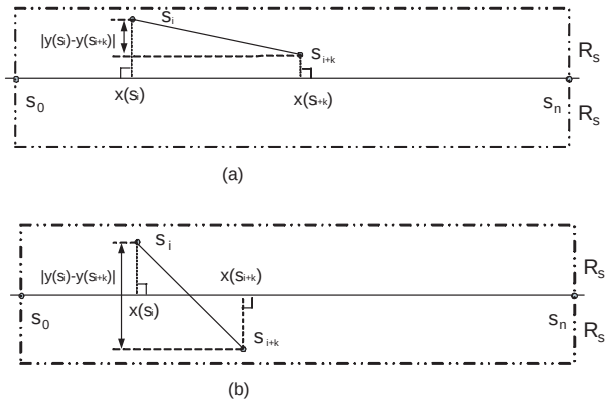


Figure 4.6: Projected progress of two non-adjacent nodes

Similarly, when s_i and s_{i+k} are on different sides of the *x-axis* as shown in Fig. 4.6(b), we can prove that the projected progress between them is more than $\sqrt{R_c^2 - 4R_s^2}$. □

From Lemma 5, we can see that the worst-case projected progress in two consecutive steps on a BVGF routing path occurs when the non-adjacent nodes on the two steps are on the different sides of the *x-axis*. We have the following Lemma (proof can be found in [68]).

Lemma 6 (Two-step Advance Lemma)**.** *In a sensing-covered network, the projected progress in two consecutive steps on a BVGF routing path is more than* Δ_2*, where* $\Delta_2 = \sqrt{R_c^2 - 4R_s^2}$.

Combining the different cases of non-adjacent node locations, we can derive the lower bound on the projected progress made by BVGF in four consecutive steps.

Lemma 7 (Four-step Advance Lemma)**.** *In a sensing-covered network, the projected progress in four consecutive steps of a BVGF routing path is more than* Δ_4*, where*

$$\Delta_4 = \begin{cases} \sqrt{R_c^2 - R_s^2} & (2 \leq R_c/R_s \leq \sqrt{5}) \\ \sqrt{4R_c^2 - 16R_s^2} & (R_c/R_s > \sqrt{5}) \end{cases}$$

Proof. Let $s_0(u)$,$s_1 \cdots s_{n-1}$,$s_n(v)$ be the consecutive nodes on the BVGF routing path between source u and destination v. s_i, s_{i+2} and s_{i+4} are three non-adjacent nodes on the path. Without loss of generality, let s_i lie above the *x-axis*. Fig. 4.7(a)(b)(c)(d) show all possible configurations of s_i, s_{i+2} and s_{i+4} (the dotted boxes denote the Voronoi forwarding rectangles). We now derive the lower bound on the projected progress between s_i and s_{i+4}.

1).When s_i and s_{i+4} lie on different sides of the *x-axis*, as illustrated in Fig. 4.7(a)(b), the projected progress δ_{ab} between s_i and s_{i+4} is the sum of the projected progress

between s_i and s_{i+2} and the projected progress between s_{i+2} and s_{i+4}. From Lemma 5 :

$$\delta_{ab} = \sqrt{R_c^2 - R_s^2} + \sqrt{R_c^2 - 4R_s^2}$$

2).When s_i and s_{i+4} lie on the same side of the *x-axis*, as shown in Fig. 4.7(c)(d), from Lemma 5, the projected progress between them is more than $\delta_{cd} = \sqrt{R_c^2 - R_s^2}$. On the other hand, the projected progress can be computed as the sum of the projected progress between s_i and s_{i+2} and the projected progress between s_{i+2} and s_{i+4}, i.e., $\delta_c = 2\sqrt{R_c^2 - 4R_s^2}$ as shown in Fig. 4.7(c) or $\delta_d = 2\sqrt{R_c^2 - R_s^2}$ as shown in Fig. 4.7(d). Since $\delta_d > \delta_c$, the $\max\{\delta_{cd}, \delta_c\}$ is the lower bound on the projected progress between s_i and s_{i+4} when they lie on the same side of the *x-axis*.

Summarizing the cases 1) and 2), the lower bound on the projected progress in four consecutive steps on a BVGF routing path is

$$\Delta_4 = \min\{\delta_{ab}, \max\{\delta_{cd}, \delta_c\}\}$$

From the relation between δ_{ab}, δ_{cd} and δ_c, Δ_4 can be transformed to the result of the theorem. □

When R_c/R_s is small, the network is relatively sparse. Although the one-step projected progress approaches zero as shown in Lemma 3 in such a case, interestingly, Lemmas 6 and 7 show that the projected progress toward the destination made by BVGF in two or four consecutive steps is lower-bounded. On the other hand, when

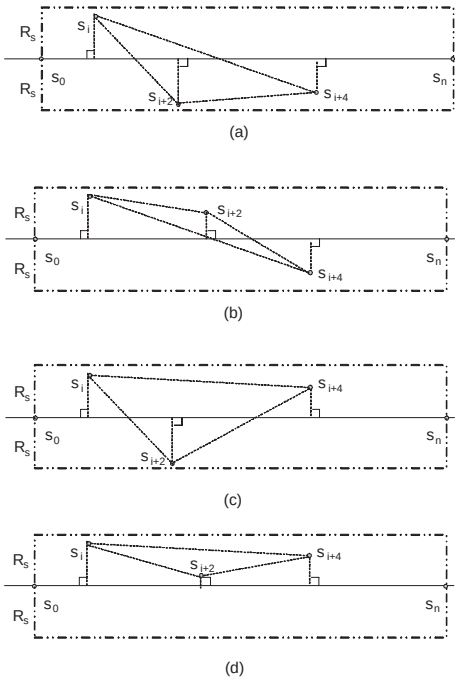


Figure 4.7: Projected Progress in Four Consecutive Steps

$R_c \gg R_s$, the sensing coverage of the network can result in a high density of nodes in the communication range of a routing node and hence the projected progress of BVGF in each step approaches R_c. In such a case the lower bound established in Lemma 3 is tighter than the lower bounds established in Lemmas 6-7.

Based on the 1-hop, 2-hop and 4-hop minimum projected progress derived in Lemmas 3, 6 and 7, respectively, we can derive the upper bounds on the network length of a BVGF routing path. Summarizing these upper bounds, we have the following theorem (the proof can be found in [68]).

Theorem 8. *In a sensing-covered network, The BVGF routing path between any two nodes u and v is no longer than* Δ *hops, where* $\Delta = \min\left\{\left\lceil \frac{|uv|}{\Delta_1} \right\rceil, 2\left\lfloor \frac{|uv|}{\Delta_2} \right\rfloor + 1, 4\left\lfloor \frac{|uv|}{\Delta_4} \right\rfloor + 3\right\}$.

From Theorem 8 and (4.1),the network dilation of a sensing-covered network $G(V, E)$ under BVGF satisfies:

$$D_n(BVGF) \leq \max_{u,v \in V} \frac{\Delta}{\left\lceil \frac{|uv|}{R_c} \right\rceil} \tag{4.5}$$

where Δ is defined in Theorem 8. The *asymptotic* bound on network dilation of sensing-covered networks under BVGF can be computed by ignoring the rounding and the constant terms in (4.5).

Theorem 9. *The asymptotic network dilation of a sensing-covered network under BVGF satisfies*

$$\tilde{D_n}(BVGF) \leq \begin{cases} \frac{4R_c}{\sqrt{R_c^2 - R_s^2}} & (2 \leq R_c/R_s \leq \sqrt{5}) \\ \frac{2R_c}{\sqrt{R_c^2 - 4R_s^2}} & (\sqrt{5} < R_c/R_s \leq 3.8) \\ \frac{R_c}{\sqrt{R_c^2 - 2R_cR_s} - R_s} & (R_c/R_s > 3.8) \end{cases} \tag{4.6}$$

4.3.3 Summary of Analysis on Network Dilations

In this section we summarize the network dilation bounds based on the deterministic communication model. Fig. 4.8 in Section 4.5 shows the asymptotic dilation bounds of GF and BVGF under different range ratios. The asymptotic bound of BVGF is competitive for all range ratios no smaller than two. The bound gets the worst-case value $\frac{8\sqrt{3}}{3} \approx 4.62$ when $R_c/R_s = 2$. That is, BVGF can always find a routing path between any two nodes u and v within $4.62 \left\lceil \frac{|uv|}{R_c} \right\rceil$ hops. The asymptotic network dilation bound of GF increases quickly with the range ratio and approaches infinity when R_c/R_s is close to two. Whether there is a tighter bound for GF in such a case is an open research question. When $R_c/R_s >\sim 3.5$, the network dilations of GF and

BVGF are very similar because the network topology is dense and both algorithms can find very short routing paths.

4.4 Extension to a Probabilistic Communication Model

The theoretical analysis and protocol design discussed in previous sections are based a simplistic communication model that assumes a deterministic communication range. Recent empirical studies showed that real sensor network platforms (e.g., Berkeley motes) yield unreliable links and irregular communication ranges [81]. A wireless sensor network must handle communication failures due to unreliable links. GF is shown to yield poor performance in lossy networks because it always chooses the node closest to the destination as the next hop, which often results in a long but unreliable communication link [53]. In this section, we extend our results to a probabilistic communication model that captures these characteristics.

In the probabilistic communication model, the quality of a communication link from node u to node v is described by *packet reception rate (PRR(u,v))* that is defined as the ratio of the number of successful transmissions from u to v to the total number of transmissions from u to v. Note that $PRR(u, v)$ may not equal $PRR(v, u)$ since the communication quality of a link is often asymmetric. In practice the PRR of a link can be estimated either offline or online. For example, in the MT routing protocol on TinyOS [67] a node computes the PRR of the link from a neighbor to itself by monitoring the reception statistics of periodic beacon messages broadcast by the neighbor.

4.4.1 Routing Algorithms with ARQ

When a node fails to deliver a packet to the next hop (e.g., indicated by a missing ACK from the receiver), it retransmits the packet through an automatic repeat request (ARQ) mechanism. We assume ARQ keeps retransmitting a packet until successful reception by the next hop node. In this section, we discuss efficient variants of GF and BVGF when ARQ is employed. The case without ARQ is discussed in Section 4.4.2. Recently, Kuruvila et al. studied several efficient routing metrics in presence of ARQ, including the product of PRR and progress traversed toward the destination, for wireless networks with a lossy physical layer [34]. Product of PRR and progress is also shown to be optimal in terms of energy efficiency for GF in [53] in lossy networks. This new metric achieves a better energy-efficiency than distance by balancing the hop count and path reliability. Both GF and BVGF can be modified to use this metric as follows. Instead of choosing the neighbor closest to destination among all routing candidates, node u chooses as the next hop a node v that maximizes $(|ut| - |vt|) \cdot PRR(u, v)$ where t is the destination. We denote the variants of GF and BVGF based on this new metric as GF_e and BVGF_e, respectively.

We extend the double range property presented in Section 4.1 as follows. For a given parameter p $(0 < p \leq 1)$, we define the *probabilistic communication range* $R_c(p)$ as the distance within which the link of any two nodes has a PRR no lower than p. The extended double range property can be formulated as $R_c(p) \geq 2R_s$. Similar arguments as the proofs of Theorem 5 and 7 can show that both GF_e and BVGF_e always find a routing path between any two nodes if $\exists p > 0$ *s.t.* $R_c(p) \geq 2R_s$. We note that the notation of $R_c(p)$ is only for the purpose of performance bound analysis. The operation of the GF_e and BVGF_e does not require the knowledge of $R_c(p)$.

The analysis presented in Section 4.3 focused on hop count and network dilation. However, in the probabilistic communication model, hop count does not indicate the quality of a routing path due to unreliable links. When ARQ is present, the energy cost and end-to-end delay of a routing path depends on the total number of transmissions needed to successfully delivery a packet from source to destination. Hence the total number of transmissions is a more accurate metric to describe the quality of a routing path. Before extending the analytical results based on the number of transmissions, we define the following notation. $\tilde{D}_n(GF)$ and $\tilde{D}_n(BVGF)$ represent the asymptotic network dilation bounds by replacing R_c with $R_c(p)$ in (4.4) and (4.6). For a given routing algorithm, Δ_i represents the progress toward the destination made at the i^{th} step of the algorithm; p_i represents the PRR of the link chosen at the i^{th} step; Δ' represents the minimum progress toward the destination made by the algorithm if only considering the neighbors within $R_c(p)$. We have the following theorem regarding the performance of $\mathrm{GF_e}$ and $\mathrm{BVGF_e}$.

Theorem 10. *In a sensing-covered network that satisfies the double range property for a probabilistic communication range $R_c(p)$, the asymptotic expected total number of transmissions used by algorithm A_e (A is GF or BVGF) to deliver a packet between two nodes u and v is no smaller than $\tilde{D}_n(A) \cdot |uv|/(p \cdot R_c(p))$ in presence of ARQ.*

Proof. According to the definition of $R_c(p)$, the PRR of any link within $R_c(p)$ is no lower than p. Since A_e chooses the next-hop node that has the maximum product of progress and PRR, we have:

$$\forall i, \Delta_i \cdot p_i \geq \Delta' \cdot p \tag{4.7}$$

For a link with PRR p_i, the expected number of transmissions is equal to $1/p_i$. From (4.7), the total number of transmissions between source s and destination t satisfies the following inequality:

$$\sum_i \frac{1}{p_i} \quad \leq \quad \frac{\sum_i \Delta_i}{\Delta' \cdot p} = \frac{|st|}{\Delta' \cdot p} \tag{4.8}$$

According to the definition of network dilation, $\Delta' = R_c(p)/\tilde{D}_n(A)$. Replacing Δ' in (4.8) gives the form in the statement of the theorem. □

Theorem 10 shows that both GF_e and BVGF_e can find routing paths with bounded number of transmissions in presence of ARQ in lossy networks.

4.4.2 Routing Algorithms without ARQ

In this section, we discuss efficient variants of GF and BVGF without support of ARQ. When ARQ is not implemented, a node drops a packet if it fails to deliver it to the next-hop node. The quality of a routing path can be quantified by *end-to-end reliability* defined as the probability that a packet can be successfully transmitted from source to destination along the path. It can be seen that the end-to-end reliability of a routing path is equal to the product of the PRR of each link on the path. We propose a new metric $(|ut| - |vt|)/\ln\frac{1}{PRR(u,v)}$ (where u, v and t are routing node, a neighbor of u and destination, respectively) that provides lower-bounded end-to-end reliability when used with GF and BVGF. We refer to GF and BVGF based on this new metric as GF_r and BVGF_r, respectively. We have the following theorem regarding the performance of GF_r and BVGF_r.

Theorem 11. *In a sensing-covered network that satisfies the double range property for a probabilistic communication range $R_c(p)$, the asymptotic end-to-end reliability of the path found by algorithm A_r (A is GF or BVGF) is no lower than $e^{\tilde{D_n}(A)\cdot|uv|\cdot\ln p/R_c(p)}$.*

Proof. Since A_r always chooses the next-hop node that has the maximum $\Delta_i/\ln\frac{1}{p_i}$, we have:

$$\forall i, \frac{\Delta_i}{\ln 1/p_i} \geq \frac{\Delta'}{\ln 1/p} \tag{4.9}$$

From (4.9), we have the following inequality:

$$\sum_i \ln p_i \quad \geq \quad \frac{\sum_i \Delta_i \cdot \ln p}{\Delta'} = \frac{|st| \cdot \ln p}{\Delta'} \tag{4.10}$$

From (4.10), the reliability of a routing path found by A_r between source s and destination t satisfies $\prod_i p_i \geq e^{\frac{|st|\cdot\ln p}{\Delta'}}$. Replacing Δ' with $R_c(p)/\tilde{D_n}(A)$ (by the definition of $\tilde{D_n}(A)$) gives the form in the statement of the theorem. □

Theorem 11 shows that both $\mathrm{GF_r}$ and $\mathrm{BVGF_r}$ can find routing paths with bounded end-to-end reliability in absence of ARQ.

4.5 Simulation Results

In this section we present our simulation results. The purpose of the simulations is twofold. First, we compare the dilations of GF and BVGF under different range ratios and investigate the tightness of the dilation bounds we derived in previous

sections. We then study the average performance of the algorithms under a realistic radio model of the Mica2 motes [15].

4.5.1 Results under the Deterministic Communication Model

Our first set of simulations are based on the deterministic communication model. 1000 nodes are randomly distributed in a $500m \times 500m$ region that is covered by a set of active nodes chosen by the Coverage Configuration Protocol (CCP) [65]. Redundant nodes are turned off for energy conservation. All nodes have the same sensing range of $20m$. We vary R_c to measure the network dilations of GF and BVGF under different range ratios. As discussed in Section 4.2, the routing metric of GF is based on Euclidean or projected distance. Since the results of the two metrics are very similar, only Euclidean distance based results are presented in this section.

The results presented in this section are averages of five runs on different network topologies produced by CCP. In each round, a packet is sent from each node to every other node in the network. As expected, all packets are delivered by both algorithms. The network lengths are logged and the dilations are then computed using Definition 4.1. To distinguish the simulation results from the dilation bounds we derived in previous sections, we refer to the former as *measured dilations*. We should note that the measured dilations characterize the average-case performance of the routing algorithms in the particular network topologies used in our experiments, which may differ from the worst-case bounds for *all* possible sensing-covered network topologies we derived in previous sections.

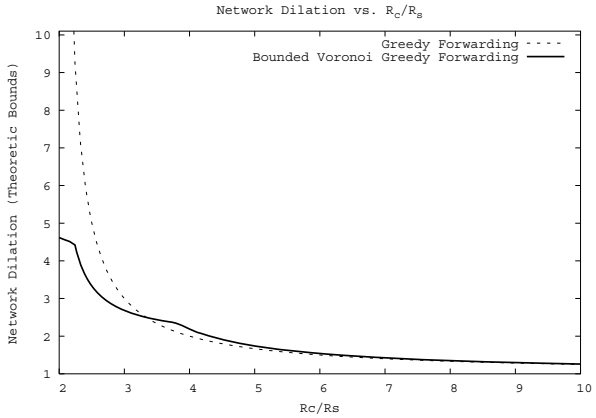


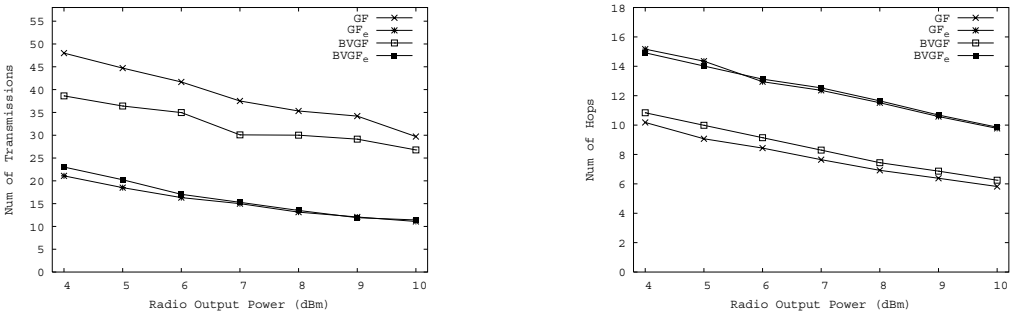
Figure 4.8: Network Dilations

Fig. 4.8 shows that the measured dilations of GF and BVGF remain close to each other. Both algorithms have very low dilations (smaller than two) in all range ratios no smaller than two. When R_c/R_s increases, the measured dilations of both algorithms approach their asymptotic bounds. When R_c/R_s is close to 2, however, the difference between the asymptotic bounds and the corresponding measurements becomes wider. This is because, the worst-case scenarios from which the dilation bounds are derived are rare when the network is less dense. Due to the rounding errors in deriving the asymptotic dilation bounds (Corollary 1 and Theorem 9), the measured dilations are slightly higher than the asymptotic bounds when $R_c/R_s > 6$, as shown in Fig. 4.8. This is because when R_c becomes large, the routing paths become very short and the effect of rounding in the calculation of network dilations becomes significant. The result also indicates that the measured dilation of GF is significantly lower than the asymptotic bound derived from our analysis. Whether GF has a tighter network dilation bound is left for our future work.

The simulation results have shown that the proposed BVGF algorithm performs similarly with GF in average cases. In addition, the upper bounds on the network dilations of BVGF and GF established in previous sections are tight when R_c/R_s is large.

4.5.2 Results under the Probabilistic Communication Model

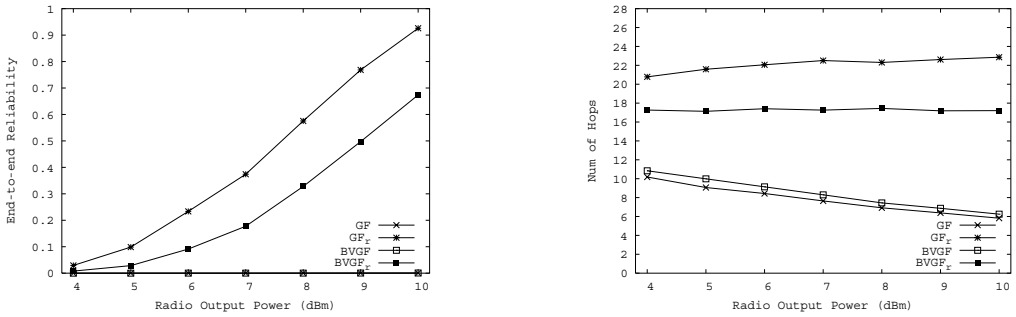
In this section, we evaluate the performance of the extended versions of GF and BVGF algorithms discussed in Section 4.4 in lossy networks. To simulate the probabilistic link reception quality, we implemented the link layer model from USC [85]. Previous empirical data shows that the USC model accurately simulates the unreliable links between Mica2 motes [85]. In our simulations, the PRR of a link is governed by the USC model according to the distance between the two nodes and the transmission power. A packet is sent using different routing algorithms between any two nodes that are more than $350m$ apart. A node ignores the neighbors whose links have a PRR lower than 10%. Previous study [67] showed that such "blacklisting" strategy can significantly improve the packet delivery performance. The rest of simulation settings is same with those in Section 4.5.1.



(a) Average num of transmissions per path (b) Average hop count per path

Figure 4.9: Performance with ARQ under the probabilistic communication model.

We first evaluate the performance of our algorithms with ARQ. Fig. 4.9(a) and (b) show the average number of transmissions and hops under different radio transmission powers set according to the specification of Mica2 mote [15]. The minimum power is chosen such that all neighbors within $2R_s$ of a node have PRRs above 10% and thus are not blacklisted. Consistent with our analysis, this condition made all algorithms successfully deliver all packets. Fig. 4.9(a) shows that all algorithms yielded fewer transmissions when the transmission power increases. GF_e and BVGF_e perform substantially better than GF and BVGF although GF and BVGF used fewer hops shown by Fig. 4.9(b). This result confirms the observation that product of progress and PRR is an efficient metric for greedy forwarding in lossy networks [53]. BVGF_e yielded similar performance as GF_e, which is consistent to the results based on deterministic communication model.



(a) Average end-to-end reliability per path (b) Average hop count per path

Figure 4.10: Performance w/o ARQ under the probabilistic communication model.

We now evaluate the performance of our algorithms without ARQ. Fig. 4.10(a) and (b) show the average end-to-end reliability and hop counts of different algorithms. We can see that both GF and BVGF yielded near zero end-to-end path reliability although they used fewer hops. This is because they tend to choose long links which, however, are more likely to be unreliable. In contrast, both GF_r and BVGF_r achieved higher end-to-end reliability as transmission power increases since the PRR of the link

at each hop becomes higher. Although GF_r used more hops than $BVGF_r$ as shown in Fig. 4.10(b), GF_r performs slightly better than $BVGF_r$ as it has more next hop candidates and hence higher chance of choosing more reliable links.

The overall results in this section show that the routing metrics that consider both progress and PRR are more efficient than purely progress based metric in lossy networks. The extended GF and BVGF based on these metrics can achieve satisfactory performance in terms of number of transmissions and reliability on sensing-covered networks with unreliable communication links.

Chapter 5

Minimum Power Configuration

A limitation of existing radio power management approaches is that they only aim at reducing the power consumed in a particular radio state. As a result, they are only effective for partial radio platforms and network conditions. *Topology control* [50, 48, 45, 30, 36, 3, 38] only reduces the transmission power of a network by adjusting the transmission range of each node while still preserving necessary network properties (e.g., connectivity). Similarly, *power-aware routing* [56, 18, 19, 10, 51] reduces the transmission power of the nodes that lie on the communication routes. Both topology control and power-aware routing reduce the energy consumption only when the radio interface is actively transmitting/receiving packets. Such approaches are only effective on the radio platforms with high transmission/reception power or networks with high workload. However, many radio interfaces (e.g., the CC1000 radio on Mica2 motes [15] and WLAN cards [11]) also consume non-negligible power even if they are idle. Furthermore, many WSN applications generate low to moderate workload, resulting most energy dissipated in idle state of the radio. *Backbone maintenance* [7, 11, 73, 74, 71] reduces the idle energy consumption by turning off redundant nodes while only keeping a small number of active nodes as relays for multi-hop communication. *Sleep scheduling* [78, 83, 63] reduces the idle energy consumption by scheduling nodes to

operate at duty cycles composed of on/off intervals. However, backbone maintenance and sleep scheduling only reduce the energy consumed by idle listening, and hence are not suitable for the radio platforms with low idle power or the networks with high workload.

In order to extend the system life of WSNs, we must effectively minimize the *total* energy consumed in all radio states for different platforms and network workload conditions. However, this goal cannot be met by combining the existing approaches in a straightforward fashion because the power-minimal configuration depends on both radio characteristics and dynamic network workload conditions. For example, when network workload is low, the power consumption of a WSN is dominated by the idle state. In such a case, scheduling nodes to sleep saves the most power. It is therefore more power-efficient for active nodes to use long communication ranges since it will require fewer nodes to remain awake in order to relay packets. Conversely, short radio ranges may be preferable when the network workload is high, as the radio tends to spend more time in the transmission and reception states. We propose a novel approach called *minimum power configuration (MPC)* that minimizes the aggregate energy consumption in all power states by taking into consideration both radio characteristics and network workloads. In sharp contrast to earlier research that treated topology control, power-aware routing, and backbone maintenance in isolation, MPC provides a unified approach that integrates them as a joint optimization problem. The total power consumption is then minimized by jointly configuring a set of active nodes and their transmission power.

5.1 An Illustrating Example

In this section, we illustrate the basic idea of our approach with a simple example. We focus on the energy consumption of radios since they tend to be the major source of power dissipation in wireless networks. We will show that when the total energy from each of the different radio states is considered, the optimal network configuration depends on the radio characteristics and data rates of the network. A wireless radio can work in one of the following states: transmitting, receiving, idle, and sleeping. The corresponding power consumptions are represented by $P_{tx}(d)$, P_{rx}, P_{id} and P_s, where d is the Euclidean distance of the transmission.

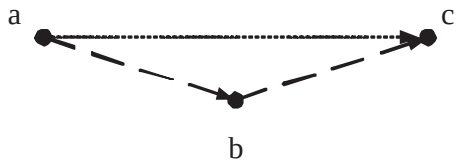


Figure 5.1: Two communication paths from a to c: $a \rightarrow c$ or $a \rightarrow b \rightarrow c$.

As shown in Fig. 5.1, a, b and c are three nodes located in 2D space. a needs to send data to c at the rate of R bps. The bandwidth of all nodes is B bps. There are two network configurations to accomplish the communication between a and c: 1) a communicates with c directly using transmission range $|ac|$ while b remains sleeping or 2) a communicates with b using transmission range $|ab|$ and b relays the data from a to c using transmission range $|bc|$. Minimizing the total energy of all nodes in the network is equivalent to minimizing the average power consumption of all radio states. We denote the average power consumption under the two configurations as P_1 and P_2, respectively. P_1 and P_2 can be computed as follows:

$$
\begin{aligned}
P_1 &= \frac{R}{B} \cdot P_{tx}(|ac|) + \frac{R}{B} \cdot P_{rx} + 2(1 - \frac{R}{B}) \cdot P_{id} + P_s \\
P_2 &= \frac{R}{B} \cdot (P_{tx}(|ab|) + P_{tx}(|bc|)) + \frac{2R}{B} \cdot P_{rx} + (3 - \frac{4R}{B}) \cdot P_{id}
\end{aligned}
$$

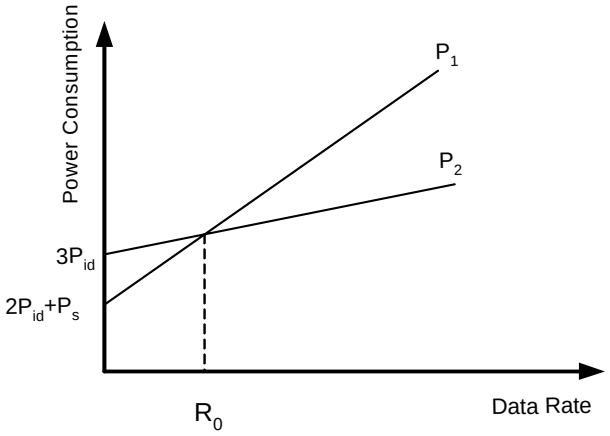


Figure 5.2: Average power consumption vs. data rate

Each term in P_1 or P_2 is the product of power consumption in a radio state and the fraction of time the radio operates in that state. For example, in the first term of P_2, $P_{tx}(|ab|) + P_{tx}(|bc|)$ is the transmission power of nodes a and b, and $\frac{R}{B}$ is the fraction of time nodes a and b operate in transmission state. Similarly, the second term of P_2 represents the contribution of the reception power of nodes b and c. In the third term of P_2, P_{id} is the idle power, and $3 - \frac{4R}{B}$ is the sum of the fractions of time when nodes a, b and c stay in the idle state. Specifically, node a is idle $1 - \frac{R}{B}$ of the time because it becomes idle when not transmitting to b, node b is idle $1 - \frac{2R}{B}$ of the time because it becomes idle only when neither transmitting to c nor receiving from a, and node c is idle $1 - \frac{R}{B}$ of the time because it becomes idle when not receiving from b.

For the given radio parameters and node locations, all symbols except R are constant in the expressions of P_1 and P_2. We plot P_1 and P_2 in Fig. 5.2 under a possible

setting of radio parameters and node locations. We can see that $P_1 > P_2$ when the data rate exceeds a threshold R_0 given by:

$$R_0 = \frac{P_{id} - P_s}{P_{tx}(|ac|) - P_{tx}(|bc|) - P_{tx}(|ab|) + 2P_{id} - P_{rx}} \tag{5.1}$$

To get a concrete estimation on R_0, we now apply the parameters of the CC1000 radio on Mica2 motes [15] to (5.1). For a $433MHz$ CC1000 radio, the bandwidth is 38.4 Kbps. There are a total of 31 transmission power levels, each of which leads to a different transmission range[1]. Suppose $P_{tx}(|ac|)$ is equal to the maximum transmission power 80.1 mW. $P_{tx}(|ab|)$ and $P_{tx}(|bc|)$ are equal to the medium transmission power 24.6 mW. P_{id}, P_{rx}, and P_s are 24 mW, 24 mW and 6 μW, respectively. Using this information, it can be calculated that relaying through node b is more power efficient when the data rate is above 16.8 Kbps.

This example leads to the following observations on the power-efficient network configuration: 1) When network workload is low, energy consumption of a network is dominated by the idle state of the radio. In such a case, scheduling nodes to sleep saves the most energy. It is therefore wise to use long communication range between any two nodes in order to allow any nodes that would otherwise be used as relays to sleep. 2) When network workload is high, the transmission energy dominates the total energy consumption of a network. Since transmission power increases quickly with distance, using shorter communication ranges that are relayed through multiple nodes saves more energy.

[1]The actual transmission range of a radio also depends on environment and antenna.

5.2 Problem Definition

We define our problem formally in this section. We first define several simple concepts. A node can either be *active* or *sleeping*. For any given time instance, an active node works in one of the following states: *transmitting*, *receiving* or *idle*. The total energy consumption of an active node is equal to the sum of the energy consumption in all states. The sleeping power consumption is orders of magnitude lower than active power consumption [15, 11]. In this thesis, we only consider the total active energy consumption in a network. We define the following notation.

1. The maximal and minimal transmission power of each node is denoted by P_{tx}^{max} and P_{tx}^{min}, respectively. $P_{tx}(u, v)$ is the minimum power needed for successful transmission from node u to node v, $P_{tx}^{min} \leq P_{tx}(u, v) \leq P_{tx}^{max}$.

2. $G(V, E)$ represents a wireless network. V includes all nodes in the network and E is defined as $E = \{(u, v) | (u, v \in V) \ \wedge (P_{tx}(u, v) \leq P_{tx}^{max})\}$.

3. P_{rx} and P_{id} represent the power consumption of a node in receiving and idle state, respectively.

4. $S = \{s_i\}$ and $T = \{t_j\}$ represent a set of source and sink nodes, respectively. $I = \{(s_i, t_j, r_{i,j}) \mid s_i \in S, \ t_j \in T\}$ represents a set of traffic demands where source s_i sends data to sink t_j at rate $r_{i,j}$.

In many sensor network applications, e.g., periodic data collection [42, 72], a source is aware of its data rate. Alternatively, a source may estimate its average data rate online. We assume that the total workload in the network is lower than the network capacity, which is in turn much lower than nodes' bandwidth in multi-hop wireless

networks due to network contention and interference. We note that this assumption holds in many sensor network applications with low data rates. For instance, in the WSN deployed at Great Duck Island for habitat monitoring [59], each mote only sends its sensor data to the base station every 20 minutes. Many other representative applications (e.g., precision agriculture and cargo tracking) also have low data rate.

The Minimum Power Configuration (MPC) problem can be stated as follows. Given a network and a set of traffic demands, find a subnet that satisfies the traffic demands with minimum energy consumption. We note that minimizing the total energy consumption of a network is equivalent to minimizing the average power consumption of all nodes. We first consider the average power consumption of a node, assuming the data path $f(s_i, t_j)$ from source s_i to sink t_j is known. To simplify the formulation, we introduce a virtual source node s_* and virtual sink node t_* to the network. s_* sends data to each source s_i at the rate of $r_{i,j}$. Each sink t_j sends data to t_* at a rate of $r_{i,j}$. Note that the additional power consumption due to the introduction of s_* and t_* is constant for a given set of traffic demands. Now the average power consumption, $P(u)$, of any active node u (excluding s_* and t_*), can be computed as the weighted average of power consumption in transmitting, receiving, and idle states:

$$\begin{aligned} P(u) &= \left(1 - 2 \sum_{(u,v)\in f(s_i,t_j)} r_{i,j}\right) \cdot P_{id} + \sum_{(u,v)\in f(s_i,t_j)} r_{i,j} \cdot (P_{tx}(u,v) + P_{rx}) \\ &= P_{id} + \sum_{(u,v)\in f(s_i,t_j)} r_{i,j} \cdot (P_{tx}(u,v) + P_{rx} - 2P_{id}) \end{aligned}$$

where $(u, v) \in f(s_i, t_j)$ represents that there exists a node v such that edge (u, v) is on the path $f(s_i, t_j)$. Based on the average power consumption of a node defined by the above equation, the MPC problem can be defined as follows.

Definition 1 (MPC problem)**.** *Given a network $G(V, E)$ and a set of traffic demands I, find a subgraph $G'(V', E')$ $(V' \subseteq V, E' \subseteq E)$ and a path $f(s_i, t_j)$ within G' for each traffic demand $(s_i, t_j, r_{i,j}) \in I$, such that the average power consumption $P(G')$ is minimal, where*

$$P(G') = \sum_{u \in V'} P(u) = |V'|z + \sum_{u \in V'} \sum_{(u,v) \in f(s_i,t_j)} r_{i,j} \cdot C_{u,v} \tag{5.2}$$

and $C_{u,v}$ and z are defined as follows:

$$C_{u,v} = P_{tx}(u, v) + P_{rx} - 2P_{id} \tag{5.3}$$

$$z = P_{id} \tag{5.4}$$

From the above formulation, we can see that an edge (u, v) has a cost $C_{u,v}$ for each unit of the data flowing through it, and each node has a fixed cost z that is independent of workload. We assume that all the data in the same flow takes the same path, i.e., a flow is not splittable. Under such a consumption, one can show that network path $f(s_i, t_j)$ is the shortest path in graph G' with edge weight $C_{u,v}$. (5.2) can then be reformulated as follows:

$$P(G') = |V'|z + \sum_{(s_i,t_j,r_{i,j})\in I} r_{i,j} \cdot P(s_i, t_j) \tag{5.5}$$

where $P(s_i, t_j)$ represents the shortest path in $G'(V', E')$ with edge weight $C_{u,v}$. According to (5.5), the total power cost is equal to the sum of the costs along the shortest path of each traffic demand and the total nodal costs.

When $\forall(u, v) \in E, \quad P_{tx}(u, v) + P_{rx} = 2P_{id}$, the cost function of the MPC problem becomes $|V'|z$. When there is only one sink t in the network, the problem is equivalent to finding the minimum-weight Steiner tree in $G(V, E)$ with uniform edge weight z to connect the nodes in $S \cup \{t\}$. This special case of the minimum-weight Steiner tree problem is NP-hard [23]. As a result, a natural reduction from this problem can show that the MPC problem is also NP-hard.

Although polynomial solutions for the general MPC problem are unlikely to exist, the following non-trivial special cases of the MPC problem can be solved optimally in polynomial time.

1. When $S \cup T = V$, i.e., every node in the network is either source or sink and hence needs to remain active. Thus the first term in (5.2) becomes $|V|z$ which is constant for a given network. In such a case, the solution is equivalent to finding the shortest paths with edge weight $r_{i,j} \cdot C_{i,j}$ connecting all sources to their sinks and hence can be solved in polynomial time.

2. When $P_{id} = 0$, as is similar to the first case, the MPC problem can be solved optimally by shortest-path algorithms.

In the problem formulation, we assume that all data sources are known offline. This assumption may not be practical in many sensor network applications where data sources are usually triggered by asynchronous events (e.g., an object passing by) or a query submitted by users. That is, the data sources in many scenarios arrive in an online fashion. In Section 6.2, we discuss both offline and oneline approximate algorithms for the MPC problem.

In our problem definition, the energy consumption of packet retransmissions on lossy communication links is ignored. Recent empirical studies show that lossy communication links are common in real sensor networks [67, 81]. In such a case, the communication quality between two nodes can be quantified by packet reception ratio (PRR) [85]. In this chapter, we assume an automatic repeat request (ARQ) mechanism is used to deal with lossy links. A node with ARQ keeps retransmitting a packet until the packet is successfully acknowledged by the receiver or the preset maximum number of retransmissions is reached. To reflect the additional energy cost caused by retransmissions, the cost function defined in (5.2) can be revised as follows. Let $PRR(u, v, P_{tx})$ represent the PRR when u communicates with v using transmission power P_{tx}. Note that $PRR(u, v, P_{tx})$ depends on the quality of both forward and reverse links between u and v when an ARQ is used[2]. The expected transmission power cost when u communicates with v with P_{tx} on the lossy links can be estimated as $P_{tx}/PRR(u, v, P_{tx})$. Hence the most efficient transmission power that should be used by u to communicate with v is determined as follows:

$$P_{tx}(u, v) = arg\min \frac{P_{tx}}{PRR(u, v, P_{tx})}, \quad P_{tx}^{min} \leq P_{tx} \leq P_{tx}^{max} \tag{5.6}$$

[2]Acknowledgment can be transmitted at a relatively high power level to reduce the number of retransmissions.

We redefine $P_{tx}(u, v)$ in (5.3) of our problem formulation according to (5.6) when the communication links are lossy.

5.3 Centralized Approximation Algorithms

We investigate approximate algorithms for the general MPC problem in the this section. We first focus on the scenario where there is only one sink in the network in this section. Each source s_i ($s_i \in S$) sends data to sink t at a data rate of r_i. We discuss the extension of some of our results to the scenario of multiple sinks in Section 5.3.3.

5.3.1 Matching based Algorithm

When there is only one sink and data flows are not splittable, the MPC problem has the same formulation as the *cost-distance* network design problem [44]. Meyerson et. al proposed a randomized approximation scheme [44] that has a best known approximation ratio of $O(\lg k)$ with k being the number of sources . We briefly review the algorithm and propose an optimization that considerably improves the practical performance of the algorithm.

The Meyerson algorithm takes a graph $G(V, E)$ and outputs a subgraph $G'(V', E')$ that contains the paths from all sources to the sink.

The time complexity of the above algorithm is $O(k^2(m + n \lg n))$ (where k, m and n represent the number of sources, total number of edges and nodes in G respectively). As shown in [44], the algorithm terminates after at most $O(\lg k)$ iterations

Input: $G(V, E)$, set $W = S \cup \{t\}$ and traffic demands I
Output: $G^{'}(V', E')$

1. Create a complete graph M containing all nodes in W as follows. Each edge between two nodes in M is the shortest path between the two nodes in G under the edge cost D. For two sources s_i and s_j, $D_{u,v} = z + \frac{2r_i r_j}{r_i + r_j} C_{u,v},\ (u, v) \in E$. For a source s_i and sink t, $D_{u,v} = z + r_i C_{u,v},\ (u, v) \in E$.

2. Find a matching of graph M that has at most half the cost of the minimum perfect matching, and has at most half of the number of total nodes.

3. The nodes and edges of G defining each matched edge of M are added into G'. For each matched edge (s_i, s_j) in M, choose s_i to be the center with probability $r_i/(r_i + r_j)$, otherwise s_j will be the center. Change the data rate of the center as $r_i + r_j$.

4. Each non-center node in a matched edge of M is removed from W. Stop if S contains only the sink. Otherwise go to step 1 with the updated W.

Figure 5.3: Matching based algorithm (MBA) for MPC problem

and the expected cost introduced by the newly added edges in each iteration is at most constant times of the cost of the optimal solution. Hence the approximation ratio of the algorithm must be $O(\lg k)$. We refer to this algorithm as *matching based approximation (MBA)* in the rest of this chapter.

We note that edge of G can lie on the matched edges of M in multiple iterations at step 3 of MBA. However, the fixed cost of each edge z is only counted once in the total cost of the solution (see (5.2)). This observation can lead to the following optimization to MBA. After the matching of M is found in step 2, we redefine the cost of each matched edge of G as $D_{u,v} = \frac{2r_i r_j}{r_i + r_j} C_{u,v}$. That is, the fixed cost of each edge z is removed if the edge is matched. The intuition behind this consideration is that the matchings in following iterations will tend to reuse the edges of G that have been previously matched due to the cost reduction on these edges. Consequently, the total cost of the solution may be reduced by more path sharing. We refer to the MBA with this optimization as MBA-opt. Although MBA-opt does not improve the approximation ratio of MBA, we show in section 5.3.5 that it can result in considerable improvement on the practical performance.

Although MBA and MBA-opt have a good performance bound, they suffer from the following drawbacks. First, efficient distributed implementations of them are difficult to realize in large-scale sensor networks. In order to find the matching of the network graph (step 2 of MBA) in a distributed environment, complex coordination between nodes is needed [66]. Secondly, MBA and MBA-opt are not applicable to the online scenario in which sources arrive dynamically because finding the matching of the network requires the knowledge of all data sources. Finally, MBA and MBA-opt only work for the scenario in which there is a single sink in the network. Because of

these drawbacks, we are forced to design other approximate algorithms that are more suitable to distributed and online implementations.

5.3.2 Shortest-path Tree Heuristic (STH)

In this section, we discuss an approximation algorithm called the shortest-path tree heuristic (STH). The idea behind this heuristic is to balance the flow dependent cost ($r_{i,j} \cdot C_{u,v}$) and the fixed nodal cost (z) of a graph using a combined cost metric. For convenience, we define a set of weight functions for edge (u, v):

$$g_i(u, v) = r_i \cdot C_{u,v} + z \tag{5.7}$$

Each weight function $g_i(u, v)$ defines a cost for edge (u, v) when the data flow from s_i travels through that edge. The pseudo-code for STH is shown in Fig. 5.4. At each iteration, STH simply finds the shortest path from one of the sources to the sink according to weight function (5.7). The output of STH is the union of all shortest paths found. Note that, the cost of an edge needs to be updated during each iteration (step 2.a) since the cost depends on the data rate of the current source (according to (5.7)).

Fig. 5.5 shows an example of the STH algorithm. The shortest paths from s_1, s_2 to t are highlighted in black. Fig. 5.5(a) shows an initial network without any flows. Fig. 5.5 (b) and (c) show two iterations of STH. In each iteration, $G(V, E)$ is weighted according to g_i and the shortest path from s_i to t is found. The output of STH is the graph composed of all of the shortest paths found. According to (5.2), the average power cost (excluding the cost of the sink) can be calculated to be 9.4.

Input: $G(V, E)$, source set S, sink t and traffic demands I
Output: $G^{'}(V', E')$

1. Initialize $G^{'}(V', E')$ to be empty.
2. **foreach** s_i
   (a) Assign edge weights for $G(V, E)$ according to g_i.
   (b) Find the shortest path connecting s_i to t.
   (c) Add the shortest path found to $G^{'}$.
3. **end**

Figure 5.4: Shortest-path Tree Heuristic (STH)

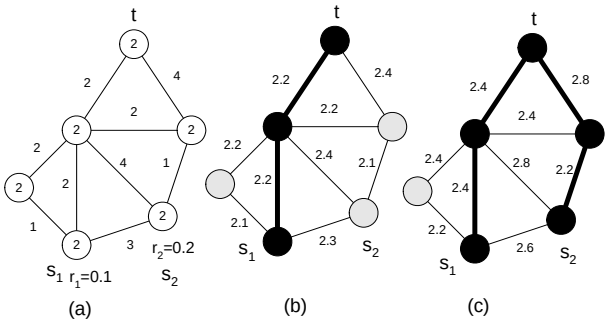


Figure 5.5: (a) Initial network with edge weight $C_{u,v}$ and node weight $z = 2$ (shown on each node). (b) edge weights are defined by $r_1 \cdot C_{u,v} + z$. (c) edge weights are defined by $r_2 \cdot C_{u,v} + z$.

Step 4 of the STH algorithm can be implemented using Dijkstra's shortest-path algorithm. The complexity of STH is $O(|S||E|\ \lg|V|)$. It can be seen that STH outputs the optimal solution for the two polynomial-time special cases of MPC problem discussed in Section 5.2.

Before we investigate the performance bound of STH for the general MPC problem, we define the following notation. We define a set of weight functions w_i for edge (u, v) as follows:

$$w_i(u, v) = r_i \cdot C_{u,v} \tag{5.8}$$

$w_i(u, v)$ represents the cost of edge (u, v) when the data flow from s_i travels through (u, v). Let $P^x_G(u, v)$ represent the cost of the shortest path between node u and v in graph G under the weight function x. Then (5.2) can be reformulated as follows:

$$P(G^{'}) = \sum_i P^{w_i}_{G^{'}}(s_i, t) + |V^{'}|z \tag{5.9}$$

We have the following theorem regarding the performance of STH.

Theorem 12. *The approximation ratio of STH is no greater than $|S|$.*

Proof. Let $P(G^{'})$ and $P(G^{'}_{min})$ represent the total cost of $G^{'}$ found by STH and the optimal solution, respectively. The total cost of the shortest paths found by STH in $G^{'}$ with weight g_i is greater than in $P(G^{'})$ because the idle power z of each node in $G^{'}$ might be counted multiple times. We have:

$$P(G^{'}) \leq \sum_{i} P^{g_i}_{G'}(s_i, t) \tag{5.10}$$

Since STH finds the shortest paths in G with weight g_i and $G^{'}_{min} \subset G$, we have:

$$\sum_{i} P^{g_i}_{G'}(s_i, t) \quad \leq \quad \sum_{i} P^{g_i}_{G^{'}_{min}}(s_i, t) \tag{5.11}$$

Consider the total cost of the shortest paths from s_i to t in $G^{'}_{min}$ with weight g_i. This cost is greater than the optimal solution $P(G^{'}_{min})$ since weight z might be counted multiple times for each node in $G^{'}_{min}$. It can be seen that z is counted at most $|S|$ times for each node (which occurs when a node lies on the paths from all the sources to the sink). Thus we have:

$$\begin{aligned} \sum_{i} P^{g_i}_{G^{'}_{min}}(s_i, t) &\leq \sum_{i} P^{w_i}_{G^{'}_{min}}(s_i, t) + |S|(|V^{'}|)z \\ &\leq |S| \left(\sum_{i} P^{w_i}_{G^{'}_{min}}(s_i, t) + (|V^{'}|)z \right) \\ &= |S| P(G^{'}_{min}) \end{aligned} \tag{5.12}$$

From (5.10) to (5.12), we have:

$$P(G^{'}) \leq |S| P(G^{'}_{min})$$

5.3.3 Incremental Shortest-path Tree Heuristic (ISTH)

In STH, the function used to weight the network is different for each source. Consequently, the shortest path from a source to the sink is not affected by whether shortest paths are already established for other sources. Intuitively, this does not seem efficient since sharing an existing path could lead to lower nodal costs. Suppose we are finding the shortest path from s_i to t and all the shortest paths from $s_j (0 < j < i)$ to t have already been found. If any edge on the existing paths is reused by the new path, the incremental cost is $r_i \cdot C_{u,v}$. This cost does not include the nodal cost z since it has been counted by the existing paths. That is, the edge weights on the existing paths should not include the nodal cost z. Based on this observation, we propose the following algorithm called *incremental shortest-path tree heuristic (ISTH)* that finds the path from each source to the sink with the minimal *incremental* cost. The pseudo-code of ISTH is depicted in Fig. 5.6. During its execution, the algorithm maintains a subgraph G' that contains the paths from the sources to the sink that have been visited so far. In each iteration, ISTH finds the remaining source node that is closest to, but not connected to the sink in G'. It then adds the shortest path from that node to the sink into G'. For convenience, we refer to the state of those nodes already in G' to be *active*. Once a node becomes active (i.e., included by G'), the cost of any edge originating from it is decreased by z to reflect the incremental cost incurred by the edge when a new flow travels through it. Formally, when ISTH finds the shortest path from source s_i to the sink, the edge cost is defined by the following function:

$$h_i(u,v) = \begin{cases} r_i \cdot C_{u,v} & \text{u is } \textit{active} \\ r_i \cdot C_{u,v} + z & \text{otherwise} \end{cases} \tag{5.13}$$

Input: $G(V,E)$, source set S, sink t and traffic demands I
Output: $G^{'}(V',E')$

1. Initialize $G^{'}(V',E')$ to be empty.
2. Label all nodes as *asleep*.
3. $W = S$.
4. **while** $W \neq \phi$
   (a) Find $s_i \in W$ that has the shortest distance in $G(V,E)$ to t with edge weight $h_i(u,v)$.
   (b) Add the shortest path from s_i to t in $G^{'}$.
   (c) Label all nodes on the path as *active*.
   (d) $W = W - s_i$.
5. **end**

Figure 5.6: Incremental Shortest-path Tree Heuristic (ISTH)

Fig. 5.7 shows the second iteration of an example of ISTH in which the shortest path from s_1 to t has been found. The first iteration of the example is the same as that of STH shown in Fig. 5.5(b). The total weights on the shortest path from s_1 to t in Fig. 5.7 are smaller than those in Fig. 5.5(c) since the nodal cost z is not included. Consequently, different from the case of STH where two paths must always be disjoint as shown in Fig. 5.5(c), the shortest path from s_2 to t shares an edge with the existing path. The total number of nodes used is therefore decreased resulting in less idle energy consumption. According to (5.2), the average power cost in this

example (excluding the cost of the sink) can be calculated to be 7.6. This value is smaller than the one obtained for the solution to STH. It can easily be seen that this solution is optimal for this example.

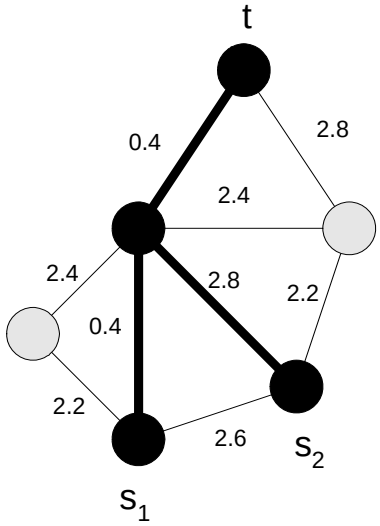


Figure 5.7: The shortest path from s_2 to t shares a edge with the existing shortest path from s_1 to t.

We now prove that the approximation ratio of ISTH is at least as good as that of STH.

Theorem 13. *The approximation ratio of ISTH is no greater than* $|S|$.

Proof. Let $P(G^{'})$ and $P(G^{'}_{min})$ represent the total cost of $G^{'}$ found by ISTH and the optimal solution, respectively. $P(G^{'})$ equals the sum of the costs of all shortest paths found by ISTH. We have:

$$P(G^{'}) \quad = \quad \sum_{i} P^{h_i}_{G^{'}}(s_i, t)$$

According to (5.13) and (5.7), $h_i \leq g_i$. Hence the incremental cost found by ISTH at each iteration must be no greater than that found by STH. We have:

$$\sum_i P_{G'}^{h_i}(s_i, t) \quad \leq \quad \sum_i P_{G'}^{g_i}(s_i, t) \tag{5.14}$$

According to (5.14), (5.11) and (5.12), we have:

$$P(G^{'}) \leq |S| P(G^{'}_{min})$$

□

As we mentioned earlier, when $\forall (u, v) \in E, C_{u,v} = 0$, the MPC problem is equivalent to finding the minimum-weight Steiner tree connecting all the sources and the sink in G with uniform edge weight z. In ISTH, once a shortest path is found, the weights on the path become zero. Finding a subsequent shortest path from a source to the sink is therefore equivalent to finding the shortest path to any node on the existing path. In such a case, ISTH is equivalent to the minimum-weight Steiner tree heuristic with an approximation ratio of 2 [24]. This result suggests that ISTH yields good performance when the idle energy dominates the total energy consumption of a network. Such a situation occurs when network workload or transmission/reception power is low. Similar to STH, ISTH finds the optimal solution for the two polynomial-time special cases of the MPC problem.

At each iteration of ISTH (see Fig. 5.6), the data source closest to the sink is chosen for processing from among all of the remaining sources. Since this operation requires

knowing about every source in the network, it can not be implemented online. A straightforward modification to handle online sources is to process one new source at each iteration of the algorithm. Although this modification likely results in average performance degradation, the approximation ratio of ISTH, $|S|$ (where S is the set of sources), remains unchanged. This holds true because the proof of Theorem 13 does not require any particular sequence for the processing of sources. This property allows ISTH to preserve its performance bound in online scenarios.

We have been focusing on the scenario involving a single sink in this section. As STH and ISTH are based on pairwise, shortest-path heuristics, they can easily be extended to a scenario containing multiple sinks. It can be shown that the approximation ratio of both algorithms still holds using similar proofs.

5.3.4 Constant-ratio Approximation Algorithm

Although the STH and ISTH algorithms described previously do find the optimal solution for the two polynomial-time special cases of the MPC problem, their known approximation ratio is equal to the number of source nodes in the network for the general MPC problem, causing them to not scale so well when the number of sources becomes large. In this section, we seek an algorithm with a constant approximation ratio. We show in the following theorem that a minimum-weight Steiner tree algorithm will lead to a constant approximation ratio for MPC problem, when the ratio of maximal transmission power to idle power is bounded.

Theorem 14. *Let H be the best approximation algorithm to the minimum-weight Steiner tree problem that has an approximation ratio β. If $\forall(u,v) \in E, C_{u,v} \leq \alpha z$, the*

solution by executing H in G with the uniform edge weight z has an approximation ratio $(1+\alpha)\beta$ to the optimal solution of MPC problem.

Proof. Suppose $G'_{min}(V'_{min}, E'_{min})$ and $G'(V', E')$ are the optimal solutions to the minimum-weight Steiner tree problem and the solution of algorithm H, respectively. Since H has an approximation ratio of β and all edges have the same weight z, we have:

$$|V'| - 1 = |E'| < \beta |E'_{min}| = \beta(|V'_{min}| - 1) \tag{5.15}$$

Let $P(G')$ and $P(G'_{min})$ represent the cost of G' and $P(G'_{min})$ in MPC problem. We ignore weight z for the constant sink node in both $P(G')$ and $P(G'_{min})$. Doing so does not affect the quality of G' or the optimality of G'_{min}. We have:

$$\begin{aligned} P(G') &= \sum_i \sum_{(u,v)\in f(s_i,t)} r_i \cdot C_{u,v} + (|V'| - 1)z \\ &\leq \sum_{(u,v)\in E'} \left(C_{u,v} \cdot \sum_i r_i \right) + (|V'| - 1)z \end{aligned} \tag{5.16}$$

where $f(s_i, t)$ represents the shortest path with edge weight $C_{u,v}$ from s_i to t. Based on the assumption that the total workload in the network is lower than network capacity, $\sum_i r_i \leq 1$. We have:

$$\begin{aligned} P(G^{'}) &\leq \sum_{(u,v)\in E^{'}} C_{u,v} + (|V^{'}| - 1)z \\ &\leq \sum_{(u,v)\in E^{'}} \alpha z + (|V^{'}| - 1)z \\ &= |E^{'}|\alpha z + (|V^{'}| - 1)z \\ &= (|V^{'}| - 1)(1 + \alpha)z \end{aligned} \tag{5.17}$$

According to (5.15) and (5.17), we have:

$$\begin{aligned} P(G^{'}) &< \beta(|V^{'}_{min}| - 1)(1 + \alpha)z \\ &< (1 + \alpha)\beta \left((|V^{'}_{min}| - 1)z + \sum_{i} P^{w_i}_{G^{'}_{min}}(s_i, t) \right) \\ &= (1 + \alpha)\beta P(G^{'}_{min}) \end{aligned}$$

□

Theorem 14 shows that the Steiner tree based algorithm performs better when the ratio of communication power to idle power, α, is low. The intuition behind this result is that, the algorithm only minimizes the idle energy and ignores the transmission/reception energy of the radio, and hence results in more energy reduction when the idle energy constitutes a bigger portion of the total energy consumption, i.e., α is low. Therefore, Theorem 14 indicates that the Steiner tree based algorithm is particularly suitable for radios with high idle power. Theorem 14 also shows that the performance of the algorithm is dependent on β - the best approximation ratio of minimum Steiner tree algorithms. Approximate algorithms of the minimum Steiner

tree problem have been studied extensively [49]. The best known approximation ratio is about 1.5 [49]. According to the measurements of the CC1000 radio on Mica2 motes [15], $\alpha \approx 2.3$. The approximation ratio of the scheme discussed in this section is therefore about 5 on the CC1000 radio.

Input: $G(V, E)$, source set S, sink t and traffic demands I
Output: $G^{'}(V', E')$

1. Set the weight of every edge in $G(V, E)$ to z.
2. $V' = t$
3. $W = S$.
4. **while** $W \neq \phi$
   (a) Find $s_i \in W$ that has the shortest distance to $G^{'}$ with edge weight z.
   (b) Add the shortest path found in the previous step to $G^{'}$.
   (c) $W = W - s_i$.
5. **end**

Figure 5.8: The Gilbert minimum Steiner tree algorithm

Fig. 5.8 shows a simple minimum Steiner algorithm proposed by Gilbert et al. [24]. At step 4(a), the shortest path from a source s_i to G' is the shortest path among the shortest paths from s_i to all nodes in G' . The algorithm has an approximation ratio of 2 [24]. In Section 5.4.2, we will discuss the design of a distributed protocol called MASP based on the Gilbert Steiner algorithm. The rationale of employing this algorithm instead of more complex algorithms with better approximation ratios is that this algorithm admits an efficient distributed implementation.

The Gilbert algorithm (see Fig. 5.8) can be extended as follows to the scenario where sources arrive online. At step 4(a) of each iteration, a shortest path is found to connect the new source to the subgraph composed of the sink and existing sources before being added to the existing subgraph. The output is the subgraph composed of all sources and their respective paths found. This scheme has been shown to have an online approximation ratio of $\lg |S|$ to the minimum Steiner tree problem (where S is the set of nodes to be connected) [28]. According to Theorem 14, the approximate ratio of this online algorithm for MPC problem is $(1 + \alpha)\ \lg |S|$.

5.3.5 Performance Evaluation

In this subsection, we evaluate the average performance of the centralized approximate algorithms we presented in previous subsections through simulations. As discussed in Section 5.3.3, STH likely performs worse than ISTH and hence is not evaluated in this section.

We implement MBA, MBA-opt, ISTH, and the Gillbert Steiner tree algorithm (referred to as Steiner hereafter) in a network simulator. To evaluate the effectiveness of other energy conservation approaches to our problem, we also implemented two baseline algorithms called Transmission-power Minimum Spanning Tree (TMST) and Transmission-power Shortest Path Tree (TSPT). TMST finds the minimum spanning tree of the network where each edge is weighted by the minimum transmission power of that edge. We choose TMST as a baseline algorithm for performance comparison since distributed MST has been shown to be an effective topology control algorithm [38]. Similarly, TSPT finds the shortest path tree of the network when weighted by

Tx Power (dBm)	Radio Range(m)	Current Consumption (mA)
-20	5	8.6
-10	18	10.1
0	50	16.8
5	68	25.4

Table 5.1: Radio transmission parameters

transmission power, and this technique has been previously proposed as an efficient power-aware routing scheme [56].

We use the parameters of the CC1000 radio on Mica2 Motes in the simulation. There is no packet loss in the simulation environment. The node bandwidth is 40 Kbps. In the simulation, only the nodes that lie on the communication paths between sources and the sink remain active (i.e., the state of their radios is either transmitting, receiving or idle). All non-communicating nodes run in the sleeping state. The power consumption of the radio in receiving, idle, and sleeping states is 21 mw, 21 mw and 6 μw, respectively [15]. The actual radio range of the CC1000 on Mica2 motes varies depending on environmental factors and transmitting power. We set the parameters of the radio range and transmitting power according to the empirical measurements presented in [2], which are listed in Tab 5.1. When a node communicates with a neighbor, it always uses the minimum radio range that can reach that neighbor. At the beginning of the simulation, a communication path from each source to the sink is found. The nodes on the communication paths remain active and all other nodes are put to sleep. The simulation time for each algorithm is 1000 seconds. 200 nodes are randomly distributed in a $500m \times 500m$ region. The results in this section are the average of 10 different network topologies.

Fig. 5.9 shows the total energy consumption of the network when the number of flows varies from 1 to 100. The data rate of each flow is 0.2 Kbps. We can see that MBA-opt, ISTH and Steiner significantly outperform the other algorithms. The good performance of Steiner and MBA-opt are expected because of their good approximation ratios. Interestingly, ISTH yields a similar performance as MBA-opt and Steiner although ISTH's known approximation ratio is worse than them. This result is due to the following facts. First, the performance bound of ISTH is derived under the worst-case scenarios, which do not exhibit in the simulation. Second, although the aggregate data rate of all flows in the simulation is up to half of network bandwidth, the data rate of each individual flow is very low. As a result, the active nodes on data routes remain idle in most of the time. In such a case, ISTH minimizes the number of active nodes, resulting similar behavior as Steiner (i.e., g_i in (5.13) is close to zero). Fig. 5.9 also shows the effectiveness of our optimization to the MBA algorithm, as discussed in Section 5.3.1. TSPT and MST result in considerably higher energy consumption than the above algorithms since they only consider transmission power and ignore idle power.

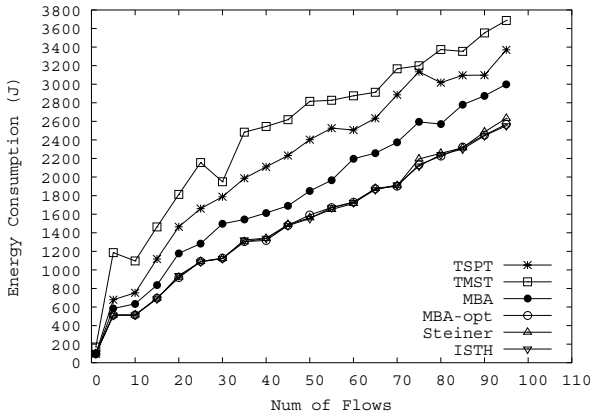


Figure 5.9: Energy consumption vs. number of flows.

The results in this section show that the average performance of ISTH and Steiner is similar to that of MBA-opt. As both ISTH and Steiner are based on the shortest-path algorithm, they have a more efficient distributed implementation than MBA-opt. We now turn our attention to the distributed implementation of ISTH and Steiner in the next section.

5.4 Distributed Protocols

In this section, we present the design and implementation of two distributed routing protocols, Minimum Power Configuration Protocol (MPCP) and Minimum Active Subnet Protocol (MASP). These protocols are based on centralized algorithms ISTH and Steiner presented in Section 5.3, respectively. We focus on a "many to one" routing scenario in our discussion since it is the most common communication paradigm in sensor networks. MPCP and MASP can be easily extended to support more general routing scenarios.

5.4.1 Minimum Power Configuration Protocol

In this section, we present the design of the minimum power configuration protocol (MPCP). MPCP finds the power-efficient routes for communicating nodes in a network based on the distributed implementation of the ISTH algorithm with online extentions discussed in Section 5.3.3.

Shortest-path based routing mechanisms have been extensively studied. We adopt the Destination Sequenced Distance Vector Routing (DSDV) protocol [46] as our

implementation framework. DSDV is based on the distributed implementation of the Bellman-Ford shortest path algorithm [5]. A node in DSDV advertises its current routing cost to the sink by broadcasting *route update* messages. A node sets the neighbor that has the minimum cost to the sink as its parent and rebroadcasts its updated cost if necessary. DSDV can avoid the formation of routing loops by using sink-based sequence numbers for route updates. The routing cost of a node in DSDV is its hop count to the sink. The routing cost of a node in MPCP, however, depends on the operational state of the node (active or power saving) as well as the data rates of the flows that travel through the node. We now discuss in detail the core components of MPCP.

Node States and Routing Table

In our design, a node operates in either *active* or *power saving* mode. A node in power saving mode remains asleep in most of the time and only periodically wakes up. This simple sleep schedule is similar to several existing power saving schemes such as SMAC [77]. Initially, all nodes operate in power saving mode. When a source node starts sending data to the sink, a power-efficient routing path from the source to the sink is found by the distributed ISTH algorithm. All nodes on the routing path are activated to relay data from the source to the sink. All the other nodes remain in the power saving mode to reduce energy consumption. Similarly, an active node switches to the power saving mode if all the data flows traveling through it disappear.

Each node in the network maintains a routing table that contains the routing entries and status of neighbors. Since the routing cost to the sink varies with the data rate of the source, we need to store an entry for each data rate in the network.

data rate packets/s	next hop	cost	seq
2.1	5	28.9	8
1	7	8.9	6
0.5	15	18.3	8
0.1	30	8.2	12

Table 5.2: A routing table in MPCP.

Specifically, an entry in the routing table of node u includes the following fields: $< r_i,\ next\_hop,\ cost,\ seq >$ where r_i is the data rate of source s_i, *next_hop* is the neighbor node with the minimum cost to the sink, *cost* is the cost of node u to the sink through *next_hop*, *seq* is a sequence number originated by the sink. Tab. 5.2 shows a routing table of an active node.

One simple method of obtaining source rates is to let each source flood the network with its rate information before finding a route to the sink. This approach incurs too much overhead, however, when a network is composed of many nodes. To reduce the overhead, only the data rates with significant difference are kept in the routing table. When a new source node starts sending data, it chooses the next hop node from a routing table entry that has the data rate closest to its own data rate. The new data rate will then be propagated to other nodes if it is significantly different from the ones stored in their table.

Route Updates

According to cost function (5.13), the routing cost from a node to its neighbors in MPCP depends on data rate and the change of the node's state (active or power saving). As a result, a new round of route updates will be triggered by any of the

following events: (1) a link is broken; (2) the data rate of an existing flow changes; and (3) a data flow is started or completed.

A node detects a broken link when multiple transmissions fail. The process of route updates caused by a broken link is similar to DSDV. A node advertise its routing information by broadcasting a route update packet to its neighbors. After receiving an update from a neighbor, a node calculates its new cost to the sink at each data rate specified in the update, and updates its routing table. This new cost is equal to the sum of the link cost to the neighbor (defined by (5.13)) and the cost of the neighbor included in the update. A node sets a timeout after the arrival of the first route update in this round to wait for more updates from other neighbors. If there exist entries in the routing table that have a cost reduction above a threshold after the timeout, the node broadcasts a route update packet containing these entries to advertise its updated routing information.

We now discuss in detail the route updates caused by the change of data rate and start/completion of a data flow. When a source node changes its data rate to a value that differs significantly from the data rates stored in the routing table, the source node notifies the sink by including the new rate in its data packets. Once the sink sees the new rate, it broadcasts a route update with a new sequence number to the network. The routing tables of nodes are updated when the route update is broadcast throughout the network. Consequently, the source with the new data rate may choose a better route due to updated routing information. When the workload of the network is dynamic, multiple rounds of route updates may be initiated at the same time resulting in high network contention. To reduce the overhead of route updates in such a case, the sink can include several default data rates in its initial route updates based on the estimation of source rates. From then on, a new round

of route updates is initiated only when the data rate of a flow changes to a value significantly different from the default ones.

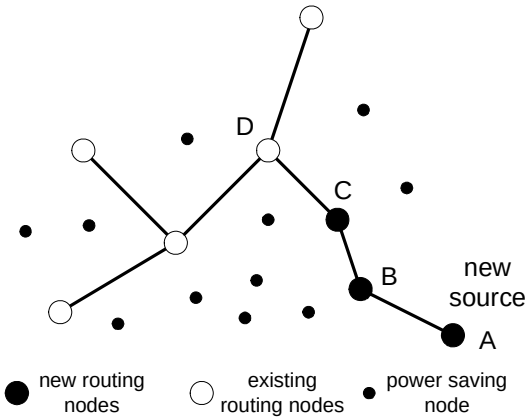


Figure 5.10: The junction node C will initiate a round of route update due to the arrival of new source A.

Route updates may also be triggered when a new data flow appears. If the new flow has a data rate significantly different from the ones stored in the routing table, a round of route updates is initiated as discussed earlier. In addition, the appearance of a new flow may activate a node previously running in power saving mode and reduce the cost of the node to its neighbors (see (5.13)). As shown in Fig. 5.10, a new data flow from source node A activates nodes A, B and C before it meets the existing routing path at a junction node D (D may be the sink node). Nodes A, B and C then lower their routing costs after being activated. In such a case, to reduce the number of route updates, only the node preceding the junction node initiates the route update since it has the minimum cost to the sink among all nodes on the new path. In Fig. 5.10, node C will broadcast a route update with a new sequence number and reduced routing costs in order to initiate a round of route updates. Nodes B, A and other nodes that have reduced routing costs to the sink participate in the route update process that has been initiated by C. Note that the route updates initiated in

this way only involve a subset of nodes in the network since many nodes (e.g., those closer to the sink) will not participate in the route update process due to no reduction in their routing costs.

Similar to the appearance of a new flow, the disappearance of an existing flow may also cause route updates. In such a case, the nodes on the existing routing path switch to the power saving mode after some timeout, resulting in higher routing costs (see (5.13)). Again, the node preceding the junction node initiates the route update process by advertising the new routing costs.

Link Estimation

In real wireless sensor networks, a routing protocol often suffers from dynamic and lossy communication links. Empirical study shows that the reliability of routing protocols can be significantly improved by only keeping "good" neighbors, e.g., those with high packet perception ratios (PRR), in neighborhood tables [67]. A simple way of obtaining the PRR of a link is by profiling the link characteristics off-line. Alternatively, the PRR can be obtained from on-line link estimators [67, 12]. For example, nodes can broadcast periodic beacon messages and the PRR of a link to a neighbor being estimated by counting the number of messages received from that neighbor. Further discussion on this issue is beyond the scope of this thesis.

5.4.2 Minimum Active Subnet Protocol

We now present the design of the minimum active subnet protocol (MASP) that finds a Steiner tree connecting all sources in the network to the sink using the minimum

number of active nodes. The MASP is also based on DSDV and has a similar design to MPCP as both protocols are based on the shortest-path algorithm. We will now discuss the major difference between MPCP and MASP.

In MASP, a node in power saving mode incurs a routing cost of P_i (idle power)[3]. Once a data flow travels through a node, it becomes active and its routing cost reduces to zero. That is, routing among active nodes is free. As a result, when a new source arrives, finding the shortest path from that node to the sink is equivalent to finding the shortest path to any active node.

Unlike MPCP, the routing cost of a node in MASP does not depend on data rates. This independence reduces the storage overhead of the routing table at each node as well as the network bandwidth used by route updates. Each entry of a routing table in MASP contains $< next\_hop,\ cost,\ seq >$. The route updates of MASP can be triggered by either a broken link, or the start or completion of a data flow. The route updates triggered by link failures are similar to DSDV, while the updates triggered by sources are similar to MPCP. MASP is expected to generate fewer routing updates than MPCP, because the change in data rates does not affect the routing cost of MASP. In other words, MASP ignores data rates because it only minimizes idle energy. As shown in our simulation results presented in Section 6.3, MASP is only suitable for radios with high idle power.

[3]Since the routing cost is the same for all power saving nodes, one can use any positive number as the routing cost.

5.5 Experimentation

5.5.1 Simulation Environment

Low-power wireless radios used by real sensor network platforms (e.g., Berkeley motes) are known to have highly irregular communication ranges and probabilistic link characterization [81]. The simplistic assumptions on wireless radio propagation made by a network simulator may cause simulation results to differ significantly from real-world experimental results [31]. Accurate simulation to the characterization of real wireless radios with different transmission power is key to evaluating the realistic performance of our protocols. Because of this importance, we took a link layer model that was developed by USC [85] and implemented it for use with the Prowler network simulator [55]. We also added improved routing support to this model based on work done during the Rmase project [80]. Experimental data showed that the USC model can simulate highly unreliable links in the Mica2 motes [85]. In our simulations, the packet reception ratio (PRR) of each link is governed by the USC model according to the distance between the two communicating nodes and the transmission power. The MAC layer in Prowler employs a simple CSMA/CA scheme without RTS/CTS, which is similar to the B-MAC protocol [47] in TinyOS. To improve the communication reliability in the lossy simulation environment, we implemented an ARQ (Automatic Repeat Request) scheme that retransmits a packet if an acknowledgment is not received after some preset timeout. The maximum number of retransmissions before dropping a packet is 8. Prowler is a Matlab-based network simulator that employs a layered event-driven structure similar to TinyOS. Using such a simulator allows us to easily implement new network modules (such as the link model from USC) and to port our protocols to Berkeley motes in future.

5.5.2 Simulation Settings

For performance comparison, in addition to MPCP and MASP, we have implemented two baseline protocols: minimum transmission (MT) routing [67] and minimum transmission power (MTP) routing. They have similar components as MPCP except for the cost metrics. MT is shown to be more reliable than the hop-count based routing scheme when given a lossy networks [67]. A node in MT chooses the next hop node with the minimum expected number of transmissions to the sink. All communication links in the original MT protocol use the same transmission power. A link between node u and v in MT has a cost of $\frac{1}{PRR(u,v)}$. To take advantage of variable transmission power, we modified the link cost of MT to $\frac{1}{PRR(u,v,P_{tx}(u,v))}$, where $P_{tx}(u,v)$ is defined in (5.6). A node in MTP chooses the next hop node with minimum total expected transmission power to the sink. The cost of a link between u and v in MTP is equal to $\frac{P_{tx}(u,v)}{PRR(u,v,P_{tx}(u,v))}$. Except for the consideration for unreliable links, MTP is similar to the minimum power routing schemes studied in [18, 19].

In each simulation, 100 nodes are deployed in a $150m \times 150m$ region divided into 10×10 grids. A node is randomly located within each grid. Source nodes are randomly chosen. The sink is located at $(150, 75)$ to increase the hop count from some of the sources. The radio bandwidth is 40 Kbps. Power parameters of the radio are set according to the empirical measurements of the CC1000 radio on Mica2 motes [54] as follows. The CC1000 radio is capable of transmitting data at 31 power levels ranging from -20 dBm to 10 dBm. To simplify our design, we chose 10 power levels from the total 31 levels. The corresponding current consumption ranges from 3.7 mA to 21.5 mA. The receiving and idle current is 8 mA. Each simulation lasts for 400 seconds. Each source sends packets at an interval randomly chosen from $10 \sim 14$

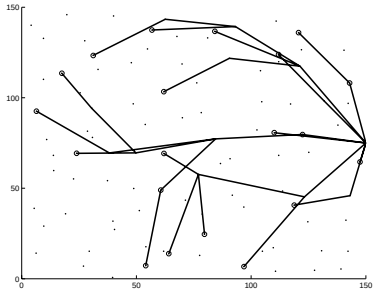
seconds, which corresponds to an average data rate between 68.5 to 96 bps. The number of sources varies from 5 to 30, which results in a total data rate of 0.4 to 2.4 Kbps at the sink. Real-world experiments show that the maximum effective multi-hop bandwidth of Mica2 motes can barely reach 6 Kbps due to channel contention and lossy wireless links [26], which conforms to our observation in simulations.

During the initialization state, a source node starts sending data at some random time after its route to the sink is found. After this initialization phase, a node that does not lie on any communication path will enter power saving mode automatically, as discussed in Section 5.4. The power saving mode has a period of 10 seconds and an active window of one second. The data packet size is 120 bytes. A routing update packet is 40 bytes. The results in this section are the average of 5 different network topologies.

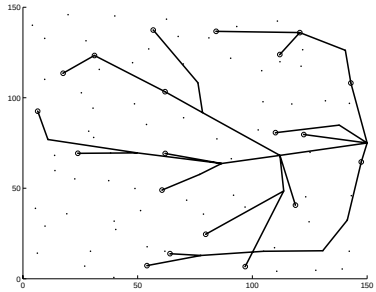
5.5.3 Performance of MPCP

In this section, we evaluate the performance of MPCP. Since the performance of MASP varies with a platform-dependent parameter α (see Section 5.3.4), we compare it with MPCP under different platform parameters in Section 5.5.4.

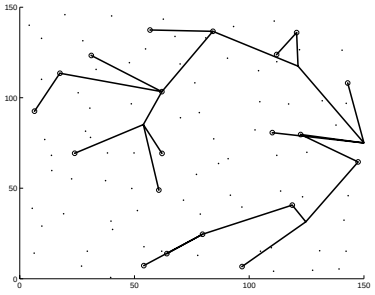
Fig. 5.11 shows the routing topologies produced by different protocols in a typical run with 20 sources. The circles in the figure represent data sources and small dots represent other nodes. We can see that the topologies produced by MT and MTP are similar and both have over 33 active nodes on the communication paths. In contrast, MPCP activates only 24 nodes, i.e., 4 more nodes besides data sources that must remain active. As the number of data sources increases, MPCP can effectively reuse



(a) MT (33 active nodes)



(b) MTP (35 active nodes)



(c) MPCP (24 active nodes)

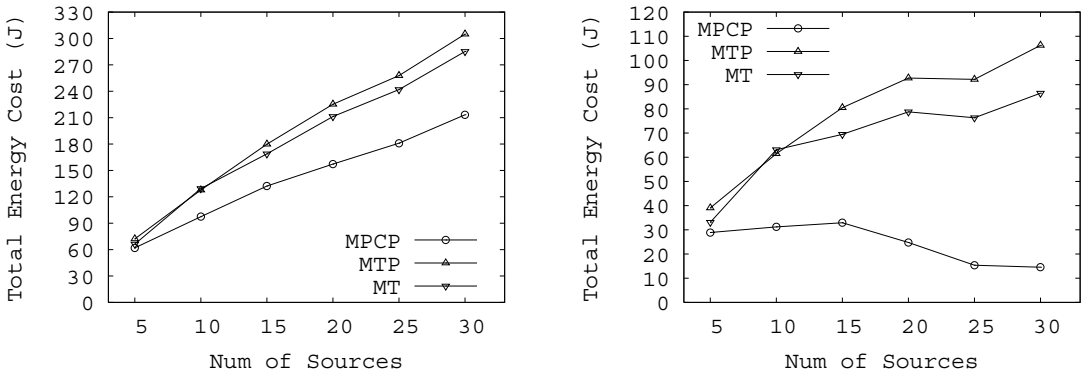
Figure 5.11: Routing topologies of different protocols with 20 sources.

more active sources on different communication paths and hence further reduce the number of active non-source nodes. For example, MPCP activates only one non-source node when there are 30 sources. This result clearly illustrates that MPCP can significantly reduce the energy wasted for idle listening by sharing active nodes on different communication paths.

The most important metric for our performance evaluation is energy consumption. For each protocol, we measure the difference between the total energy cost of the communicating network and that of an idle network where there is no communication activity and all nodes run in the power saving mode. This metric indicates the net energy consumed by a protocol due to the communication activities of the network. As shown in Fig. 5.12(a), MPCP consumes considerably less energy than other protocols. As the number of sources increases, routing paths from different sources share more nodes under MPCP and MASP, resulting in more energy reduction in the idle state

and better energy efficiency. The overall energy reduction of MPCP can be as high as 30% over MTP and 26% over MT.

Another interesting result in Fig. 5.12(a) is that, although MTP optimizes the transmission energy, it has a similar total energy cost to that of MT, even though MT makes simpler routing decisions based on the number of transmissions. As transmission power grows quickly with transmission distance, the routing paths found by MTP are likely to consist of more hops. Consequently, more nodes have to remain active on the routing paths, resulting in more energy wastage due to idle listening. On the other hand, although MT does not optimize transmission energy, its routings paths contain fewer hops and hence more nodes can run in power saving mode. In contrast to MTP or MT who only reduces the radio energy costs under partial working modes, MPCP effectively minimizes the total energy cost of radios based on data rates.

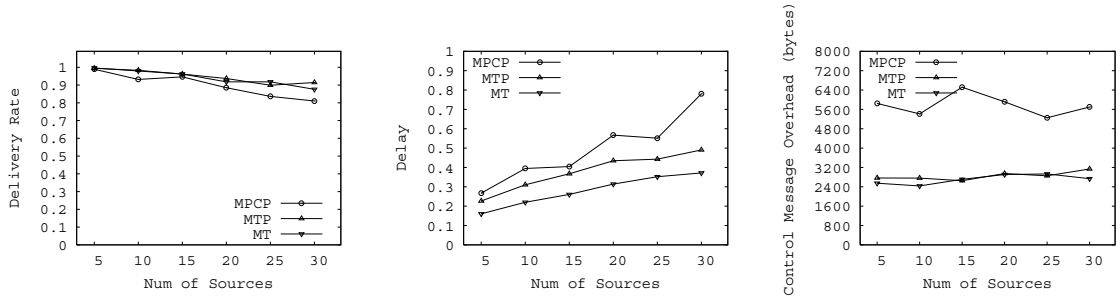


(a) Total network energy. (b) Total energy excluding idle energy of sources.

Figure 5.12: Energy consumption of different protocols.

We observe that, when the number of source nodes is large, most of the energy consumption is due to the idle listening of the sources. This phenomenon reduces the difference in total energy consumption between different protocols. To focus our analysis on the energy consumption of non-source nodes, we measure the difference

between the total energy consumption of the network and that of the same network where there is no communication activity. That is, a network all non-source nodes remain in the power saving mode but all source nodes remain in the idle state. This metric indicates the net energy consumption of the communication activities *excluding* the idle listening of source nodes. Fig. 5.12(b) shows that MPCP consumes at most 86% less energy than MT or 83% less than MTP. This result is consistent with the observation from the routing topology of MPCP in Fig. 5.11(c) that MPCP activates much fewer non-source nodes by effectively sharing intermediate source nodes on different paths. Another interesting result in Fig. 5.12(b) is that MPCP may consume less energy on intermediate nodes as the number of sources increases. This is because MPCP tends to route the data from a source through other sources that must remain active anyway. Reusing these sources, results in lower routing costs to the sink. More intermediate nodes may, therefore, run in power saving mode as the number of sources increases. We note that although the energy reduction by routing through other active sources is generally viable in the "many to one" communication pattern, it may be affected by the spatial distribution of sources in other scenarios.



(a) Delivery rate. (b) End-to-end delay. (c) Overhead

Figure 5.13: Communication performance and overhead of different protocols.

Next we evaluate the communication performance of the various protocols. Fig. 5.13(a) shows the data delivery ratio at the sink under different protocols. We can see that the delivery ratio of all protocols decreases slowly the more sources there

are in the network. MPCP delivers slightly less data than the other protocols when the number of sources exceeds 15. This occurs because MPCP causes slightly higher network contention due to path sharing between different sources when the network workload is high.

We plot the average end-to-end delay of data packets in Fig. 5.13(b). Not surprisingly, MT yields the shortest latency since it finds the routing paths with fewer retransmissions. MPCP yields a higher latency when network workload becomes higher due to the network contention caused by path sharing between different sources.

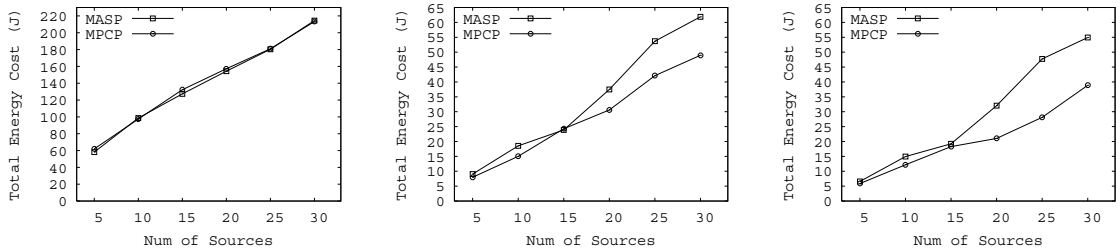
Finally, Fig. 5.13(c) shows the overhead of different protocols in terms of the total number of useful bytes in all route update messages. The overhead of MT and MTP is similar and remains roughly constant as more sources appear. MPCP incurs a higher overhead because the appearance of a new source node changes the node states and routing costs (see (5.13)), triggering more route updates than MTP and MT. However, consistent to the discussion in Section 5.4, most route updates are triggered by the first several sources and hence the total number of updates remains roughly the same as the number of sources increases. This behavior allows MPCP to scale well to large-scale networks. Despite the additional overhead compared with MT and MTP, MPCP still achieves significantly less energy consumption, as shown in Fig. 5.12(a) and (b).

5.5.4 Comparison of MPCP and MASP

As discussed in Section 5.4.2, MASP may incur a lower overhead than MPCP because it does not depend on information about the current set of sources and their data

rates. A disadvantage of MASP, however, is that its energy performance depends on the power characteristics of the radio. We now compare the performance of MPCP and MASP with different radio characteristics.

With the advancement in radio technology, the idle power of radio will continue to decrease in the future. To measure the impact of radio characteristics on MPCP and MASP, we simulate the two protocols using three different idle currents: 8 mA, 0.365 mA, and 0.02 mA. These three idle currents span three different orders of magnitude, and hence allow us to evaluate the energy performance of MPCP and MASP on a wide range of possible radio platforms. The transmission/reception current remains the same as the setting used in Section 5.5.2.

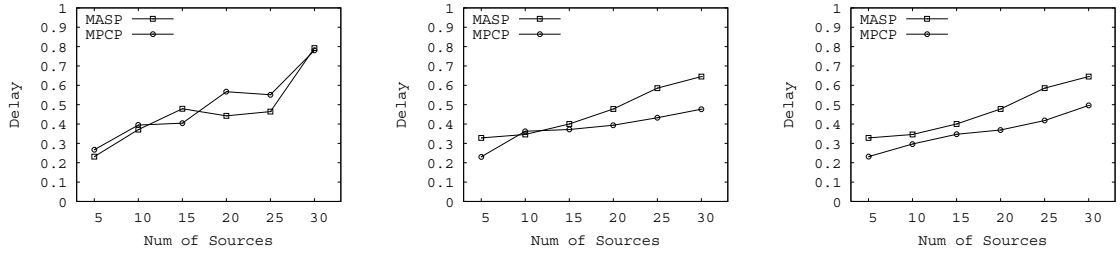


(a) Idle current is 8mA. (b) Idle current is 0.365mA. (c) Idle current is 0.02mA.

Figure 5.14: Energy consumption on different platforms.

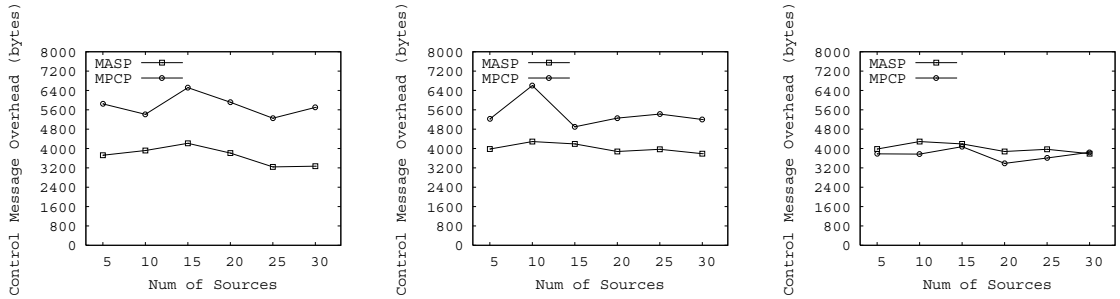
Fig. 5.14 shows the energy consumption of MPCP and MASP. When the idle current is 8 mA, MASP consumes similar energy to MPCP, even though MASP only minimizes the number of active nodes and does not directly optimize the overall energy consumption like MPCP does. MPCP considerably outperforms MASP when the idle current is lower. This result can be explained as follows. First, the achievable maximum bandwidth on multi-hop networks is fairly low compared with the ideal radio bandwidth. For example, the practical maximum bandwidth of Mica2 motes can barely reach 6 Kbps due to channel contention and lossy wireless links [26]. This

results in having only one sixth of the ideal radio bandwidth. Consequently, most energy consumption is due to idle listening of nodes instead of transmission/reception when the idle current is 8 mA. In other words, the impact of data rates on the overall energy consumption is limited when the idle current is high, making MPCP behave similar to MASP, as discussed in Section 5.3.3. When the idle current is 0.365 mA or 0.02 mA, the transmission/reception energy dominates the total energy consumption. In such a case, the performance of MASP degrades significantly as it only aims at minimizing the idle listening energy. This performance degradation of MASP is consistent to the analysis of the Steiner algorithm on which MASP is designed. As discussed in Section 5.3.4, the approximation ratio of the Steiner algorithm increases with α, which in turn increases as the idle current becomes lower. In contrast, MPCP yields a much better performance than MASP when the idle current is low because it always minimizes the total energy consumption of all radio states.



(a) Idle current is 8mA. (b) Idle current is 0.365mA. (c) Idle current is 0.02mA.

Figure 5.15: End-to-end delay on different platforms.



(a) Idle current is 8mA. (b) Idle current is 0.365mA. (c) Idle current is 0.02mA.

Figure 5.16: Routing overhead on different platforms.

Fig. 5.15 shows the end-to-end packet delay under MPCP and MASP. Consistent with the results on energy consumption, MPCP performs similar to MASP when the idle current is 8 mA and considerably outperforms MASP when the idle current is 0.365 mA or 0.02 mA. When the idle current is low, the routing cost under MPCP is dominated by the transmission/reception power (see (5.13)), resulting in a shortest-path tree like routing topology with more intermediate nodes than the Steiner tree like routing topology of MASP. A packet therefore travels fewer hops to the sink under MPCP causing the end-to-end delay to be shorter.

Finally, Fig. 5.16 shows the overhead of MPCP and MASP in terms of the total number of useful bytes in all route update messages. We can see that MASP incurs significantly lower overhead than MPCP when the idle current is 8 mA. This is due to the fact that each route update of MPCP contains more routing information as the routing cost depends on data rates. MPCP does, however, incur a lower overhead as the idle current decreases. In particular, MPCP incurs a overhead similar to MASP when the idle current is 0.02 mA. As the idle current decreases, the impact of node state on the routing cost, i.e., whether a node is active or not, decreases accordingly. As a result, the activation of nodes due to the appearance of new data flows causes fewer route updates. In contrast, MASP generates a similar number of route updates for all the three idle currents because the routing cost of a node in that protocol only depends on its state, i.e., whether the node is active or not.

The results in this section indicate that MPCP preserves the satisfactory performance under a wide range of radio characteristics. When the idle power of the radio is high, it reduces the energy wasted in the idle state by minimizing the number of active nodes. On the other hand, when the idle power of the radio is low, it saves energy by reducing the transmission energy. Such a joint optimization approach adopted by

MPCP enables it to flexibly adapt to different radio platforms. In contrast, MASP is only suitable for radios with high idle power and introduces less overhead than MPCP.

Chapter 6

Configurable Topology Control

Topology control is a key technique for improving energy efficiency and communication capacity of multi-hop wireless networks. In such a scheme, nodes minimize their transmission power while guaranteeing certain global network performance metric. Most previous works [50, 48, 45, 30, 36, 3, 37] aim at maintaining connectivity-based metrics of a network through reduced transmission power. However, network connectivity does *not* suffice to provide satisfactory communication performance when the underlying links among nodes are lossy. Recent empirical studies [81, 1] revealed that lossy and asymmetric links are common in real WSN deployments even with low network contention [84, 8]. These empirical findings have questioned some fundamental assumptions made by many existing works, like the unit disk graph (UDG) model of the radio. Although UDG allows a geometric treatment to the topology control problem, it does not capture the real characteristics of the lossy links since there does not exist a clear distance threshold that defines the communication range of a node [1]. Moreover, new topology control objectives need to be devised since traditional connectivity-based metrics do not guarantee satisfactory communication performance in presence of lossy wireless links. Finally, unlike many traditional ad hoc networks, many WSN applications only impose light load on the network. For instance, in the

WSN deployed at Great Duck Island for habitat monitoring [59], each of the 98 motes wakes up every 20 minutes, sends its sensor data to the base station, and goes back to sleep. Many other representative applications (e.g., precision agriculture and cargo tracking) also have low data rate. Topology control algorithms may be designed to exploit the light load of this common class of applications.

We propose a new formulation of topology control problem for lossy WSNs based on a new metric called *dilation of transmission count (DTC)*. DTC accounts for lossy links and quantifies the worst-case path quality of a network topology. We then propose a set of novel, localized *configurable topology control (CTC)* algorithms that can achieve different DTC bounds. CTC has two salient features. It can provide path quality assurance over lossy and asymmetric links in WSNs. Furthermore, it enables applications to achieve desired tradeoff between transmission power and path quality based on their specific requirements.

6.1 Problem Formulation

In this section, we first introduce a network model that captures the realistic properties of WSNs. We then give new formulation of the topology control problem in WSNs.

6.1.1 Network Model

Each node can transmit at any power from a discrete set $S = \{P_i | 1 \leq i \leq n\}$. $P_i > P_j \Leftrightarrow i > j$. For example, the CC1000 radio on Mica2 motes [15] can transmit at 32 different power levels.

The *packet reception rate*, $PRR_{u,v,i}$, is defined as the probability for node v to successfully receive a packet that u transmitted using power P_i. $PRR_{u,v,i}$ can be estimated based on the physical model of the radio [85, 84, 8], or using a link estimator [67, 12] that collects the packet reception statistics online. The *transmission count*, $R_{u,v,i}$, is defined as the expected number of transmissions needed for node u to successfully send a packet to v at power P_i. $R_{u,v,i} = 1/PRR_{u,v,i}$. Note that $R_{u,v,i}$ may not equal $R_{v,u,i}$ due to link asymmetry.

The network is denoted by a directed graph $G(V, E)$. V includes all nodes in the network. $E = \{(u, v, i) \mid R_{u,v,i} \leq T;\ u, v \in V;\ P_i \in S\}$. We ignore the links that have a transmission count higher than T[1]. $G(V, E)$ is a multi-graph where there may exist multiple links between two nodes at different power. Suppose u transmits to v at power P_1 and P_2. Then there are two directed links from u to v if $R_{u,v,i} \leq T,\ i = 1, 2$. The *transmission count of a path* is the sum of transmission counts of all links on the path. The *minimum transmission count* from u to v is the minimum transmission count among all paths from u to v.

We note that the above network model is very general. It can model realistic network properties such as lossy and asymmetric links. It can also be generalized to model heterogenous networks which have radios with different communication ranges.

[1] T corresponds to the maximum number of transmissions in the automatic repeat request (ARQ) mechanism at the MAC layer.

In this chapter, we focus on the WSNs with light workload and hence little interference caused by concurrent transmissions. Accordingly, we assume that a higher transmission power leads to a better link quality (and hence a lower transmission count), i.e., $P_i > P_j \Rightarrow R_{u,v,i} < R_{u,v,j}$. This assumption is justified by the fact that, when the interference is low, a higher transmission power will result in a higher signal strength, which alleviates the impact of path fading and noise. This assumption has been confirmed by several recent empirical studies on real WSNs [57, 12].

We assume nodes are stationary. Note that most existing WSNs comprise of stationary nodes.

6.1.2 Topology Control Problem

The problem of topology control has different formulations corresponding to different control strategies and optimization metrics. In this chapter, we consider both *per-node* and *per-link* power control strategies. While per-node power control assigns each node a power, per-link power control may assign a node different power for each link that originates at it. We consider two optimization metrics: min_sum that minimizes the total power of all nodes or links in the network, and min_max that minimizes the maximum power among all nodes or links. The min_max metric may lead to a longer lifetime by balancing the power consumption of different nodes. We first formulate the problem of topology control for WSNs with per-node control and the min_sum metric, and then give the formulations in other cases.

We use G_M to denote the network topology where each node transmits at the maximum power. G_M achieves the best path quality among all network topologies under

any possible power assignment when the network workload is light. When per-node power control is used, a *power assignment* assigns a transmission power to every node in the network. $G_\Omega(V, E_\Omega)$ represents the network graph under power assignment Ω. We define the *dilation of transmission count (DTC)* of G_Ω as the maximum ratio of the minimum transmission count between any two nodes in G_Ω to that between the same nodes in G_M. DTC quantifies the worst-case degradation in network's path quality under a power assignment relative to the maximum-power case.

Given a DTC bound $t \geq 1$ specified by the application, our objective is to choose a power assignment Ω that minimizes the total power of all nodes under the constraint that the DTC is no greater than t:

$$\Omega = argmin \sum_{P_i \in \Omega} P_i, \ subject \ \ to \\ \max_{u,v \in V} \frac{\Gamma_{G_\Omega}(u,v)}{\Gamma_{G_M}(u,v)} \leq t \tag{6.1}$$

where $\Gamma_{G_X}(u, v)$ denotes the minimum transmission count from u to v in the network under power assignment X.

The other variations of the topology control problem can be formulated similarly. When the optimization metric is *min_max*, the term to be minimized in the above formulation needs to be replaced by $\max_{P_i \in \Omega} P_i$. When the per-link power control is used, power assignment Ω assigns a transmission power to the source node of each link in the network.

6.2 The CTC Algorithms

In this section, we present a set of localized Configurable Topology Control (CTC) algorithms specially designed for lossy WSNs. CTC enforces the DTC requirement by replacing each max-power link with a low-power path that has a bounded transmission count relative to the replaced link. This strategy can be implemented in a localized fashion since a replacement path is likely located within the neighborhood of the replaced link in a dense network. However, the challenge is to ensure the replacement paths found by different nodes are consistent. The *key feature* of CTC is that it achieves this consistency in a localized fashion without any decision exchange among neighboring nodes.

In the rest of this section, we first describe the concept of neighborhood used by CTC. We then illustrate the basic idea of CTC using a simple example, followed by the detailed description of the CTC algorithms. Finally we present the theoretical analysis of CTC.

6.2.1 Neighborhood

The CTC algorithms use a two-hop neighborhood graph based on link qualities. Link (u, v, i) originates from u to v if $R_{u,v,i} \leq T$. Node v is node u's one-hop neighbor if there exists at least one link, $(u, v, i),\ P_i \in S$, from u to v. The one-hop neighborhood graph of u includes u and all the one-hop neighbors of u, and all the links from u to its neighbors. The two-hop neighborhood graph of node u is the union of the one-hop neighborhood graphs of u and u's neighbors. We use $N_i(u) = (V_i(u), E_i(u))$ $(i = 1, 2)$ to denote the one-hop and two-hop neighborhood graphs at node u.

Although links may be asymmetric, we require the neighborhood relation to be symmetric, i.e., $(u, v, i) \in E_1(u) \Leftrightarrow (v, u, j) \in E_1(v)$. Each node u can easily enforce this requirement by pruning the links to the neighbors who do not include u within their one-hop neighborhood.

In order to construct $N_2(u)$, node u needs to collect the packet reception rate (PRR) of the links within its two-hop neighborhood at different power levels. Each node can measure PRRs of one-hop links based on data or hello messages, and exchange them with its one-hop neighbors. Efficient algorithms for neighborhood discovery and link quality estimation have been proposed in earlier work [67, 12] and is not the focus of this thesis.

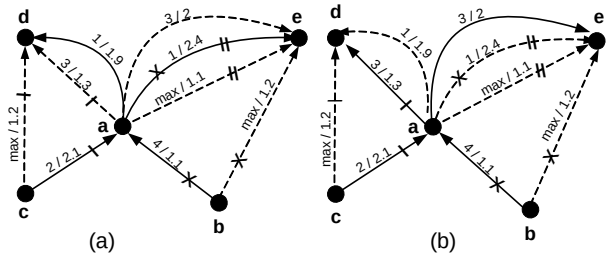


Figure 6.1: The execution of two different algorithms.

6.2.2 An Illustrative Example

In this section, we illustrate the basic idea of CTC using a simple example depicted in Fig. 6.1. We will discuss how CTC is executed at three nodes a, b, and c when per-node power control and the min_sum metric are used. Fig. 6.1 only shows a subset of the links that exist between nodes $a \sim e$ for the clarity of illustration. In this example, the DTC bound specified by the application is 3. Fig. 6.1(a) illustrates a naive algorithm in which each node only replaces its own max-power links. Fig. 6.1(b) illustrates the CTC algorithm with the min_sum metric. Each link is labeled

by *power / transmission count*. *max* represents the maximum transmission power. Solid radials represent the actual links after the execution of the algorithm. The max-power links and their corresponding replacement paths are labeled by the same symbols.

We first discuss the naive localized algorithm in Fig. 6.1(a) that may result in conflicting power assignments by different nodes. In this algorithm, each node independently replaces each of the max-power links that originate from it with a low-power path whose transmission count satisfies the specified DTC bound. Fig. 6.1(a) depicts a possible configuration output after the algorithm is executed at nodes a, b, and c. Node b replaces the max-power link (b, e, max) with path $(b, a, 4) \rightarrow (a, e, 1)$. The total transmission counts of the new path is $1.1 + 2.4 = 3.5$, which is lower than triple of that of the replaced max-power link (b, e, max). Similarly, nodes a and c replace their max-power links (a, e, max) with $(a, e, 1)$, and (c, d, max) with $(c, a, 2) \rightarrow (a, d, 3)$. Notice that a is assigned two different powers, 3 and 1, on the three replacement paths. If each node sets its power independently according to the replacement paths it finds, node a will set a power of 1 as it is not aware of the existence of the other replacement paths. As a result, the actual quality of link from a to e is lower than required by the path found by b. Consequently, the path from c to d has a dilation of $(2.1 + 1.9)/1.2 = 3.3$ that violates the required DTC bound of 3. We can see that this problem is caused by the inconsistency of the local paths found by different nodes. An simple solution to resolve such conflicts is to have nodes exchange their local solutions with their neighbors. However, such solution is not desirable due to the overhead and convergence latency.

We now discuss how CTC solves this problem. The basic idea is that, in addition to replacing its own max-power links, each node also computes its power assigned by

its neighbors on their local paths. As a result, it always chooses a power no lower than any power assigned by itself and its neighbors, which preserves the dilation of all replacement paths.

Specifically, a node finds a replacement path for each max-power link in its two-hop neighborhood. The replacement path must yield the minimum total power among all possible paths that satisfy the dilation constraint. For instance, the replacement path of (b, e, max) is $(b, a, 4) \rightarrow (a, e, 1)$, which has the minimum total power among all paths from b to e with a dilation lower than 3. Node a starts with the lowest power, and once finds a new replacement path that includes itself, it increases its power to match its power assigned on the path if necessary. As shown in Fig. 6.1(b), node a first assigns itself a power of 1 after replacing the links (a, e, max) and (b, e, max), and then increases its power to 3 after finding the replacement path for links (c, d, max). As a result, all replacement paths are preserved after a executes the algorithm.

We can see from Fig. 6.1(b) that all the nodes on a replacement path find the same path when they replace the same max-power link by executing the algorithm locally. For example, the path $(c, a, 2) \rightarrow (a, d, 3)$ is found by both c and a to replace (c, d, max) in their executions. As a result, the dilation of the path is preserved as a and c will assign their power no lower than the values on the path. In other words, the network after power reduction satisfies the dilation requirement. We offer a rigorous proof of the correctness of a generalized algorithm in Section 6.2.5.

6.2.3 Per-node Power Control

In this section, we present the CTC algorithm with the per-node control strategy. We first describe the algorithm with the min_sum metric, and then discuss how the algorithm can be modified to adopt the min_max metric. For each max-power link, CTC finds a replacement path composed of up to d hops in the node's two-hop neighborhood[2]. d is referred to as *search depth* hereafter. A higher search depth increases the opportunity for a CTC algorithm to find lower power assignment at the cost of higher computation complexity.

The algorithm executed at node u with the *min_sum* metric is depicted in Fig. 6.2. To enforce consistent power assignments on the replacement paths found by different nodes, u "simulates" the execution of the algorithm at other nodes by invoking *LabelSet(v)* for each node $v \in V_1(u)$. Function *LabelSet(v)* finds the replacement paths with dilation bound t for all the max-power links that originate from v. Special care needs to be taken at this step since a node has different neighborhood view from its neighbors. The key of the CTC algorithms is that if a node lies on a replacement path found by its neighbor, it will also find the same path in its own execution of the algorithm. Once node u finds a replacement path that includes it, it improves its power to match its power assigned on the path if necessary.

The function *LabelSet* extends the Generalized Permanent Labeling Algorithm (GPLA) [16] proposed for the shortest path problem with time window (SPPTW). A special case of SPPTW, the weight-constrained shortest path (WCSP) problem, resembles finding a replacement path in our problem. Each link in a WCSP problem is assigned

[2]Note that a two-hop neighborhood graph consists of all the immediate links of the node's one-hop neighbors. Hence, its diameter can be higher than two.

two different weights. The goal is to find the shortest path between two nodes in terms of one weight metric under the constraint that the total weights of the other metric is bounded. The power and transmission count of a local path correspond to the two different weight metrics in a WCSP problem. However, different from GPLA that finds a single best path between two nodes, *LabelSet(v)* finds the best replacement paths from v to all its neighbors. In additional to minimizing the total power of a replacement path, we also extend GPLA to incorporate other optimization metrics like min_max.

Input: t, d, $N_1(u)$, $N_2(u)$
Output: $power(u)$

$power(u) = min$;
for $v \in V_1(u)$ **call** $LabelSet(v)$; **end**

function *LabelSet(v)*

1. $W = t \cdot \max\{R_{v,w,max} | (v, w, max) \in E_1(v)\}$. Set $L_v = \{(0, 0)\}$ and $L_i = \emptyset$ for all $i \in V_1(v) - \{v\}$.
2. If all labels have been marked, go to 5); else choose $i \in V_1(v)$ that has an unmarked label (R_i^q, P_i^q) with minimal R_i^q.
3. For each link $(i, j, k) \in E_2(u)$ do
   $L_j = L_j \cup \{(R_i^q + R_{i,j,k}, P_i^q + P_k)\}$, if the following conditions are met:

$$R_i^q + R_{i,j,k} \leq W \tag{6.2}$$

$$|q| < d \tag{6.3}$$

$$j \in \bigcap_{k \in q} V_1(k) \tag{6.4}$$

$$\nexists (R_j^q, P_j^q) \in L_j,$$
$$(R_j^q \leq R_i^q + R_{i,j,k}) \wedge (P_j^q \leq P_i^q + P_k) \tag{6.5}$$

4. Mark label (R_i^q, P_i^q). Go to step 2.
5. For each link (v, w, max) in $E_1(v)$, do:
   (a) Find the label (R_w^q, P_w^q) in L_w such that $R_w^q \leq t \cdot R_{v,w,max}$ and has the minimal P_w^q.
   (b) If there exists a u's link $(u, z, k) \in q$ and $power(u) < P_k$, $power(u) = P_k$.

Figure 6.2: The Per-node CTC Algorithm with the *min_sum* metric (executed at u)

LabelSet(v) is a dynamic programming procedure during which the partial paths found are stored by *labels* on nodes. Specifically, a label on node i is a tuple (R_i^q, P_i^q) where q corresponds to a path from v to i, and R_i^q and P_i^q are the transmission count and total power of the path respectively. Such a path is a candidate replacement path for the max-power link from v to i, and can also be a partial path on the replacement paths for the links from v to other neighbors. L_i represents the set of labels on i that corresponds to all such partial paths.

The procedure starts from the source node v with label $\{(0, 0)\}$ and all label sets on other nodes are initiated to be empty. Then the algorithm executes in iterations. In each iteration (composed of step 2 to 4), an existing label (R_i^q, P_i^q) with minimum transmission count is extended along all outgoing links of node i, which corresponds to extending the partial path q to all possible next-hop nodes (see step 3). The label is *marked* after all such next-hop nodes are found (see step 4). The search process initiated from source node v terminates if all labels on the nodes within $V_1(v)$ have been marked.

Step 3 extends label (R_i^q, P_i^q) along a link (i, j, k) by adding the transmission count and power of link (i, j, k) to R_i^q and P_i^q respectively. The link will be added to the label set of j, if the constraints (6.2)-(6.5) are met.

Constraint (6.2) requires that the total transmission counts of the expanded path must be smaller than W that is t times the maximum transmission count of all the max-power links originated from v. This constraint reduces the search space by eliminating the paths that would have a dilation higher than t. Note that (6.2) is a necessary but not sufficient condition. That is, a path that satisfies (6.2) may still exceed the dilation bound t because W is equal to t times the *maximum* transmission count of

all max-power links of v. However, such paths are still preserved during the searching process since they may be segments of valid replacement paths found later in the process. Constraint (6.3) limits the maximum hop count of a path to d. Constraint (6.4) enforces that all nodes on a path are located within the one-hop neighborhood of each other. As we will see in the analysis in Section 6.2.5, this constraint is critical for ensuring the consistency in the power assignments computed by different nodes.

Finally, (6.5) ensures that there does not exist a label on the next-hop node that represents a better path than the extended path. A path A is better than path B if and only if A has a lower transmission count *and* lower power than B. If (6.5) does not hold, we keep the paths with higher power but lower transmission count, or the paths with higher transmission count but lower power, since they both may satisfy constraint (6.2) and evolve into valid replacement paths in following iterations.

At the end of the procedure, for each max-power link (v, w, max), we choose a path found from v to w (represented by a label on w) as the replacement path if it has the minimum total power among all paths that satisfy the dilation constraint (see step 5.a). Note that such a path must exist since in the worst case the max-power link (v, w, max) will be found. Finally, if node u (that executes the algorithm) lies on the replacement path, it sets the power to the max of its current power and the power on the path.

Minimizing the maximum power on a replacement path may lead to more balanced power on different nodes. We modify CTC depicted in Fig. 6.2 as follows to adopt the min_max metric. In a label (R_i^q, P_i^q), instead of storing the total power of path q in P_i^q, we redefine P_i^q as the maximum power of the links on q. Accordingly, constraint (6.5) needs to be changed to $\nexists (R_j^q, P_j^q) \in L_j,\ (R_j^q \leq R_i^q + R_{i,j,k}) \wedge (P_j^q \leq \max(P_i^q, P_k))$.

6.2.4 Per-link Power Control

Different from per-node control that restricts a node to a fixed power, per-link control allows a node to use different power to transmit to different neighbors. As a result, per-link control may lead to more energy saving. An advantage of the algorithm depicted in Fig. 6.2 is that it can be easily modified to use per-link control. Specifically, node u stores a power $power(u, v)$ with an initial value of minimum power for each of its one-hop neighbors, $v \in V_1(u)$. In addition, step 5.b needs to be modified as follows: If there exists u's link $(u, z, k) \in q$ and $power(u, z) < P_k$, $power(u, z) = P_k$. Notice that both per-node and per-link control share the same procedure for searching replacement paths (step 1 to 4 of function *LabelSet* in Fig. 6.2). Hence, the same modification introduced in Section 6.2.3 can also be used to adopt different optimization metrics, including min_sum and min_max, in per-link control.

6.2.5 Correctness of CTC

We now prove the correctness of CTC. We first show that CTC with per-node control and the min_sum metric achieves the required dilation bound. We then extend this result to per-link control and the min_max metric.

Theorem 15. *Suppose M is the power assignment where each link is assigned the maximum power, Ω is the power assignment produced by the CTC algorithm with a DTC bound $t \geq 1$. Then the network G_Ω satisfies the DTC bound t:* $\max_{u,v \in V} \frac{\Gamma_{G_\Omega}(u,v)}{\Gamma_{G_M}(u,v)} \leq t$.

Proof. To prove that the minimum transmission count between any pair of nodes in G_Ω is no greater than t times of that between the same pair of nodes in G_M, we show

that any link in G_M, say (v, w, max), is replaced by a path in G_Ω with a dilation no greater than t. We prove that this holds after the execution of the algorithm in Fig. 6.2 at each node. We use $F^u_{v,w}$ to denote the local replacement path from node v to node w found by node u in u's execution of the algorithm. Note that $F^u_{v,w}$ corresponds to the label (R^q_w, P^q_w) found by u at step 5.a. Since (v, w, max) is replaced by u with a path $F^u_{v,w}$ that has a dilation no greater than t (enforced at step 5.a), it remains to show that this local path is preserved by the power choices made by other nodes on the path in their executions of the algorithm. Suppose (x, y, i) is an arbitrary link on path $F^u_{v,w}$. That is, u assigns power P_i to x. We now show that the power choice of node x, $power(x)$, computed by itself is no lower than its power, P_i, assigned by u on path $F^u_{v,w}$.

One observation is that $x \in V_1(u)$ since a node in $V_2(u) - V_1(u)$ does not have any outgoing links, and hence (x, y, i) would not exist. It is easy to see that $power(u) \geq P_i$ holds if $x = u$ according to step 5.b in Fig. 6.2. We now show $power(x) \geq P_i$ also holds for any $x \in V_1(u) - \{u\}$.

We define graph $G^* = (V^*, E^*)$ as follows.

$$
\begin{aligned}
V^* &= \bigcap_{k \in V_1(u)} V_1(k) \\
E^* &= \bigcup_{a,b \in V^* \wedge (a,b,i) \in E_1(a)} (a, b, i)
\end{aligned}
$$

V^* comprises the set of nodes that are included in the one-hop neighborhood graphs of u and all u's one-hop neighbors, and E^* comprises the links between these nodes. In

the following, we will prove that G^* is the searching space for the replacement path of (u, v, max) in both x and u's local executions of the algorithm. First, constraint (6.4) in Fig. 6.2 enforces that all possible replacement paths of (u, v, max) must reside in G^*. Second, $E^* \subseteq E_2(u) \bigcap E_2(x)$, because E^* contains only the links between the nodes in V^* that are included by both $V_1(u)$ and $V_1(x)$ by definition. Hence $G^* \subseteq N_2(u) \bigcap N_2(x)$. In other words, both u and x search for the replacement path of (u, v, max) in G^*. This conclusion, combined with the fact that the replacement path is optimal (in terms of total power) among all paths in G^* that satisfy the length constraint ((6.3) in Fig. 6.2) and the dilation constraint, leads to the conclusion that x and u find the same replacement path $F^u_{v,w}$. Hence, according to step 5.b, the power of x after its execution of the algorithm is no lower than its power on the replacement path. We have shown that each replacement path is preserved by the power choices made by all the nodes in their local executions of the algorithm. That is, each max-power link is replaced by a path with a dilation lower than t after the execution of the algorithm at each node. Therefore, the resultant network has a dilation lower than t. □

We note that similar arguments can prove the correctness of the CTC algorithm when it adopts per-link control strategy or the min_max metric. This is because, the nodes on a replacement path will find the same path as long as the the path is optimal (in min_sum or min_max metric) within the neighborhood shared by all the nodes on the path.

6.2.6 Time Complexity of CTC

We now analyze the time complexity of CTC. Suppose the number of links in each node's two-hop neighborhood is bounded by $O(E_2)$. Procedure $LabelSet(v, W)$ without constraints (6.3) and (6.4) is similar to the original GPLA algorithm that has a complexity of $O(|E_2|W)$. Since we only keep undominated labels, there is a at most one label kept for each value of transmission counts, and there are at most W values of transmission counts. That is, a node has at most W labels. Hence, in step 2, a link is processed at most W times. Summing the number of times an link is processed over all links gives a time complexity of $O(|E_2|W)$. We note that this complexity is pseudo-polynomial as it depends on parameter W.

On the other hand, the actual time complexity of $LabelSet(v, W)$ is lower due to the constraints (6.3) and (6.4) in Fig. 6.2. Specifically, (6.3) requires the number of hops of a path to be smaller than d. Suppose the number of nodes within a one-hop neighborhood is bounded by $O(V_1)$, the total number of link processing in *LabelSet* is bounded by $O(|V_1|^{d-1})$. Hence the time complexity of *LabelSet* is bounded by $O(min(|V_1|^{d-1}, |E_2|W))$. Since *LabelSet* is invoked for each one-hop neighbor, the overall time complexity of the generalized CTC algorithm is $O(|V_1| \cdot min(|V_1|^{d-1}, |E_2|W))$. We note that this complexity result is an upper bound, which does not consider constraint (6.4). Although this bound is exponential in d, we show experimentally that small search depth, (e.g., choosing $d = 2$ or 3) gives a very good performance in Section 6.3. This is, in part, because it is unlikely to find long replacement paths within a two-hop neighborhood.

6.3 Evaluation

We have evaluated CTC through two sets of simulations. We first study the network topology produced by CTC using a simple simulator. We then evaluate CTC in realistic packet-level simulations using an open-source WSN simulator called Prowler [55]. To create a realistic simulation environment, we implemented the probabilistic link model from USC [85] in both simulators. Previous experiments showed that the USC model produces lossy and asymmetric links similar to Mica2 motes [85].

6.3.1 Quality of Network Topology

In this section, we evaluate how well CTC can configure a network topology to different dilation bounds using the simple simulator. The packet reception ratio of each link is computed according to the link model from USC [85]. These simulations allow us to evaluate the quality of network topologies produced by different CTC algorithms under a wide range of settings.

In each simulation, nodes are uniformly deployed in a $150 \times 150\ m^2$ region. The total number of nodes is 100 unless otherwise mentioned. Each data point presented in the results is the average of five different network topologies. Its 90% confidence interval is also shown. Each node can transmit at 11 different power levels from -20 dbm to 10 dbm, at an increment of 2 dbm[3].

We compare CTC against an existing topology control algorithm called LMST [38]. Each node running LMST builds a minimum spanning tree (in term of Euclidean

[3]The Chipcon CC1000 radio on Mica2 motes supports 32 power levels. While we only use 11 power levels in our simulations, using more power levels may further improve the performance and configurability of the network at the cost of higher overhead.

distance) within its neighborhood and reduces its transmission power to reach only the neighbors on the tree. We chose LMST as the baseline algorithm for performance comparison since it is reported in [38] that it yields a better power efficiency than several earlier algorithms such as CBTC [36] and R&M [50].

The original design of LMST relies on the neighborhood within the maximum communication range of a node. However, the notation of communication range is not applicable when communication links are lossy. We extend LMST to handle lossy networks by blacklisting all links with a PRR lower than a threshold. A node includes another node in its neighborhood only when the link to the node has a PRR higher than the preset threshold.

We first vary the search depth of CTC from 2 to 5 to evaluate its impact on the topology quality. For each combination of optimization metric and search depth, we measure the DTC of the network topology configured by each algorithm. Each setting is denoted as *CTC-control-metric-depth*. For example, *CTC-node-mm-3hop* represents the per-node control algorithm with the min_max metric with a search depth of 3 hops.

Fig. 6.3 shows the measured DTC under CTC-node with different search depths when the required dilation ranges from 1.5 to 5.5. CTC-node-ms yields the same DTC 1.5 irrespective the search depth. This is because the min_sum metric results in unbalanced node power on replacement paths. As a result, a node has a high chance of being assigned a high power even when the search depth is small, because it lies on many replacement paths. When the search depth increases, CTC-node-mm achieves a better configurability as it can find replacement paths with lower power. Fig. 6.3 shows that CTC-node can produce highly configurable network topologies with a

search depth as low as 3. A small search depth is desirable as the time complexity of CTC increases with the search depth.

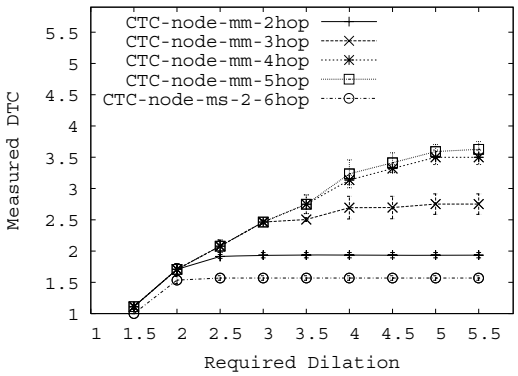


Figure 6.3: Measured DTC of per-node CTC algorithms

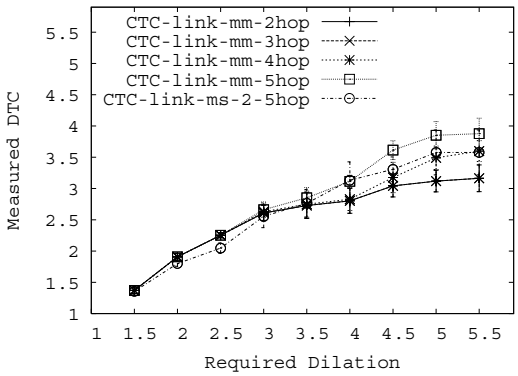


Figure 6.4: Measured DTC of per-link CTC algorithms

Fig. 6.4 shows the measured DTC under the CTC-link algorithms. Similar to CTC-node-ms, CTC-link-ms yields the same DTC irrespective the search depth. We can see that CTC-link demonstrates better configurability than CTC-node. This is because per-link control allows a node to use different transmission power when it lies on multiple replacement paths. Furthermore, a search depth of only 2 enables CTC-link to achieve a high degree of configurability at low computation cost. Overall our results show that the CTC-link algorithms can provide more efficient and flexible topology control than the CTC-node algorithms.

Fig. 6.5 compares the DTC of CTC and LMST algorithms under different node densities. LMST-0.4 and LMST-0.6 represent the LMST algorithm with a PRR threshold of 0.4 and 0.6, respectively. Under all node densities, CTC consistently produces topologies that satisfy the required quality bounds. In contrast, the DTC of LMST has a high variation for different networks with the same density, and is heavily affected by node densities. This is because LMST tends to connect nodes with short and

low-power links that are more likely to be lossy. This result shows that connectivity-based topology control algorithms cannot provide guaranteed path quality in lossy WSNs as they do not account for link quality. The DTC of LMST decreases with a larger PRR threshold, because the links retained by each node become more reliable. However, a high PRR threshold may cause a node to blacklist too many links resulting network partition. Choosing a PRR threshold for LMST that achieves both low DTC and network connectivity under different network settings is difficult. We set the maximum PRR threshold to 0.6 in the following simulations as it results in the lowest DTC without partitioning the network under our settings.

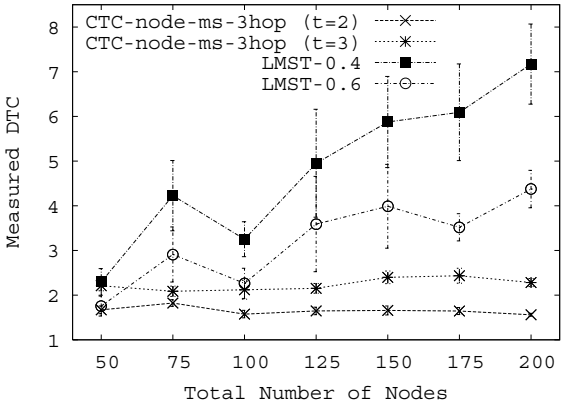


Figure 6.5: Measured DTC of CTC and LMST.

6.3.2 Simulation Settings on Prowler

We also implemented CTC and the USC link model in Prowler [55], an open-source network simulator that employs a layered event-driven structure similar to TinyOS. The MAC protocol employs a CSMA/CA scheme similar to B-MAC [47]. The maximum number of retransmissions before dropping a packet is 3. DSDV [46] is used as the routing protocol. Similar to the MT protocol [67] we modified DSDV [46] to

use transmission count as the routing metric which is superior to hop count in lossy wireless networks [67, 20, 14].

The node distributions are the same as in the first set of simulations. The node bandwidth is 40 Kbps. The data packet size is 120 bytes. Each node runs an online link estimator similar to the one described in [67] to estimate the link quality in its two-hop neighborhood. The network follows a traffic pattern in typical data collection applications [59]. Every source sends a packet to the base station every 5 minutes. The base station is located in the middle of the right border of the region. Sources are randomly chosen from the left 60% of the region to increase the distance to the base station. We varied the number of sources from 5 to 50. Each result is the average of 10 different network topologies with a 90% confidence interval. Each run lasts 80 minutes of simulation time.

6.3.3 Performance Results

We evaluate both communication performance and energy consumption of different algorithms. The search depth of all CTC algorithms is set to 3. We used two CTC algorithms: ctc-node-mm with a required DTC bound of 2, and ctc-link-ms with a required DTC bound of 3. Besides LMST, we also use the network where each node transmits at the maximum power as a baseline, which is denoted by *MAX-POWER* in all figures. As light workload is generated in our simulations, MAX-POWER yields the best performance in terms of delay and delivery ratio.

Fig. 6.6 shows the data delivery ratio under each algorithm. Similar to MAX-POWER, all CTC algorithms delivered over 95% of the total packets to the base station. LMST yields the lowest delivery ratio due to the lossy links on its topology.

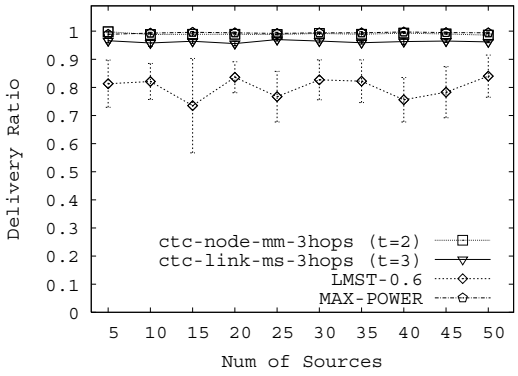


Figure 6.6: Packet delivery ratio

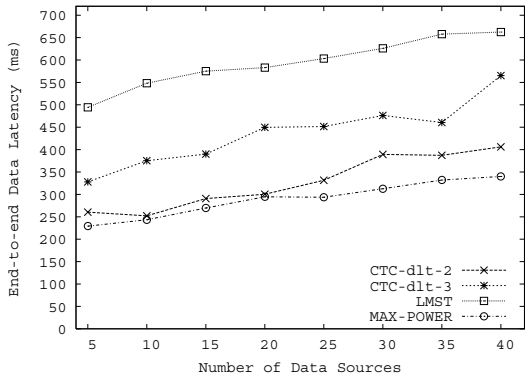


Figure 6.7: Average delay of the received packets at the sink

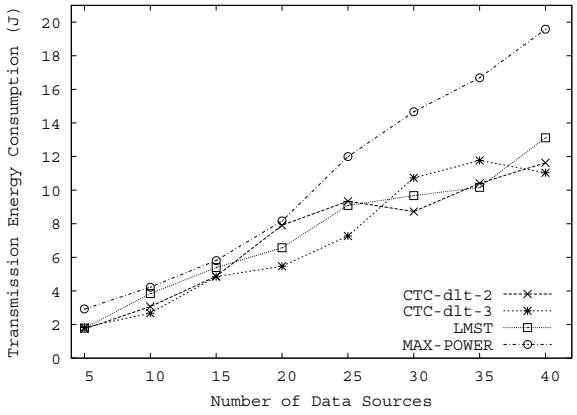


Figure 6.8: Measured DTC of CTC and LMST.

Fig. 6.7 shows the average delay of the received packets at the base station. LMST yields the highest delay because a packet often experiences retransmissions over lossy links. Both CTC algorithms achieve lower delay than LMST. Furthermore, the delay under CTC increases with a higher DTC bound. This result shows that CTC enables

applications to effectively control the network performance by adjusting the DTC bound.

Fig. 6.8 shows the transmission energy consumed by different algorithms. CTC-link performs slightly better than CTC-node. Interestingly, although LMST assigns *lower* power than the other algorithms, the network consumes almost the same amount of *energy* under LMST as under MAX-POWER. This is because, the links on LMST's topology are less reliable resulting in excessive energy waste on packet retransmissions. Therefore, the benefit of lower power is offset by the increase in the number of transmissions in lossy networks. In contrast, CTC-link-ms reduces the energy consumption by 27% $\sim$ 36% compared with MAX-POWER. This result demonstrates the importance of considering lossy link models in both design and evaluation of topology control.

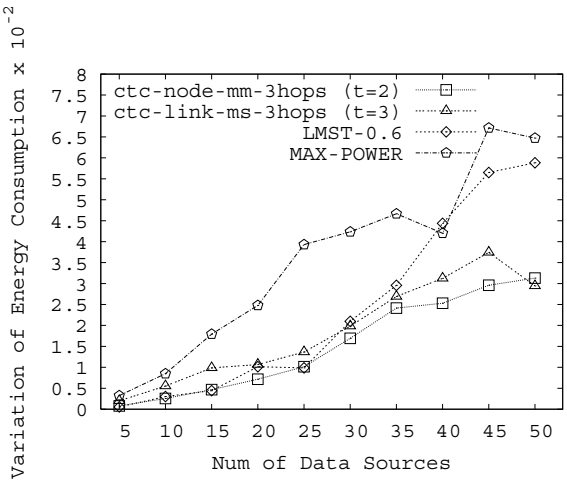


Figure 6.9: The variation of transmission energy of all nodes.

Fig. 6.9 shows the variation of nodes' transmission energy consumption in a typical run. The variation of the energy consumption affects the lifetime of the network before partition. Both CTC-node and CTC-link achieve significantly lower variation in nodes' energy consumption than LMST when source density is high. They also achieve much more balanced energy consumption in the network than MAX-POWER

under all source densities. This result indicates that CTC can effectively prolong the lifetime of the network.

Chapter 7

Probabilistic Coverage Maintenance

Coverage maintenance protocols provide required sensing coverage over a geographic region by activating a subset of nodes, while scheduling the others to sleep, to conserve energy. Unfortunately, although a number of coverage maintenance protocols have been developed [9, 61, 65, 75, 76], they are often designed based on deterministic detection models (such as the disc model) that are not applicable to real-world distributed sensing applications. For example, existing protocols assume that each sensor performs sensing independently. This assumption is invalidated by many sensor network applications which rely on data fusion to improve sensing performance.

In this chapter, we investigate coverage maintenance based on a probabilistic distributed detection model that allows efficient data fusion from multiple nodes. We characterize coverage by the minimum event detection probability in a region and the system false alarm rate from the active nodes in a network. In our model, the event detection probability and false alarm rate are computed based on an existing data fusion algorithm that correlates detection decisions from multiple nodes. While the

adopted fusion algorithm is not new, this coverage formulation provides a basis for bridging the gap between coverage maintenance protocols and distributed detection algorithms.

Distributed detection problem has been shown to be computationally difficult and even NP-complete in some cases [62]. In general, the computational cost for determining the detection probability from a large number of nodes is high due to the need to consider the combination of detection decisions from multiple nodes. Consequently, the key challenge to developing a coverage maintenance protocol that is based on realistic distributed detection model is to (re-)configure the network coverage within a short time while reducing the number of active nodes in order to prolong the network lifetime. A coverage configuration (or *configuration* for abbreviation) of a sensor network is characterized by a set of active nodes that can maintain the coverage of the network. Reconfiguration is needed at runtime when the current active nodes fail to cover the network. Quick reconfiguration is particularly important in large and dynamic sensor networks subject to node failure and changing application requirements. Unfortunately, the need to reduce the number of active nodes and the network coverage configuration time can conflict with each other. For example, while a centralized algorithm that treats the whole network as a single fusion group can result in a small number of active nodes, it often requires an extremely long time to configure a large network. On the other hand, although a protocol can significantly reduce the configuration time by dividing the network into separate fusion groups that can configure themselves in parallel, it may result in excessive number of active nodes due to the lack of coordination among neighboring fusion groups.

We develop a novel distributed coverage maintenance protocol called the *Coordinating Grid (Co-Grid)* protocol that meets both key requirements. Co-Grid organizes

the network into fusion groups located on overlapping virtual grids. Through effective coordination among neighboring fusion groups, Co-Grid can achieve comparable number of active nodes as the centralized algorithm. Furthermore, the coverage configuration time of Co-Grid is competitive with the protocol based on separate fusion groups. We also present a theoretical analysis of the efficiency of the Co-Grid protocol. Despite the inter-dependency among fusion groups, our analysis shows that Co-Grid can achieve a high degree of parallelism in the coverage configuration process. This property enables Co-Grid to accomplish configuration quickly and scale to large networks.

7.1 Detection Model

In this section, we present a distributed detection model that can be combined with our coverage maintenance protocol. We assume that each node in a sensor network belongs to one or more *fusion groups*. Each fusion group has a node serving as its *fusion center*. Each node in the fusion group measures the signal and makes its own *local* decision on whether a target is present or not. Then the local decisions of individual nodes are transmitted to the fusion center. The fusion center uses a fusion rule to reach a global decision based on the local decisions.

7.1.1 Signal Model

In a large-scale sensor network, the system detection performance depends on the spatial distribution of nodes. To capture the correlation between the spatial property of a sensor network and the system detection performance, we introduce spatial

signal decay parameters into the Neyman-Pearson detector model [64]. Each node detects the targets of interest by measuring the sound power with its acoustic sensor. The power of the acoustic signal emitted by a target decays over the distance of propagation. Specifically, for a target located at point (x, y), we assume that the signal power measured by node i located at position (x_i, y_i) can be described by the following equation:

$$e(x_i, y_i) = \frac{be_0}{d((x, y), (x_i, y_i))^a} \tag{7.1}$$

where $d((x, y), (x_i, y_i))$ is the distance between the target and node i. e_0 represents the initial power of the signal emitted by the target. We assume that the location of a potential target is not known to the nodes in advance. a and b are attenuation factors determined by propagation properties of sound signals.

A node makes a decision on whether a target is present or not based on its measurement. The noise in the measurement of a node is modeled as a Gaussian distribution with zero-mean and the target signal is modeled as a Gaussian distribution with nonzero-mean. We assume that when the target is present, the mean of the target signal observed at node i equals the square root of the signal power. Thus the task of detection at node i is to test the following two hypotheses:

$$H^0 : \ p(z_i|H^0) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{z_i^2}{2\sigma^2}\right) \tag{7.2}$$

$$H^1 : \ p(z_i|H^1) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(z_i - \sqrt{\frac{be_0}{d((x,y),(x_i,y_i))^a}})^2}{2\sigma^2}\right) \tag{7.3}$$

where z_i represents the measurement at node i. H^0 and H^1 represent the hypothesis that the target is absent and present, respectively. When the signal power decays

with distance, the difference between the means of the two hypothesis decreases accordingly, resulting in worse detection performance.

7.1.2 Decision Rules

We assume that node 0 serves as the fusion center in a fusion group and all other nodes ($1 \sim n$) in the fusion group send their decisions to node 0 which makes a global decision. In this section, we investigate the decision rule at node i ($1 \leq i \leq n$) and the fusion center, respectively.

At node i ($1 \leq i \leq n$), the optimal decision rule is LRT (Likelihood Ratio Test)[64]:

$$\frac{P(z_i|H^1)}{P(z_i|H^0)} \underset{u_i=0}{\overset{u_i=1}{\gtrless}} \lambda_i \tag{7.4}$$

where u_i represents the decision (0 or 1) of detection at node i. Using (7.2) and (7.3), the LRT can be transformed to a test on node measurement z_i and a *decision threshold* λ_i^*, i.e., node i decides on one (a target is detected) if its measurement z_i is greater than λ_i^*, otherwise it decides on zero (no target is detected).

For node i ($1 \leq i \leq n$), the *false alarm rate*, denoted by P_{Fi}, represents the probability that node i decides on one while no target is present. The *detection probability*, denoted by P_{Di}, represents the probability that a target located at (x, y) is detected by node i. The false alarm rate (detection probability) of node i ($1 \leq i \leq n$) is referred to as the *local* false alarm rate (detection probability) while that of the fusion center

is referred to as the *system* false alarm rate (detection probability). The local false alarm rate and detection probability are given by:

$$
\begin{aligned}
P_{Fi} &= \int_{\lambda_i^*}^{\infty} p(z_i|H^0)dz_i && (7.5)\\
&= Q\left(\frac{\lambda_i^*}{\sigma}\right) && (7.6)\\
P_{Di}(x,y) &= \int_{\lambda_i^*}^{\infty} p(z_i|H^1)dz_i && (7.7)\\
&= Q\left(Q^{-1}(P_{Fi}) - \frac{\sqrt{\frac{be_0}{d((x,y),(x_i,y_i))^a}}}{\sigma}\right) && (7.8)
\end{aligned}
$$

$Q(x)$ is given by $\frac{1}{\sqrt{\pi}}\int_x^{\infty} e^{-t^2/2}dt$. λ_i^* can be solved from (7.6) when P_{Fi} is known. From (7.8), we can see that the detection probability of node i depends on the local false alarm rate and the distance to the target.

In this chapter we assume that the fusion center uses the majority rule[1]. That is, when the number of ones is larger than that of zeros in the local decisions, fusion center decides that a target is present. The system detection probability at location (x, y) (denoted by $P_D(x, y)$) and the false alarm rate (denoted by P_F) can be expressed as follows:

$$
\begin{aligned}
P_D(x,y) &= \sum_{|S_1|>|S_0|} \prod_{i\in S_0}(1-P_{Di}(x,y)) \prod_{j\in S_1} P_{Dj}(x,y) && (7.9)\\
P_F &= \sum_{|S_1|>|S_0|} \prod_{i\in S_0}(1-P_{Fi}) \prod_{j\in S_1} P_{Fj} && (7.10)
\end{aligned}
$$

[1]The LRT (Likelihood Ratio Test) at fusion centers is dependent on the target location, and hence is not applicable here.

where S_0 and S_1 represent the set of nodes whose decisions are zeros and ones, respectively. We can see that the number of addends in (7.9) equals the number of node combinations in which more than half of nodes decide one. Thus the complexity of computing $P_D(x, y)$ is $O(2^n)$ (where n is the number of nodes). According to the definition of detection probability in (7.9), the minimal detection probability in region A is:

$$P_{Dmin} = \min_{(x,y)\in A} \sum_{|S_1|>|S_0|} \prod_{i\in S_0} (1 - P_{Di}(x, y)) \prod_{j\in S_1} P_{Dj}(x, y) \tag{7.11}$$

We assume that all nodes have the same local false alarm rate α_0. From (7.13) and (7.10), we have:

$$\sum_{|S_1|>|S_0|} \prod_{i\in S_0} (1 - \alpha_0) \prod_{j\in S_1} \alpha_0 \leq \alpha \tag{7.12}$$

Since α and the number of nodes are known, the maximal value of local false alarm rate at each node can be solved from (7.12) and will be used by each node in order to achieve the highest system detection probability $P_D(x, y)$. Then the *decision threshold* λ_i^* on the measurement of node i $(1 \leq i \leq n)$ can be solved from (7.6).

7.2 Problem Formulation

In this chapter, we define *coverage* of a sensor network based on a probabilistic detection model. A point p is *covered* by a sensor network if the probability that a target, located at p, is detected by the active nodes is above threshold β and the system false alarm rate is below threshold α. A geographic region is covered by a sensor network

if *all* the points in this region are covered. Formally, the coverage requirement of a region A is defined as:

$$(\forall(x,y) \in A, P_D(x,y) \geq \beta) \wedge (P_F \leq \alpha) \tag{7.13}$$

where $P_D(x,y)$ and P_F represent the detection probability of a target located at (x,y) [2] and the system false alarm rate, respectively.

This probabilistic coverage formulation captures the requirements of many detection-based applications in sensor networks. In addition, it is also useful for other types of sensing applications. For example, a coverage maintenance protocol based on this detection model can be used in a surveillance application. The network can execute the protocol to maintain sufficient detection probability. Once a target is detected, the sleeping nodes are woken up to execute more sophisticated sensing tasks such as intruder tracking.

In this chapter, we focus on the design of distributed network protocols that can provide the required coverage over a region by activating a small number of nodes within a short time. Note that while this thesis does not focus on the design of data fusion algorithms, our coverage maintenance protocols can be extended to incorporate different data fusion algorithms.

[2]For convenience, $P_D(x,y)$ is referred to as the *detection probability at* (x,y) hereafter.

7.3 Design of Coverage Maintenance Protocols

In this section we present the design of three coverage maintenance protocols. The first protocol, *Central*, employs a centralized algorithm that treats the whole region as a single fusion group. To reduce the coverage configuration time, we further design two distributed algorithms based on "virtual grid"s. The *Separate Grid (Se-Grid)* protocol divides the region into separate grids and all nodes in each grid form a fusion group. Fusion centers perform coverage configuration for their respective grids independently of each other. The *Coordinating Grid (Co-Grid)* protocol organizes the region into overlapping grids that coordinate with each other to achieve coverage.

7.3.1 Centralized Coverage Maintenance Protocol

In the Central protocol one node is elected among all nodes in the region A to serve as the fusion center. In the coverage configuration phase, the fusion center decides which nodes should remain active and compute their local false alarm rate such that the coverage requirement (7.13) is met. Initially all nodes are marked as *sleep* by the fusion center. In each iteration of the algorithm, a node is marked as *active*. Given the system false alarm rate threshold α and the number of current active nodes, the fusion center computes a local false alarm rate for active nodes by (7.12). Using the active nodes' locations and the local false alarm rate, the fusion center finds the location (x_{min}, y_{min}) in region A that has the minimal detection probability P_{Dmin}. If P_{Dmin} is less than β, the fusion center finds the node closest to point (x_{min}, y_{min}) among all sleeping nodes and marks it as *active*. This process repeats until the

minimal detection probability P_{Dmin} in region A is greater than β. Then, the fusion center sends a list of active node IDs and the local false rate it computed to all nodes in region A. If a node finds its ID in the list, it remains active and sets its *decision threshold* according to the local false alarm rate using (7.6). Otherwise it goes to sleep and wakes up periodically to check whether it should activate itself (by listening to messages from the fusion center). The pseudo code for the coverage configuration algorithm of Central is shown as follows:

/*α and β are the thresholds on the system false alarm rate
and detection probability, respectively*/
Central(α,β)
begin
Initialize table *actNodes* that stores the IDs and
locations of active nodes to $\emptyset$;
$P_{Dmin} = 0$;
Randomly pick a sleeping node and put in *actNodes*;
/*Initialize the local false alarm rate to the system
false alarm rate*/
$\alpha_0 = \alpha$;
while ($P_{Dmin} < \beta$)
if all nodes in $G(i, j)$ are *active*
return FAILURE;
fi
/*If more than one point has detection probability
P_{Dmin}, randomly pick one as (x_{min}, y_{min})*/
Find the point $(x_{min}, y_{min}) \in A$ that has minimal
detection probability P_{Dmin} based on α_0 and the
active node locations in *actNodes* according to
(7.8) and (7.11);
Put the sleeping node closest to point (x_{min}, y_{min})

```
        in actNodes;
        /*Adjust the local false alarm rate to satisfy the
        constraint on the system false alarm rate*/
        Re-compute α0 based on α using (7.12);
    end
    broadcast the active node table actNodes and α0
    to all nodes;
end
```

In each iteration of Central, the fusion center computes the minimal detection probability P_{Dmin} in the region. From (7.11), we can see that the optimal solution of P_{Dmin} is computationally difficult to obtain and only the numerical solution exists. We compute the approximate solution of P_{Dmin} as follows. The region is divided into a matrix of small square patches and the target is assumed to only appear at the corners of the patches (referred to as *sample points*). The detection probability associated with each sample point is then computed using (7.9) and the minimum detection probability is obtained. As discussed in Section 7.1.2, the complexity of computing the detection probability of a point is $O(2^n)$. Central may incur high computational cost and unacceptable coverage configuration time when the number of active nodes is large. Furthermore, distant nodes make irrelevant detection decisions due to signal decay. Consequently, fusing the decisions from all active nodes in the region may not improve the overall detection performance. We evaluate this effect experimentally in Section 6.3.

7.3.2 Coverage Maintenance Protocol based on Separate Grids

To facilitate parallel processing, we propose the Se-Grid protocol. In Se-Grid, the deployment region of a sensor network is divided into a matrix of identical grids. Each grid is labeled as $G(i, j)$ where (i, j) is the grid index. No grids overlap with each other. The nodes in each grid form a fusion group that executes the Central protocol within its own grid. That is, each fusion group is responsible for covering its own grid by activating nodes within the grid.

Se-Grid can effectively reduce the configuration time because the grids in the region are configured in parallel. However, since Se-Grid restricts decision fusion within each grid, a node cannot contribute to the decision fusion of a neighboring fusion group even if it is close to the grid's boundary. Furthermore, the nodes close to grid boundary are more likely to be activated. This is because the detection probability of nodes decreases quickly with distance, and hence the vicinity of grid boundary usually has lower probability than other locations. Therefore, Se-Grid may activate redundant nodes on both sides of a grid boundary.

7.3.3 Coverage Maintenance Protocol with Inter-grid Coordination

Since the problem of Se-Grid is mainly caused by the lack of collaboration among adjacent fusion centers, we design the Co-Grid protocol that provides efficient inter-grid coordination. In Co-Grid, the network deployment region consists of overlapping grids. Each grid is composed of four identical sub-grids and each sub-grid belongs to

up to four grids. Figure 7.1 illustrates nine overlapping grids composed of 16 sub-grids. The fusion center of each grid is located at the center of the grid and denoted by a small black circle. Each sub-grid is labeled as $s(i)$ and each grid is labeled as $G(i, j)$. For example, grid $G(1, 2)$ consists of four sub-grids $s(5)$, $s(6)$, $s(9)$ and $s(10)$. $s(10)$ (shaded in Figure 7.1) belongs to four overlapping grids $G(1, 2)$, $G(1, 3)$, $G(2, 2)$, $G(2, 3)$. We say two fusion centers are *adjacent* if their grids overlap, i.e., share one or more sub-grids. A fusion center can have up to eight adjacent fusion centers. Since each node belongs to multiple grids, it can contribute to the detection at multiple fusion centers around it. Therefore this algorithm can potentially result in less active nodes by reducing the redundancy in neighboring grids.

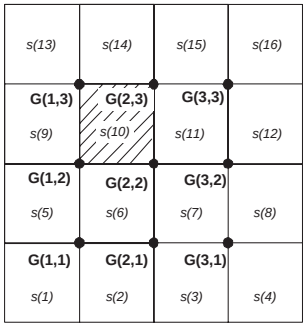


Figure 7.1: Overlapping Grid Layout

In each iteration of Co-Grid, similar to Se-Grid, a fusion center computes the detection probability of each sample point in a grid and activates the node closest to the sample point with the minimum detection probability until the minimum detection probability in the grid is above threshold β. However, the overlapping grid layout introduces additional complexity in the design due to the dependency among overlapping grids. Since each sample point p in the region belongs to up to four grids (referred to as the *master grids* of point p), we only need to guarantee that the *joint* detection probability from p's master grids is no lower than β. However, the joint

detection probability is difficult to compute because the master grids of p share nodes with each other and hence their detection probabilities are not independent. Instead of enforcing the joint probability, Co-Grid guarantees the coverage of point p by ensuring that at least one of its master grids has a detection probability at p that is no lower than β. That is, Co-Grid uses the maximum of the detection probabilities computed by all the master grids of p as an approximation to the joint detection probability at p. For example, to compute the detection probability of a sample point in sub-grid $s(10)$ in Figure 7.1, the fusion center of $G(2,2)$ computes the maximum of the point's detection probabilities computed by the fusion centers in $G(1,2)$, $G(1,3)$ and $G(2,3)$.

To find the point with the minimum detection probability in a grid, a fusion center running Co-Grid needs to compute every sampling point's detection probability, which is the maximum of all detection probabilities computed by the point's mater grids. To reduce the inter-grid communication, instead of communicating detection probabilities of sample points among adjacent fusion centers, we let each fusion center compute the detection probabilities on behalf of its adjacent fusion centers. A fusion center keeps a local false alarm rate and a list of locations of active nodes for each adjacent fusion center. The procedure performed by the fusion center in grid $G(i,j)$ to compute the minimal detection probability $P_{Dmin}(i,j)$ of grid $G(i,j)$ can be formulated as follows:

$$P_{Dmin}(i,j) = \min_{(x,y)\in G(i,j)} \max_{(x,y)\in G(m,n)} P_D(x,y,m,n) \tag{7.14}$$

where $P_D(x, y, m, n)$ represents the sample point (x, y)'s detection probability in its master grid $G(m, n)$ ($G(m, n)$ is an adjacent grid of $G(i, j)$). Note each sample point has up to four master grids and all sample points in a sub-grid share the same master grids. $P_D(x, y, m, n)$ is computed according to (7.9) and (7.12) using the locations of active nodes and the local false alarm rate of grid $G(m, n)$.

At runtime, when the fusion center in $G(i, j)$ activates a node in its grid, the local false alarm rate of active nodes in $G(i, j)$ needs to be re-computed using (7.12) to satisfy the constraint on the system false alarm rate. This may potentially result in changes in the detection probabilities of all sample points in the grid. Due to sharing of sub-grids, up to eight adjacent fusion centers need to know the updated local alarm rate and locations of active nodes in $G(i, j)$ before they can activate any new node. Therefore, activating a new node in a grid may invalidate the ongoing processes of activating any other new node in its adjacent grids. To resolve the contention among adjacent grids, we consider the following two approaches. 1) The fusion center in $G(i, j)$ notifies its adjacent fusion centers both before it starts and after it completes the process of finding a new active node. Because the location of the new active node is unknown before the process completes, all the adjacent fusion centers have to wait until they receive the result from the fusion center in $G(i, j)$. While this "locking" strategy sequentializes all the computations of adjacent fusion centers, it is pessimistic and may unnecessarily reduce the efficiency of Co-Grid (detailed analysis is presented in Section 7.4). 2) Each fusion center performs the process of activating new nodes independently. Whenever the fusion center in $G(i, j)$ activates a new node, it advertises the local false alarm rate and the locations of current active nodes to its adjacent fusion centers, which cancel their current computations and restart with

the updated parameters received from the fusion center of $G(i, j)$. This approach maximizes the parallelism of adjacent fusion centers and is adopted by Co-Grid.

The pseudo-code of the Co-Grid protocol at fusion center $G(i, j)$ is shown as follows:

```
/*Global definitions*/
struct Grid {
    /*IDs and locations of the active nodes in the grid*/
    list actNodes;
    /*Local false alarm rate of the active nodes*/
    double α0;}
/*Information of grid G(i,j)*/
struct Grid self;
/*Information of adjacent grids*/
struct Grid adjGrids[8];
/*α and β are the thresholds on the system false
alarm rate and detection probability*/
ActivateNode(α,β)
begin
    if all nodes in G(i,j) are active
        return FAILURE;
    fi
    if  self.actNodes is empty
        Randomly pick a sleeping node and put
        in self.actNodes;
    fi
    do
        /*Adjust the local false alarm rate α0 to satisfy
        the constraint on the system false alarm rate*/
        Compute self.α0 using self.actNodes
        according to (7.12);
```

/*If more than one point has detection probability
P_{Dmin}, randomly pick one as (x_{min}, y_{min})*/
Find the point $(x_{min}, y_{min}) \in G(i,j)$ that has the
detection probability P_{Dmin} according to (7.14)
using *self* and *adjGrids*;
if $P_{Dmin} \geq \beta$ **return** *SUCCESS*;
Put the sleeping node closest to point (x_{min}, y_{min})
in *self.actNodes*;
/*Advertise the change of state*/
send an *Update* message including $self.\alpha_0$ and
$self.actNodes$ to all adjacent fusion centers;
while($P_{Dmin} < \beta$)
return *SUCCESS*;
end
/* The following function is called whenever the
fusion center receives an *Update* message from
an adjacent fusion center. */
UpdateParameters
begin
Stop the execution of *ActivateNode*;
Update *adjGrids* array with received information;
Call ActivateNode(α,β);
end

After the process of coverage maintenance completes, each fusion center sends the list of active nodes and the local false alarm rate it computed to all nodes in its grids. If a node finds itself in the list of active nodes, it remains active and sets a decision threshold according to the local false alarm rate it received (see (7.6)). Since each node belongs to up to four master grids, an active node may have up to four

decision thresholds. During detection phase, an active node periodically compares its measurement with each decision threshold and sends a decision (0 or 1) to the corresponding fusion center. Hence an active node needs to send up to four decision messages in each detection period.

7.4 Analysis of the Degree of Parallel Configuration

In Se-Grid, each fusion center can turn on new nodes independently from each other. However, this is not the case for Co-Grid due to the inter-dependencies among overlapping grids. As described in Section 7.3.3, when a fusion center adds a new active node, all the adjacent fusion centers have to re-start their computation from scratch. We define *effective computation* as the computation in a fusion center that will lead to the addition of a new active node. Any two adjacent fusion centers cannot perform effective computation at the same time. We define the *degree of parallel configuration (DPC)* as the total number of fusion centers that can perform effective computations simultaneously in the whole network. Clearly DPC has a significant impact on the coverage configuration time of the whole network. The DPC of Se-Grid is the total number of grids because all the grids can configure themselves in parallel. In contrast, the DPC of Central is only 1. However, it is less straightforward to quantify the parallelism of Co-Grid due to the inter-grid dependencies.

In order to understand the cost of inter-grid coordination on configuration time, we now analyze the degree of parallel configuration under Co-Grid. We model the network as a graph (referred to as *parallelism graph*) where each fusion center is a vertex

and an edge exists between two adjacent fusion centers to represent the fact that any two adjacent fusion centers cannot perform effective computation simultaneously. The fusion centers that can proceed simultaneously form an *independent set*[3] of the parallelism graph. *Maximal independent set* of a graph is a subset of vertices such that there is no edge between any pair of vertices in the set and no more vertices can be added without making it a non-independent set. It is clear that the best-case and worst-case DPCs are equal to the maximal and minimal cardinality of the maximal independent sets of the parallelism graph, respectively.

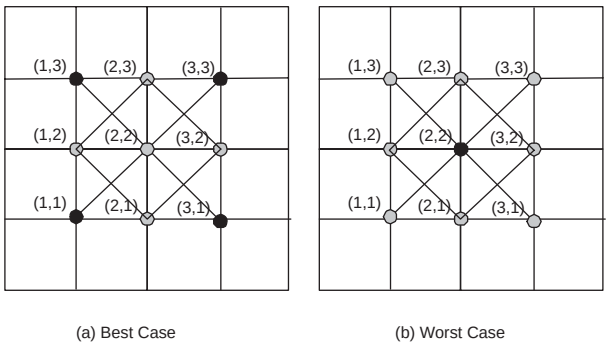


Figure 7.2: An example of degree of Parallel Configuration.

Figure 7.2 shows two possible states of the network at configuration time. The network is composed of 9 overlapping grids. One fusion center is located at each grid center. Figure 7.2(a) shows the maximal possible degree of parallel configuration where each of the four fusion centers denoted by black circles can activate a node simultaneously. That is, in the best case the DPC of the network is 4 under Co-Grid. On the other hand, Figure 7.2(b) shows the worst-case DPC is 1 where only fusion center $(2, 2)$ activates a node while all other fusion centers cannot proceed.

The best-case DPC of Co-Grid equals the maximal number of grids that do not overlap in the region and can be as good as Se-Grid. Figure 7.3 shows the worst-case DPC of

[3]An independent set of a graph is a subset of the vertices such that no any two vertices in the subset is connected by an edge.

Co-Grid and the DPC of Se-Grid under different grid width in a 120×120 m^2 region. The worst-case DPC of Co-Grid is obtained by computing the minimal cardinality of the maximal independent sets in the corresponding parallelism graph.

When the network size approaches infinite, it can be shown that the worst-case and best-case DPCs of Co-Grid are $n/9$ and $n/4$, respectively, where n is the total number of grids. Since Co-Grid has four times as many grids as Se-Grid for the same grid size, the lower-bound on the ratio between the DPCs of Co-Grid and Se-Grid approaches 4/9 for large regions. This result indicates that the DPC of Co-Grid increases about proportionally with the number of grids and hence can scale well in large networks in term of configuration time.

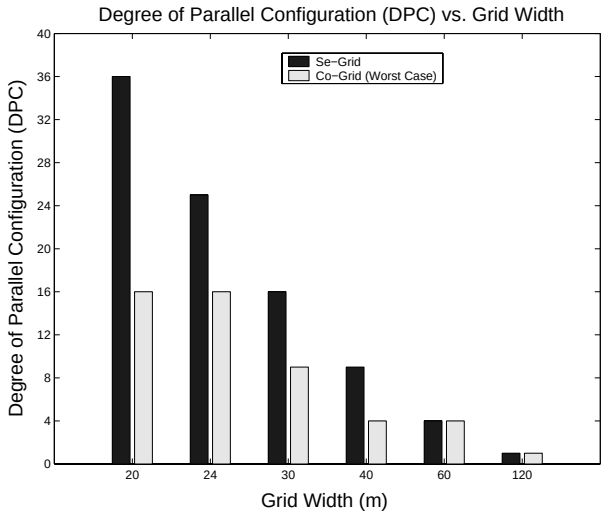


Figure 7.3: Degree of Parallel Configuration vs. grid width.

7.5 Performance Evaluation

In this section we present the simulation results using Matlab. In addition to our protocols presented in Section 7.3, we implemented a protocol called Random as the baseline for performance comparison. Random works similarly to Se-Grid except

that a fusion center always *randomly* activates a new node in each iteration until the desired detection probability is achieved. Comparing our protocols against Random allows us to study the effectiveness of the greedy strategy that activates the node closest to the point with minimum detection probability.

In each experiment, 2000 nodes are randomly deployed in a 120×120 m^2 region. The thresholds on the system detection probability and false alarm rate are 90% and 5% respectively. All results in this section are averages of five runs. The attenuation factors in the signal decay model (see (7.1)) are set to $b = 1$, $a = 2$ and the initial signal power of the target (e_0) is set to 200. The variance σ of the noise and node measurement distribution is set to 1.

Since the complexity of computing the detection probability of a point using (7.9) is $O(2^n)$ (n is the number of active nodes), the simulations are extremely time consuming when the number of active nodes is large.

Chapter 8

Conclusions and Future Work

In this thesis we propose a unified power management approach that contributes the state-of-the-art of power management in WSNs in several important aspects.

We first conduct theoretical analysis of the fundamental relationship between sensing coverage and network connectivity. Specifically, our analytical results show that K-coverage implies K-connectivity when the double range property (i.e., the communication range is at least twice the sensing range) holds. We then analyze the impact of sensing coverage on the performance of greedy geographic routing algorithms. We prove that simple greedy geographic routing algorithms such as Greedy Forwarding and Bounded Voronoi Greedy Forwarding may be highly efficient in sensing-covered networks with deterministic or probabilistic communication links. This result indicates that the redundant nodes can be turned off without significant increase in network length as long as the remaining active nodes maintain sensing coverage. Therefore, our analysis justifies the coverage maintenance protocols [61, 71, 75, 76] that conserve energy by scheduling redundant nodes to sleep. Moreover, we derive lower bounds on the lengths of the network routes found by GF and BVGF. These bounds enable a source node to efficiently compute an upper-bound on the network

length or expected number of transmissions of its routing path based on the location of the destination. This capability can be useful to real-time communication protocols that require such bounds to achieve predictable end-to-end communication delays.

Our analyses yield key insights for treating sensing and communication within a unified power management framework, which is in sharp contrast to several existing approaches that address the two problems in isolation. Specifically, given the QoS requirements of K_s-coverage and K_c-connectivity, a power management protocol only needs to satisfy $max(K_s, K_c)$-coverage when the double range property holds. Moreover, simple greedy geographic routing algorithms have analytical performance guarantees on sensing-covered networks.

We then develop a novel radio power management approach called *Minimum Power Configuration (MPC)*. In contrast to the existing approaches that treat different radio states (transmission/reception/idle) in isolation, MPC integrates them in a unified optimization framework that considers both the set of active nodes and their transmission power in a network. We have presented a set of approximation algorithms with provable performance bounds, and the practical MPCP protocol that dynamically (re)configures a network based on current data rates. We also proposed a more efficient protocol called MASP that only minimizes the total number of active nodes in a network. Simulations based on a realistic radio model of Mica2 motes show that MPCP can conserve significantly more energy than representative topology control and power-aware routing schemes. Furthermore, while MASP is suitable for radios with high idle power, a key advantage of MPCP is that it yields satisfactory performance under a range of representative radio characteristics, allowing it to flexibly adapt to different radio platforms.

Finally, we develop two power management schemes that account for realistic sensing and communication properties (e.g., asymmetric and lossy communication links and probabilistic sensing range) of WSNs. These works is an important step toward providing robust sensing and communication performance for real-world sensor network applications.

Configurable Topology Control (CTC) is a topology control protocol designed for lossy WSNs. The key novelty of CTC lies in its capacity of configuring a network topology to achieve desired path quality bounds in a localized fashion. We have also presented four CTC algorithms that combine per-node/per-link power control with two metrics for power assignment. Simulations based on realistic characteristics of Mica2 motes show that CTC can provide desired tradeoff between power consumption and network performance according to application requirements. Furthermore, CTC outperforms an existing topology control algorithm, LMST, in terms of both communication performance and energy consumption. Our results demonstrate the significant impact of realistic link models on power management for WSNs.

We have developed a new coverage maintenance protocol called Co-Grid that maintains probabilistic detection guarantees over a region. Co-Grid is designed based on a distributed detection model that considers data fusion among multiple nodes. This distinguishes Co-Grid from existing protocols that are based on simpler detection models. Furthermore, Co-Grid can meet sensing QoS (e.g., maximum detection probability and minimum false alarm rate) that are more consistent with the requirements of real-world event monitoring applications. Our theoretical analysis and simulation results demonstrate that Co-Grid not only competes well against the centralized protocol in terms of the number of active nodes, but also consistently outperforms the protocol based on separate grids in terms of the configuration time.

The presented unified power management approach suggests several interesting future work and research directions. This thesis mainly focuses on minimizing the total energy consumption of a network. One interesting direction for future research is to extend the proposed approach to maximize the system lifetime of WSNs. The definition of system lifetime is specific to application scenarios. For example, the lifetime of a data collection network may be defined as the time duration in which certain throughput can be constantly maintained at base station, while a surveillance network may consider the loss of sensing coverage to be the end of system lifetime. Based on the former lifetime definition, we will extend MPCP to incorporate appropriate routing metrics (e.g., those based on node residual energy) to achieve more balanced energy dissipation and prolong network lifetime [56, 39]. Similarly, Co-Grid can be extended to meet the coverage-based lifetime requirement by balancing the energy consumption of different nodes in sleep schedule.

Another direction is to extend our analyses of the relationship between sensing and communication to more realistic radio and sensing models. To account for asymmetric and loss communication links, we have adopted a generic graph model in the design of the CTC protocol where each link is associated with a communication direction and the packet reception ratio in this direction. In contrast to the widely adopted disc sensing model, Co-Grid is designed based on a probabilistic sensing model that accounts for signal decay and interference from environmental noise. More sophisticated sensing models can be derived from established distributed signal processing theories or empirically estimated at run-time. We will incorporate these realistic models into a unified power management framework for WSNs. Such a framework will provide a sound foundation for designing and analyzing energy conservation protocols for real-world sensing applications.

This thesis work focuses on the power management of radios. We plan to integrate our work with the power management of other components (e.g., sensors, CPU, and Flash) on WSN platforms. Although the issue of power management for each of these components has been extensively studied, a holistic system approach to power management still does not exist. In particular, the complex interactions between the power management of different components require careful analysis and may lead to interesting system solutions.

References

[1] Daniel Aguayo, John C. Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh network. In *SIGCOMM*, 2004.

[2] Falchi Alessio. Sensor networks: performance measurements with motes technology. Technical report, University of Pisa, Italy, 2004.

[3] Khaled Alzoubi, Xiang-Yang Li, Yu Wang, Peng-Jun Wan, and Ophir Frieder. Geometric spanners for wireless ad hoc networks. *IEEE Transactions On Parallel And Distributed System*, 14, May 2003.

[4] Franz Aurenhammer. Voronoi diagrams -a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[5] D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall, 1987.

[6] Bogdan Carbunar, Ananth Grama, Jan Vitek, and Octavian Carbunar. Redundancy and coverage detection in sensor networks. *ACM Transactions on Sensor Networks*, 2(1), 2006.

[7] Alberto Cerpa and Deborah Estrin. Ascent: Adaptive self-configuring sensor networks topologies. In *INFOCOM*, 2002.

[8] Alberto Cerpa, Jennifer L. Wong, Louane Kuang, Miodrag Potkonjak, and Deborah Estrin. Statistical model of lossy links in wireless sensor networks. In *IPSN*, 2005.

[9] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Transactions on Computers*, 51(12):1448–1453, December 2002.

[10] Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad hoc networks. In *INFOCOM*, 2000.

[11] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *MobiCom*, 2001.

[12] Octav Chipara, Zhimin He, Guoliang Xing, Qin Chen, Xiaorui Wang, Chenyang Lu, John Stankovic, and Tarek Abdelzaher. Real-time power control in wireless sensor networks. Technical Report WUCSE-2005-31, Washington University in St.Louis, 2005.

[13] T. Couqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja. Sensor deployment strategy for target detection. In *Proceeding of The First ACM International Workshop on Wireless Sensor Networks and Applications*, pages 169–177, Sep 2002.

[14] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom*, 2003.

[15] Crossbow. Mica and mica2 wireless measurement system datasheets. 2003.

[16] M Desrochers and F. Soumis. A generalized permanent labeling algorithm for the shortest path problem with time windows. *INFOR.*, 26:191-212, 1988.

[17] Qunfeng Dong, Suman Banerjee, Micah Adler, and Archan Misra. Minimum energy reliable paths using unreliable wireless links. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005.

[18] Sheetalkumar Doshi, Shweta Bhandare, and Timothy X Brown. An on-demand minimum energy routing protocol for a wireless ad hoc network. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(3):50–66, 2002.

[19] Sheetalkumar Doshi and Timothy X Brown. Minimum energy routing schemes for a wireless ad hoc network. In *INFOCOM*, 2002.

[20] Richard Draves, Jitendra Padhye, and Brian Zill. Comparison of routing metrics for static multi-hop wireless networks. In *SIGCOMM*, pages 133–144, 2004.

[21] G.G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI Research Report ISU/RR-87-180, Inst. for Scientific Information, Mar, 1987.

[22] Jie Gao, Leonidas J. Guibas, John Hershberger, Li Zhang, and An Zhu. Geometric spanner for routing in mobile networks. In *Proc. 2nd ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 45–55, October 2001.

[23] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[24] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM J. Appl. Math.*, 16:1-29, 1968.

[25] MohammadTaghi Hajiaghayi, Nicole Immorlica, and Vahab S. Mirrokni. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. In *MobiCom*, 2003.

[26] Tian He, Sudha Krishnamurthy, John A. Stankovic, Tarek Abdelzaher, Liqian Luo, Radu Stoleru, Ting Yan, Lin Gu, Jonathan Hui, and Bruce Krogh. Energy-efficient surveillance system using wireless sensor networks. In *Mobisys*, 2004.

[27] IEEE. Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Standard 802.11*, 1999.

[28] M. Imase and B. Waxman. Dynamic steiner tree problem. *SlAM J. Discrete Math.*, 4:3:369–384, 1991.

[29] Brad Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *MobiCom*, pages 243–254, 2000.

[30] Vikas Kawadia and P. R. Kumar. Power control and clustering in ad hoc networks. In *INFOCOM*, 2003.

[31] David Kotz, Calvin Newport, Robert S. Gray, Jason Liu, Yougu Yuan, and Chip Elliott. Experimental evaluation of wireless simulation assumptions. In *MSWiM*, October 2004.

[32] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society.*, 7:48-50, 1956.

[33] Fabian Kuhn, Roger Wattenhofer, and Aaron Zollinger. Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing. In *Proc. 4^{th} ACM Int. Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc)*, 2003.

[34] Johnson Kuruvila, Amiya Nayak, and Ivan Stojmenovic. Hop count optimal position based packet routing algorithms for ad hoc wireless networks with a realistic physical layer. In *1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2004.

[35] Dan Li, Kerry Wong, Yu Hen Hu, and Akbar Sayeed. Detection, classification and tracking of targets in distributed sensor networks. *IEEE Signal Processing Magazine*, 19 (2), 2002.

[36] Li Li, Joseph Y. Halpern, Paramvir Bahl, Yi-Min Wang, and Roger Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, 2001.

[37] Ning Li and Jennifer C. Hou. Flss: a fault-tolerant topology control algorithm for wireless networks. In *MobiCom*, 2004.

[38] Ning Li, Jennifer C. Hou, and Lui Sha. Design and analysis of an mst-based topology control algorithm. In *INFOCOM*, 2003.

[39] Qun Li, Javed Aslam, and Daniela Rus. Online power-aware routing in wireless ad-hoc networks. In *MobiCom*, 2001.

[40] Xiang-Yang Li, Peng-Jun Wan, Yu Wang, and Chih-Wei Yi. Fault tolerant deployment and topology control in wireless networks. In *MobiHoc*, 2003.

[41] Gang Lu, Narayanan Sadagopan, Bhaskar Krishnamachari, and Ashish Goel. Delay efficient sleep scheduling in wireless sensor networks. In *IEEE INFOCOM*, 2005.

[42] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *WSNA*, pages 88–97, 2002.

[43] S. Meguerdichian and M. Potkonjak. Low power 0/1 coverage and scheduling techniques in sensor networks. Technical Report Technical Reports 030001, January 2003.

[44] A. Meyerson, K. Munagala, and S. Plotkin. Cost-distance: two metric network design. In *FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, 2000.

[45] Swetha Narayanaswamy, Vikas Kawadia, R. S. Sreenivas, and P. R. Kumar. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol. In *European Wireless Conference*, 2002.

[46] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *SIGCOMM: Proceedings of the conference on Communications architectures, protocols and applications*, 1994.

[47] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *SenSys*, 2004.

[48] Ram Ramanathan and Regina Hain. Topology control of multihop wireless networks using transmit power adjustment. In *INFOCOM*, 2000.

[49] Gabriel Robins and Alexander Zelikovsky. Improved steiner tree approximation in graphs. In *SODA*, 2000.

[50] Volkan Rodoplu and Teresa H. Meng. Minimum energy mobile wireless networks. *IEEE J. Selected Areas in Communications*, 17(8), 1999.

[51] Arvind Sankar and Zhen Liu. Maximum lifetime routing in wireless ad-hoc networks. In *INFOCOM*, 2004.

[52] P. Santi. Topology control in wireless ad hoc and sensor networks. Technical Report IIT-TR-04, Istituto di Informatica e Telematica, Pisa - Italy, 2003.

[53] Karim Seada, Marco Zuniga, Ahmed Helmy, and Bhaskar Krishnamachari. Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. In *SenSys: Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004.

[54] Victor Shnayder, Mark Hempstead, Bor rong Chen, Geoff Werner Allen, and Matt Welsh. Simulating the power consumption of large-scale sensor network applications. In *SenSys*, 2004.

[55] G. Simon. Probabilistic wireless network simulator. http://www.isis.vanderbilt.edu/projects/nest/prowler/.

[56] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, 1998.

[57] Dongjin Son, Bhaskar Krishnamachari, and John Heidemann. Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks. In *SECON*, 2004.

[58] Ivan Stojmenovic and Xu Lin. Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1023–1032, 2001.

[59] Robert Szewczyk, Alan Mainwaring, Joseph Polastre, John Anderson, and David Culler. An analysis of a large scale habitat monitoring application. In *SenSys*, 2004.

[60] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.

[61] Di Tian and Nicolas D. Georganas. A coverage-preserved node scheduling scheme for large wireless sensor networks. In *Proceedings of First International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, pages 169–177, Atlanta, USA, Sep 2002.

[62] J.N. Tsitsiklis and M. Athans. On the complexity of decentralized decision making and detection problems. *IEEE Transactions on Automatic Control*, May 1985.

[63] Tijs van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Sensys*, 2003.

[64] P. Varshney. *Distributed Detection and Data Fusion*. Spinger-Verlag, New York, NY, 1996.

[65] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher D. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Sensys*, 2003.

[66] Mirjam Wattenhofer and Roger Wattenhofer. Distributed weighted matching. In *18th Annual Conference on Distributed Computing (DISC)*, 2004.

[67] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys*, 2003.

[68] Guoliang Xing, Chenyang Lu, Robert Pless, and Qingfeng Huang. On greedy geographic routing algorithms in sensing-covered networks. In *Proc. Fifth ACM Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Tokyo, Japan, May 2004.

[69] Guoliang Xing, Chenyang Lu, Robert Pless, and Qingfeng Huang. Impact of sensing coverage on greedy geographic routing algorithms. *IEEE Transactions on Parallel and Distributed Systems*, 17(4), 2006.

[70] Guoliang Xing, Chenyang Lu, Robert Pless, and Joseph A. O'Sullivan. Co-grid: An efficient coverage maintenance protocol for distributed sensor networks. In *The 3rd International Symposium on Information Processing in Sensor Networks (IPSN)*, 2004.

[71] Guoliang Xing, Xiaorui Wang, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Senor Networks*, 1(1):36–72, 2005.

[72] Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindan, and Deborah Estrin. A wireless sensor network for structural monitoring. In *SenSys*, 2004.

[73] Ya Xu, John Heidemann, and Deborah Estrin. Adaptive energy-conserving routing for multihop ad hoc networks. Research Report 527, USC, October 2000.

[74] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *MobiCom*, 2001.

[75] Ting Yan, Tian He, and John A. Stankovic. Differentiated surveillance for sensor networks. In *The First ACM Conference on Embedded Networked Sensor Systems(Sensys 03)*, 2003.

[76] F. Ye, G. Zhong, S. Lu, and L. Zhang. Peas: A robust energy conserving protocol for long-lived sensor networks. In *The 23rd International Conference on Distributed Computing Systems (ICDCS'03)*, pages 169–177, May 2003.

[77] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *INFOCOM*, 2002.

[78] Wei Ye, John Heidemann, and Deborah Estrin. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, June 2004.

[79] Honghai Zhang and Jennifer C. Hou. Maintaining coverage and connectivity in large sensor networks. *The Wireless Ad Hoc and Sensor Networks: An International Journal*, 2005.

[80] Ying Zhang. Routing modeling application simulation environment. http://www2.parc.com/spl/projects/era/nest/Rmase/.

[81] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Sensys*, Los Angeles, CA, November 2003.

[82] Rong Zheng, Jennifer C. Hou, and Lui Sha. Asynchronous wakeup for ad hoc networks. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 35–45. ACM Press, 2003.

[83] Rong Zheng and Robin Kravets. On-demand power management for ad hoc networks. In *INFOCOM*, 2003.

[84] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *MobiSys*, 2004.

[85] Marco Zuniga and Bhaskar Krishnamachari. Analyzing the transitional region in low power wireless links. In *First IEEE International Conference on Sensor and Ad hoc Communications and Networks (SECON)*, October 2004.

Guoliang Xing

Date of Birth April 20, 1977

Degrees B.E. Electrical Engineering, Xi'an Jiao Tong University, Xi'an, China, July 1998

M.E. Computer Science, Xi'an Jiao Tong University, Xi'an, China, May 2001

M.S. Computer Science, Washington University in St. Louis, MO, USA, May 2003

D.Sc. Computer Science, Washington University in St. Louis, MO, USA, August 2006

Publications

1. Guoliang Xing, Chenyang Lu, Robert Pless, and Qingfeng Huang. Impact of sensing coverage on greedy geographic routing algorithms. *IEEE Transactions on Parallel and Distributed Systems*, 17(4), 2006.

2. Guoliang Xing, Xiaorui Wang, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Transactions on Senor Networks*, 1(1):36–72, 2005.

3. Guoliang Xing, Chenyang Lu, and Robert Pless. Localized algorithms for configurable topology control in lossy

wireless sensor networks. Technical Report WUCSE-05-36, Washington University in St.Louis, 2005.

4. Octav Chipara, Zhimin He, Guoliang Xing, Qin Chen, Xiaorui Wang, Chenyang Lu, John Stankovic, and Tarek Abdelzaher. Real-time power-aware routing in wireless sensor networks. In *IWQoS*, 2006.

5. Kevin Klues, Guoliang Xing, and Chenyang Lu. Supporting flexible radio power management in wireless sensor networks. Technical Report WUCSE-2006-06, Washington University in St.Louis, 2005.

6. Sangeeta Bhattacharya, Guoliang Xing, Chenyang Lu, Gruia-Catalin Roman, Brandon Harris, and Octav Chipara. Dynamic wake-up and topology maintenance protocols with spatiotemporal guarantees. In *International Conference on Information Processing in Sensor Networks (IPSN)*, 2005.

7. Guoliang Xing, Chenyang Lu, Ying Zhang, Qingfeng Huang, and Robert Pless. Minimum power configuration in wireless sensor networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, 2005.

8. Guoliang Xing, Chenyang Lu, Robert Pless, and Joseph A. O'Sullivan. Co-grid: An efficient coverage maintenance protocol for distributed sensor networks. In *The 3rd*

International Symposium on Information Processing in Sensor Networks (IPSN), 2004.

9. Sangeeta Bhattacharya, Octav Chipara, Brandon Harris, Chenyang Lu, Guoliang Xing, and Chien-Liang Fok. Demo abstract: Mobiquery - a spatiotemporal data service for sensor networks. In *SenSys*, 2004.

10. Chenyang Lu, Guoliang Xing, Octav Chipara, Chien-Liang Fok, and Sangeeta Bhattacharya. A spatiotemporal query service for mobile users in sensor networks. In *ICDCS*, 2005.

11. Guoliang Xing, Chenyang Lu, Robert Pless, and Qingfeng Huang. On greedy geographic routing algorithms in sensing-covered networks. In *Proc. Fifth ACM Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Tokyo, Japan, May 2004.

12. Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher D. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Sensys*, 2003.