Low Duty-Cycled Wireless Sensor Networks: Connectivity and Opportunistic Routing

by

Dongsook Kim

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2008

Doctoral Committee:

Associate Professor Mingyan Liu, Chair
Professor David Neuhoff
Professor Demosthenis Teneketzis
Assistant Professor Zhuoqing M. Mao

To my parents

ACKNOWLEDGEMENTS

I owe enormous gratitude to a lot of people who gave me their endless support during my doctoral courses. Among all, I would like to specially express my gratitude to my advisor, Mingyan Liu, who was a role model to me for all these years and has guided me with her passion and patience to complete this thesis. I also thank my committee members, David Neuhoff, Demosthenis Teneketzis and Zhuoqing M. Mao, for their valuable comments.

I owe tremendous thank to my beloved family, specially my parents who have thrown full support with their love and patience. There have been many friends who spent the same period in Ann Arbor, shared up and down, and encouraged one another. I would like to thank my colleagues among those who came to Ann Arbor in the same year, Taesik Yang, Junho Choi and Sahika for their care and friendship. I also would like to thank my friends and juniors, Jungkeun Yoon, Iljoo Na, Yoonna Oh, Junglan Cho, Jinho Kim, Hyoil Kim, Sangwon Yoon, Sanghyun Seo, Junyoung Park, Jooseok Kim, Gyemin Lee, Christine Kyungmin Kim, Seonghee Son, Heesun Min, Heejoo Kang, Nayoung Shin, Hyewon Yeom, Jean An, Kyongae Lee, Kyongja Kim, Hyunhee Lee, Eunsook Jeon, Myungkil An, Minkyong Lee, Jinhee Han and those whom I didn't mention here, for their help and support nearby or from a distance.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF APPENDICES

Appendix

CHAPTER 1

Introduction

1.1 Introduction and Motivation

Wireless sensor networks used for monitoring and surveillance purposes rely heavily on the efficient use of unattended sensors to detect, identify and track targets in order to enhance situation awareness, agility and survivability. The successful deployment and operation of these sensor fields require low cost transceivers and processors, and a reliable, robust, secure and jam-resistant communication infrastructure to gather and disseminate sensor data. Among different types of sensors, the unattended ground sensors (UGS) are typically deployed and left to self-organize and carry out various sensing, detection and communication tasks. These sensors are operated on battery power, and energy is not always renewable due to cost, environmental and form-size concerns. Therefore in order to ensure that these sensor fields can accomplish planned missions (which may need to last for weeks and months or even more), it is critical to operate these sensors in a highly energy efficient manner. This places a stringent energy constraint on the design of the communication architecture, communication protocols, and the deployment and the operation of these sensors.

It has been observed that low power, low range sensors consume significant amount of energy while idling compared to that consumed during transmission and reception. Consequently, it has been widely considered a principle method of energy conservation to turn off sensors that are actively involved in sensing or communication. By functioning at a low duty cycle, i.e., the fraction of time that a sensor is active/on, sensors are able to conserve energy and consequently increase their lifetime. This is especially applicable in scenarios where sensors are naturally idle for most of the time (e.g., detection of infrequent events such as fire, fault, etc., and transmission of very short messages). In some cases we may also be forced to put sensors in a power-saving (or *sleep*) mode for a large fraction of the time in order to meet a certain lifetime requirement.

By putting sensors into sleep, we obtain prolonged lifetime of a sensor network. The price we pay is that the network communication and sensing capabilities become intermittent since sensors alternate between sleep and wake modes. The intermittent sensing capability disrupts the sensing coverage of the network, i.e., certain area of the network may not be covered by any sensor and events may fail to be detected. Similarly, turning off radio transceivers results in loss of connectivity between nodes. In other words, paths between some nodes may be unavailable from time to time. Therefore, sensors need to establish another path to forward data or wait for the node in the path to wake up. Either choice may lead to prolonged delay. Clearly, there is a trade-off between energy saving and performance degradation.

A central theme of this thesis is to understand this trade-off and design good networking algorithms that will work well with low duty cycles. Within this context, there are a number of ways to mitigate the adverse effect of low duty-cycling. Below we discuss some of these methods and highlight the motivation behind our work.

One way to mitigate performance degradation caused by low duty cycles is by adding redundancy to the initial deployment, i.e., to deploy more sensors (e.g., [1, 14, 41, 51]). Intuitively, the more sensors we deploy, the more we should be able to reduce the duty cycle of each individual sensor while meeting certain performance criteria. One of the main focuses of this study is to investigate the fundamental relationship between the amount of redundancy required vs. the achievable reduction in duty cycle for a fixed performance criterion. We examine this relationship within the context of network asymptotic connectivity, the subject of the first part of this thesis.

A second way to mitigate such performance degradation caused by duty-cycling is to design good algorithms and protocols that effectively deal with the temporal loss of connectivity. One of the most challenging problems is the design of a routing scheme that finds an energy/delay-efficient path from a source to a destination in the presence of duty-cycling as well as unreliable wireless channels [3, 6, 7, 19, 40, 58]. Furthermore, to quantify the performance advantage of such routing algorithms over conventional algorithms is highly nontrivial from an analytical perspective. Typical studies employ numerical simulation. In the second part of this thesis we first investigate the design of such a scheme using an optimal stochastic routing framework. We then investigate how the performance of this class of algorithms scales compared to conventional routing algorithms in a limiting regime where the network becomes dense.

A third method to mitigate performance disruption caused by low duty cycles is to carefully design the sleep schedules of sensors (i.e., determining when and for how long a sensor should be kept off), and preferably in combination with the design of networking algorithms, e.g., a joint design of routing and sleep scheduling. This is conceptually possible because in many sensor network applications sensors are not expected to engage in time-consuming activities. For instance, for intrusion detection, a sensor ideally only needs to be active at the time when an intrusion event occurs and stay active long enough to report such an event. In a routing example, a sensor ideally does not need to be active till a packet is ready for it to relay. Therefore it is possible to coordinate the sleep schedules among sensors and to jointly optimize the determination of sleep schedules and that of network algorithms like routing. This idea, while conceptually appealing and for which heuristics abound (see e.g., [25, 26, 42, 52]), is very difficult from an analytical point of view, and can only be done in a highly application specific way. This is because unless the underlying application is well defined and relatively narrowly scoped, it will be very hard to obtain the set of assumptions needed to shape the type of coordination schemes. Furthermore, coordinated sleep scheduling usually requires much more overhead in coordinating between neighboring nodes [10, 37, 38, 52, 53, 54].

For these reasons, we do not study this type of method in this thesis. Instead, in all the studies contained in this thesis we will focus on a class of *random* sleep schedules, where sensors turn themselves on and off randomly (i.e. each sensor has its own random sleep/wake cycle which is independent of other sensors). The main advantage in using random sleeping lies in its simplicity in implementation and that it makes analysis feasible. Moreover, since coordinated sleeping typically performs better than random sleeping, studying the latter provides a performance lower bound.

The third and last part of this thesis focuses on broadcast algorithms, which are an essential component of any communication protocols, especially routing protocols. Most routing algorithms rely on a underlying broadcasting scheme to handle functionalities like topology discovery and topology dissemination in order to gather information needed to perform routing. Broadcast algorithms have been widely studied in the context of wireless ad hoc and sensor networks where frequent topological changes occur due to either nodes' mobility as well as duty-cycling [31, 49, 56, 32, 43, 21, 45, 17, 2]. However, to date there exists no comprehensive yet computationally efficient mathematical framework to evaluate and compare the performance of competing broadcasting strategies. In the last part of this thesis we develop an analysis-emulation hybrid model that combines analytical models with elements of numerical simulation to obtain the desired modeling accuracy and computational efficiency.

With the studies listed above, we address a number of design issues that arise in a low duty-cycled wireless sensor network, from understanding the performance impact, to building efficient networking algorithms. In the remainder of this chapter, we will first review related works on wireless sensor networks, and then outline individual research problems studied in this thesis, the methodologies used and our main contributions.

1.2 Outline of the Dissertation and Main Contributions

This dissertation consists of three parts: connectivity, routing, and broadcasting. The first part of this dissertation consists of an analysis of some fundamental aspects of duty-cycling in the context of network connectivity. Under the assumption of random duty-cycling, Chapter 2 derives the relationship between node density, transmission power, and the percentage of time nodes are in the sleep mode (the duty cycles). The second part of this dissertation deals with routing and consists of Chapter 3 and Chapter 4. In Chapter 3 we analyze the structure of optimal routing policies, shown to be event-dependent, for a low duty-cycled sensor network and develop a distributed routing algorithm. In Chapter 4 we compare this class of event-dependent routing algorithms (also referred to as opportunistic routing) to the class of traditional non-event dependent routing in terms of routing delay in an asymptotic setting. The last part of the dissertation includes Chapter 5 which provides a numerical framework to evaluate the performance of broadcast schemes considering transmission failures.

Throughout this dissertation, a challenge we repeatedly face is the randomness induced by duty-cycling: the network topology and connectivity, even when the nodes are static, become time-varying because of duty-cycling. While randomness is a familiar concept to the wireless communications and network research community, it typically arises from the uncertainly associated with the communication channel[1]. This uncertainty is primarily reflected in the outcome of packet transmission. On the other hand, the uncertainty induced by duty-cycling is reflected in the dynamic topology and node availability. There is a difference between these two sources of randomness. With the former, we can attempt to reduce the randomness in transmission outcome by measuring the channel quality, but we cannot completely remove it. We will not know for sure whether a packet transmission is successfully received till it happens. However, with the latter, we may be able to obtain precise information on the (local) topology via messaging passing, thereby finding out whether certain nodes are asleep and completely removing this uncertainty (there is still uncertainty in future sleep schedules which we may not foresee, but we can find out the current state). For this reason there are cases where these two sources of randomness can be abstracted into one (become indistinguishable), and in other cases we may not, as we will see in subsequent chapters.

Most of the work developed in this dissertation are based on existing analytical

[1]Randomness is also associated with mobility in a mobile network; this however does not apply to the type of networks studied in this thesis, which are primarily static.

frameworks. However, the introduction of randomness caused by duty-cycling, in addition to the randomness in wireless transmission, often leads to nontrivial extensions to these existing frameworks. This is seen throughout Chapters 2, 3, and 4.

For instance, in Chapter 2 to show the conditions for asymptotic connectivity of a network when a fixed number nodes are deployed in a unit square and nodes are randomly duty-cycled, we adopt the framework developed in [51, 14] for non-duty-cycled networks. The results themselves, however, cannot be directly applied. In particular, in [51] the number of neighbors a node has is a fixed number, whereas it is a random variable in our case due to random duty-cycling. The results for a Poisson network with a fixed density in [14] is also not applicable because the number of active neighbors is a binomial random variable in our network, but a Poisson random variable in [14]. To derive the desired results, we need to show the relationship between these difference cases.

As another example, in Chapter 3, we look for an optimal routing algorithm when nodes are duty-cycled and adopt the stochastic routing analysis framework developed in [24]. The optimal strategies derived in [24], which is for non-duty-cycled networks, however is shown not to be optimal for our case. In particular we have an augmented state space due to duty-cycling, which leads to a generalization of the original framework as we show in Chapter 3.

The above describes a consistent theme throughout this thesis. Below we elaborate on the problems studied and the main contributions of each chapter.

1.2.1 Asymptotic Connectivity

In Chapter 2, we analyze the asymptotic connectivity of a low duty-cycled wireless sensor network under random sleep strategies of the radio transceivers, where nodes are on/active in each time slot with a fixed probability independent of other nodes. Since nodes are not always available due to this type of duty-cycling, the network connectivity is lower than a non-duty-cycled network. While it is intuitively clear that increasing transmission radius, decreasing sleep probability, and/or increasing the deployment density can help improve connectivity, our study sets out to understand how these factors are quantitatively related to connectivity.

This is done via the notion of asymptotic connectivity. Within the context of a low duty-cycled sensor network, the network is said to be asymptotically connected if for all realizations of the random duty-cycling (i.e., the combination of on and off nodes) there exists a path of active nodes from every node to every other node in the network with high probability as the network density approaches infinity. With this definition, we derive conditions under which a low duty-cycled sensor network is asymptotically connected. These conditions essentially specify how the nodes' communication range and the duty-cycling probability should scale as the network grows in order to maintain connectivity. We also consider the random sleep of both the radio transceiver and the sensing device, in order to study the sensing coverage, defined as the probability that any given point in the network is covered by at least one sensor.

The main contributions of this chapter are as follows.

- We derive the relationship between the amount of redundancy required vs. the achievable reduction in duty cycle for a fixed performance criterion within the context of network asymptotic connectivity.

- We derive the conditions for asymptotic connectivity with coverage of the network.

1.2.2 Stochastic Routing for Low Duty-Cycled Sensor Networks

In Chapter 3 we consider the problem of designing good routing algorithms for wireless sensor networks in the presence of very low duty cycles as well as transmission failures due to channel uncertainty. We again consider random duty cycles where a sensor has a fixed probability of being awake during a time slot independent over time and of other sensors. Our model focuses on capturing the randomness in topology caused by duty-cycling and the randomness in transmission outcome caused by uncertain channel conditions. Traditionally, most routing algorithms are deterministic in nature and the route selection is done independent of the sleep state or the success/failture state of the network. They therefore do not fully utilize the information available to nodes in the network. In this study we adopt a stochastic routing framework developed in [24]. Under this framework an event-dependent routing scheme was shown to be optimal with respect to a cost measure, and the routing decisions are functions of the outcome of transmissions (success and failure). This framework, however, does not directly apply to the case of duty-cycling. This motivated us to extend this framework so we can develop similar stochastic routing algorithms for low duty-cycled networks.

The model used in this chapter is thus an extension to [24] in that it captures the randomness of topology caused by sleep state in addition to the randomness in channel conditions. The objective is to seek an optimal routing policy in such networks with respect to performance metrics such as transmission cost and delay, and to resolve the trade-off between these two performance metrics. Various policies are explored and characterized for optimality. The main contributions of this chapter are as follows.

- As a benchmark we develop and analyze a centralized optimal stochastic algorithm for a randomly duty-cycled wireless sensor network.
- We develop a centralized stochastic routing algorithm with reduced state space performing near-optimal when local sleep/wake states of neighbors are available.
- We further develop a distributed algorithm utilizing local sleep/wake states of neighbors which performs better than some existing distributed algorithms such as ExOR [5], etc.

1.2.3 Routing Delay Analysis

Chapter 4 presents a comparison study on opportunistic routing and non-opportunistic routing. Opportunistic routing methods, also referred to as event-based routing, are modeled after the type of routing algorithms we studied in Chapter 3. They have recently been proposed and studied as an effective way of dealing with uncertainties such as transmission failure in a wireless network, as we will see from Chapter 3. The fundamental idea is to make routing decisions like the next hop/relay after (rather than before) the actual transmission has taken place so as to take advantage of the information on realizations of transmission successes, i.e., by selecting as relay a node that actually has successfully received the packet. While intuitively appealing, opportunistic routing methods are not easy to analyze due to the randomness in the actual route a given packet follows. In this chapter we examine the asymptotic delay performance of this class of routing methods in a network with increasing node density, and compare it with that under a non-opportunistic routing method.

Specifically, we consider a network of a fixed area with increasing node density. Each pair of nodes is associated with a transmission success probability, whose value is drawn from a given distribution. We show that when the transmission success probabilities are not bounded away from zero, non-opportunistic routing results in infinite hop delay while opportunistic routing has a constant hop delay and an overall routing delay on the same order as straight-line routing with no transmission failures. In the case where non-opportunistic routing has infinite delay, we show that combining it with multi-path routing is sufficient to turn the delay finite, albeit at the expense of increased transmission overhead. The main contributions of this chapter are as follows.

- We show that when the transmission success probabilities are not bounded away from zero, non-opportunistic routing results in infinite routing delay while opportunistic routing has the same order as a straight-line error-free routing when no packet loss is assumed.
- In the case where non-opportunistic routing has infinite delay, we show that combining it with multi-path routing is sufficient to turn the delay finite, albeit at the expense of increased transmission overhead.

1.2.4 Performance Analysis of Broadcasting Algorithms

In Chapter 5 we develop a method to evaluate the performance of broadcasting algorithms introduced in wireless ad hoc and sensor networks. The objectives of this study is two-fold. One is to capture essential features of each broadcasting scheme and the other is to estimate its performance reasonably accurately. The central question we seek to answer is how to estimate the average performance of a broadcast scheme given a specific network topology (the locations of nodes and a source, and transmission success probability between any pair of nodes). The performance metrics of interest include the fraction of nodes reached by a broadcast, also referred to as reachability, the amount of transmissions incurred in a broadcast, the time it takes for the broadcast to complete (or delay), etc.

We consider two approaches: a *state-space* based model and an *analysis-emulation hybrid* model. The former calculates the metrics by averaging over all possible realizations of the system, and is thus accurate but computationally prohibitive. Its lack of scalability motivates our second approach. In this approach we selectively evaluate a subset of *representative* sample paths, the selection of which depends on certain analytical model, and use such evaluation to estimate the average performance of the broadcast algorithm. This method can thus be viewed as a combination of mathematical analysis and simulation.

The main contribution of this chapter is as follows.

- Given a broadcast scheme and a network topology, we develop a hybrid apporach that combines analytical models with elements of numerical simulation to obtain the desired modeling accuracy and computational efficiency.

1.3 Organization of the Dissertation

The organization of this dissertation comprises as follows. In Chapter 2 we derive the scaling law of a wireless sensor network in the context of connectivity under random duty-cycling. It is followed by Chapter 3 where we present an optimal routing algorithm under random duty-cycling and a decentralized routing algorithm utilizing the limited sleep states information of local neighbors. Chapter 4 analyzes asymptotic routing delay of routing algorithms under a probability distribution of transmission success/failures caused by the lossy nature of wireless channels or/and duty-cycling. In Chapter 5 we present a method to analyze numerically the performance of underlying broadcasting schemes required in most routing algorithms. Finally Chapter 6 concludes this dissertation.

CHAPTER 2

Asymptotic Connectivity of Low Duty-Cycled Wireless Sensor Networks

2.1 Introduction

Many emerging sensor network applications rely heavily on the efficient use of unattended sensors to detect, identify and track targets in order to enhance situation awareness, agility and survivability. Among different types of sensors, unattended ground sensors (UGS) are typically deployed and left to self-organize and carry out various sensing, monitoring, surveillance and communication tasks. These sensors operates on battery power, and energy is not always renewable due to cost, environmental and form-size concerns. This imposes a stringent energy constraint on the design of the communication architecture, communication protocols, and the deployment and operation of these sensors. It is thus critical to operate these sensors in a highly energy efficient manner.

It has been observed that low power sensors consume significant amount of energy while idling in addition to that consumed during transmission and reception. Consequently, it has been widely considered as a key method of energy conservation to turn off sensors that are not actively involved in sensing or communication. By functioning at a *low duty cycle*, i.e., by reducing the fraction of time that a sensor is active/on, sensors are able to conserve energy, which consequently leads to prolonged lifetime. This is particularly applicable in scenarios where sensors are naturally idle for most of the time (e.g., detection of infrequent events such as fire, fault, etc., or transmission of very short messages). However, as sensors alternate between sleep and wake modes, its coverage and communication capability are inevitably disrupted. Duty-cycling sensory devices directly leads to loss of sensing coverage, while duty-cycling radio transceivers directly leads to loss of network connectivity. It is therefore crucial to understand the performance degradation as a result of duty-cycling the sensor nodes, and to design good networking mechanisms that work well with low duty-cycled sensor networks.

In this chapter we aim to understand the fundamental relationship between duty-cycling the radio transceivers and the resulting network connectivity. Specifically we will consider *random* duty-cycling where sensor nodes are on/awake with a certain probability (called the wake/active probability). Connectivity refers to the existence of a route (consisting of active nodes) from each *active* node to every other *active* node in the network. While intuitively increasing nodes' transmission radius and decreasing nodes' active probability have opposite effects on the connectivity, it is less clear how they are related quantitatively to ensure connectivity. We will focus on understanding how these quantities scale as the network density increases, by studying the asymptotic connectivity of the network. Asymptotic connectivity in this context refers to the existence of a route (consisting of active nodes) from each *active* node to every other *active* node in the network, as the number of nodes approaches infinity.

More precisely, we consider a network with n nodes uniformly and independently placed in a unit square in $\Re^2$. Each node is awake with probability $p(n)$ and is connected to active neighbors within the range of transmission $R(n)$ when it is active.

The problem under consideration is how $p(n)$ and $R(n)$ are related to ensure that the network is connected with high probability as n goes to infinity. An important prior work is [51]. Our network model is essentially the same as that studied in [51], with the only difference that in [51] the wake/active probability $p(n)$ is always 1. [51] showed that it is sufficient and necessary for each node to be connected to $\Theta(\log n)$ nearest neighbors to achieve asymptotic connectivity as n approaches infinity. Building on this result, in this study we show that the above randomly duty-cycled network is asymptotically connected with probability one if and only if the *average* number of active neighbors of a node is on the order of $\log(np(n))$. It has to be mentioned that this result cannot be obtained by a straightforward extension to [51] as discussed in more detail in subsequent sections.

The rest of the chapter is organized as follows. In Section 2.3, we include the literature review and discuss the relevance of our study to the related work. We present the network model and our main result in Section 2.2. In Section 2.4 we give a number of preliminary results, and Section 2.5 outlines the proof of the main result. In Section 2.6, the results of connectivity with coverage are presented. We provide thorough discussion on other related issues in Section 2.7. Finally, Section 2.8 concludes the chapter.

2.2 Network Model and Main Results

Consider a unit square in $\Re^2$, where n nodes are deployed uniformly and independently. Time is slotted. In each time slot, a node has a probability $p(n)$ of being awake or active, referred to as the *active probability*. An active node is connected to its active neighbors within a circle of radius $R(n)$, referred to as the transmission range. Such a network is said to be asymptotically connected if there exists a path of active nodes between any pair of two active nodes with high probability as the density n approaches infinity. In order to study the conditions under which such a network is asymptotically connected, we will utilize a number of results derived for a similar, but not duty-cycled network (i.e., where $p(n) = 1$ for all n). We begin by introducing the following types of networks/graphs that will be used in this chapter.

- $\mathcal{G}_p(n, R(n))$ denotes the duty-cycled network mentioned above, i.e., a network formed in a unit square where n nodes are deployed uniformly and independently. In this network a node is active with probability $p(n)$ and when active is connected to its active neighbors within a circle of radius $R(n)$.

- $\mathcal{G}(n, R(n))$ denotes a non-duty-cycled network formed in a unit square with n nodes deployed uniformly and independently. In this network a node is always active and is connected to neighbors within a circle of radius $R(n)$.

- $\mathcal{G}^\lambda(n, R(n))$ denotes a network formed as a Poisson point process with intensity n. In this network a node is always active and is connected to neighbors within a circle of radius $R(n)$.

- $\mathcal{F}(n, \phi_n)$ denotes a network formed in a unit square with n nodes deployed uniformly and independently. In this network a node is always active and is connected to its ϕ_n nearest neighbors.

- $\mathcal{F}^\lambda(n, \phi_n)$ denotes a network formed as a Poisson point process with intensity n. In this network a node is always active and is connected to its ϕ_n nearest neighbors.

The following notations are used throughout this chapter. For two functions $f(n)$ and $g(n)$ defined on some subset of the real line, (1) $f(n) = O(g(n))$ implies that there exist numbers n_0 and M such that $|f(n)| \leq M \cdot |g(n)|$ for all $n > n_0$ (asymptotic upper bound); (2) $f(n) = \Theta(g(n))$ implies that $f(n) = O(g(n))$ and $g(n) = O(f(n))$ (asymptotic tight bound); and (3) $f(n) = o(g(n))$ implies that $\lim_{n\to\infty} f(n)/g(n) = 0$ (asymptotically negligible).

Our main result is shown in the following theorem.

Theorem 2.1. *There exist two constants k_1 and k_2, $0 < k_1 < k_2$, such that:*

1. *for $np(n)R^2(n) = k_2 \log(np(n))$, we have*

$$\lim_{n\to\infty} Pr\{\mathcal{G}_p(n, R(n)) \text{ is connected } \} = 1 \ , \tag{2.1}$$

2. *for $np(n)R^2(n) = k_1 \log(np(n))$, we have*

$$\lim_{n\to\infty} Pr\{\mathcal{G}_p(n, R(n)) \text{ is disconnected } \} = 1 \ . \tag{2.2}$$

Eqn. (2.1) is also commonly viewed as a *sufficient* condition on connectivity and Eqn. (2.2) commonly viewed as a *necessary* condition on connectivity. Put together, $np(n)R^2(n) = \Theta(\log(np(n))$ can be viewed as the sufficient and necessary conditions for asymptotic connectivity. In subsequent sections we will also refer to these two equations as part I and part II of the theorem.

Below we sketch the idea of the proof of the above theorem and discuss this result within the context of other existing results on asymptotic connectivity.

Figure 2.1 summarizes the main idea of the proof, and illustrates where our technical contributions lie. The network we are interested in, $\mathcal{G}_p(n, R(n))$, is shown on the top left. To prove the theorem, we first show that if a Poisson network with intensity $np(n)$, i.e., $\mathcal{G}^\lambda(np(n), R(n))$, is asymptotically connected/disconnected given the condition $np(n)R(n)^2 = k\log(np(n))$ for some $k > 0$, then $\mathcal{G}_p(n, R(n))$ is asymptotically connected/disconnected given the same condition (for possibly different constants).

$$
\begin{tikzcd}
G_p(n, R(n)) & G^{\lambda}(np(n), R(n)) \arrow[l, "A"'] \\
F^{\lambda}(n, \phi_n) \arrow[r, "C"] & F^{\lambda}(np(n), \phi_{np}) \arrow[u, "B"']
\end{tikzcd}
$$

Figure 2.1: Outline of the proof of Theorem 2.1.

This process is illustrated by the arrow labeled with "A" in the figure. Conceptually, because of the random duty-cycling, there are only on average $np(n)$ nodes awake in the network at any instance of time. This makes the network $\mathcal{G}_p(n, R(n))$ behave like a Poisson network rather than one with a fixed number of nodes. However, in order to study asymptotic connectivity $np(n)$ needs to approach infinity, which renders inapplicable the standard result of approximating a binomial distribution with a Poisson distribution (which assumes a finite intensity). Although this seems a highly intuitive result, we were not able to find a prior proof. We give one such proof in Lemma 2.3, where we establish the Poisson approximation of a binomial distribution when $np(n) \to \infty$.

We next show that if the network $\mathcal{F}^{\lambda}(np(n), \phi_{np})$, i.e., a Poisson network with intensity $np(n)$ where each node is connected to its ϕ_{np} nearest neighbors, is asymptotically connected/disconnected given the condition $\phi_{np} = c \log(np(n))$, for some $c > 0$, then the network $\mathcal{G}^{\lambda}(np(n), R(n))$ is asymptotically connected/disconnected given the condition $np(n)R(n)^2 = k \log(np(n))$ for some $k > 0$.

This process is illustrated by the arrow labeled with "B" in the figure. Here $\mathcal{F}^{\lambda}(np(n), \phi_{np})$ is a Poisson network with ϕ_{np} neighbors for each node, and $\mathcal{G}^{\lambda}(np(n), R(n))$ is a Poisson network with neighbors within a finite radius $R(n)$ of each node. Note that for the latter, the condition $np(n)R(n)^2 = k \log(np(n))$ for some $k > 0$ is on the average number of neighbors a node has, whereas for the former the condition

$\phi_{np} = c\log(np(n))$ for some $c > 0$ is on the actual number of neighbors a node has.

The last step is to show that network $\mathcal{F}^{\lambda}(np(n), \phi_{np})$ is asymptotically connected/disconnected given the condition $\phi_{np} = c\log(np(n))$, for some $c > 0$. This network is essentially the same as $\mathcal{F}^{\lambda}(n, \phi_n)$ (with a different intensity). This result is obtained in similar ways as in [14], which showed the same result for $\mathcal{F}(n, \phi_n)$. This step is illustrated by the arrow labeled with "C" in the figure.

2.3 Related Works

Two most relevant results to that studied in this chapter are from [51] and [14], respectively. In particular, as mentioned above [51] studied a network of the type $\mathcal{F}(n, \phi_n)$, and it was shown that it is sufficient and necessary for each node to be connected to its $\Theta(\log n)$ nearest neighbors in order to achieve asymptotic connectivity for this network. An immediate thought was whether one could simply replace n with $np(n)$ in this result to obtain the conditions for a network of the type $\mathcal{G}_p(n, R(n))$, assuming $np(n) \to \infty$. Although intuitively appealing, there is a conceptual difference. Replacing n with $np(n)$ in this result implies that the sufficient and necessary conditions for asymptotic connectivity are for every active node to be connected to $np(n)$ nearest active neighbors. However, these conditions are not directly guaranteed when the neighborhood of each node is defined by a fixed radius $R(n)$ with randomly deployed nodes, and when the nodes are randomly duty-cycled. Instead, what Theorem 2.1 shows is that it is sufficient and necessary for each active node to be connected to an average of $\Theta(\log(np(n)))$ active neighbors for asymptotic connectivity of a network of the type $\mathcal{G}_p(n, R(n))$.

In [14] a network of the type $\mathcal{G}(n, R(n))$ was considered, and it was shown that with $\pi R^2(n) = \frac{\log n + c(n)}{n}$, the network is asymptotically connected with probability one if and only if $c(n) \to \infty$. This result is not directly used in our study. However, throughout this chapter we follow heavily the basic definitions and methods used by [51] and [14], as well as use a number of (intermediate) results derived in them with appropriate modifications. These will be pointed out in subsequent sections.

[41] showed that the sufficient and necessary conditions for asymptotic coverage with connectivity in a *grid* network are $p(n)R^2(n) = \Theta(\frac{\log n}{n})$. Although mathematically similar, these conditions are not the same as the ones given by Theorem 2.1, since asymptotic coverage with connectivity is a different measure from asymptotic connectivity, and a grid network is different from a random network. [41] also showed that the sufficient condition for asymptotic connectivity in the grid network is in the form of

$$np(n)e^{-\frac{\pi p(n)R^2(n)n}{2}} \to 0 \text{ as } n \to \infty.$$

It can be shown that $p(n)R^2(n) = \Theta(\frac{\log n}{n})$ implies $np(n)e^{-\frac{\pi p(n)R^2(n)n}{2}} \to 0$ as n and $np(n)$ both go to infinity. The reverse is not necessarily true. Therefore, we see that the condition for a randomly deployed network, i.e., $p(n)R^2(n) = \Theta(\frac{\log n}{n})$, is more restrictive than that for a grid network. Other related work includes [35], which studied the necessary and sufficient conditions of both asymptotic coverage and connectivity for a network with fixed node density λ but increasing area A. The necessary condition is $R = \sqrt{\frac{(1-\epsilon)\log A}{\lambda\pi}}$. The sufficient condition is $R = \sqrt{\frac{(1+\epsilon)\log A}{\lambda\pi}}$. Therefore, the expected number of neighbors needed for asymptotic connectivity is $\Theta(\log A)$. Furthermore, [35] derives a bound for the expected number of neighbors needed to ensure the existence of an infinite component of connected nodes as network area A approaches ∞. However, this does not imply that all nodes are connected. There will be infinitely many isolated nodes. A graph is said to be k-connected if for each node pair there exist at least k mutually independent paths connecting them.

[1] studies the probability of k-connectivity. The degree is the number of neighbors of a node. The minimum node degree of a graph G is denoted as $d_{min}(G)$. This paper claims that P(G is k-connected) $= P(d_{min} \geq k)$ with high probability. In order to calculate $P(d_{min} \geq k)$, a Poisson-approximation approach is used. Furthermore, the degrees of nodes are assumed to be mutually independent, which is not true because two nodes may have overlapped transmission area.

[55] introduces path connectivity. [55] claims that in most cases it is an overkill to demand "all nodes are able to communicate with each other simultaneously in 95% time" (graph connectivity which is the definition we uses). It would be more reasonable to demand "any nodes could find a path to any other node at 95% success rate at any time" (path connectivity). Given the total node number is n, the path connectivity is $C_{path} = \frac{\sum_{i \neq j} Conn(i,j)}{n(n-1)}$ where $Conn(i,j) = 0/1$ if no/at least 1 path between node i and j. C_{path} is closely related to the size of the largest component with connected nodes (ζn), i.e. $C_{path} \cong \zeta^2$, where ζ is a constant. Furthermore, [55] discovers the different behaviors of graph connectivity and path connectivity. It is well known that there is a critical density where graph connectivity/path connectivity arises abruptly. For a network with node density less than critical density, path connectivity decreases as total node number increases (the density is fixed). For a network with node density larger than critical density, path connectivity increases as the total number of nodes increases (the density is fixed). However, graph connectivity always decreases as total node number increases (the density is fixed).

[46] proposes a scheme which can reduce system overall energy consumption by turning off some redundant nodes and guarantee that the original sensing coverage is maintained. A node decides to turn it off when it discovers that its whole sensing area is fully embraced by the union set of its neighbors'. This is done by some

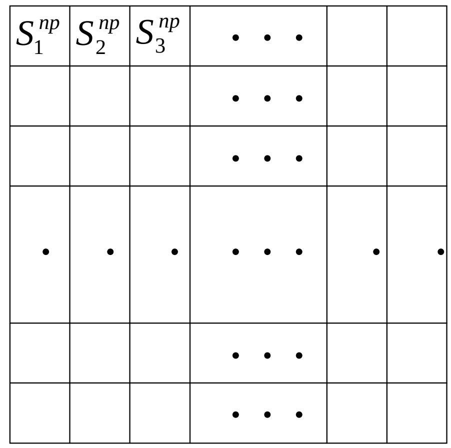


Figure 2.2: The square tessellation τ_S^{np}.

geometry calculation. [18] improves the algorithm proposed by [46] and provides a mathematical analysis on the design about how to put sensors in the sleep state randomly.

2.4 Preliminaries

For the proof of Theorem 2.1, we need the following definitions which were originally defined in [51], with slight generalization to account for $p(n) < 1$.

Definition 2.1. Square tessellation τ_S^{np}. *The unit square is split equally into* $M_{np} = \lceil \sqrt{\frac{np(n)}{K \log(np(n))}} \rceil^2$ *small squares as depicted in Figure 2.2, where a constant* $K > 0$ *is a tunable parameter, and* $\lceil x \rceil$ *is the smallest integer larger than or equal to* x*. This tessellation of the unit square will be denoted by* τ_S^{np}*. The small squares are denoted by* $S_i^{np}, i = 1, 2, \cdots, M_{np}$*, from left to right, and from top to bottom.*

Definition 2.2. k-filling event. *Consider a structure composed of 21 squares each of side length* $d/6$ *and placed in a larger square of side length* d*: one at the center and the others at the periphery of the larger square with distance* $d/4$ *between the center square and the others. A k-filling event occurs if there are at least k nodes in each of 21 small squares and no nodes in the space between the center square and the others.*

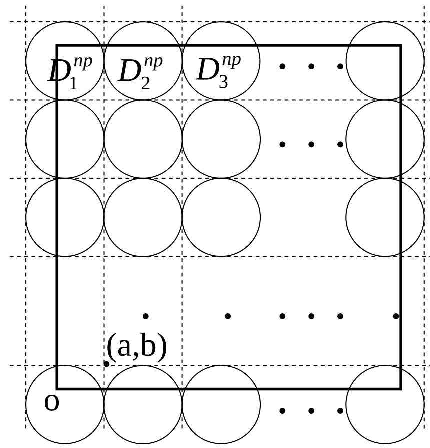


Figure 2.3: The disk tessellation τ_D^{np}.

Definition 2.3. Disk tessellation $\tau_D^{np}(a,b)$. *Consider a unit square with its bottom left corner being the origin, as shown in Figure 2.3. Let r be such that $\pi r^2 = \frac{K\log(np(n))}{np(n)}$, where $K > 0$ is a tunable parameter. Consider a grid of squares of size $2r$, with corners at $(a \mod 2r,\ b \mod 2r)$. Inside each square, we inscribe a disk of area $\frac{K\log(np(n))}{np(n)}$. The set of all disks intersecting the unit square are called the Disk Tessellation $\tau_D^{np}(a,b)$. The disks intersecting the unit square are denoted by $D_i^{np}, i = 1 \leq M_{np}$.*

Throughout our analysis, the asymptotic regime of interest is where the duty cycle $p(n) \to 0$, $n \to \infty$ and $np(n) \to \infty$.

Consider the network $\mathcal{G}^\lambda(np(n), R(n))$, where $0 < p(n) < 1$. Denote the number of nodes that fall into the unit square by $\widetilde{M}_{np}$, and denote the number of nodes that fall into square S_i^{np} by $\widetilde{N}_i^{np}$.

Lemma 2.1. $\lim_{np(n)\to\infty} Pr\{|\widetilde{M}_{np} - np(n)| \leq \sqrt{np(n)\log(np(n))}\} = 1.$

Proof. Since $\widetilde{M}_{np}$ is a Poisson random variable with mean $= np(n)$ and variance

$= np(n)$, by Chebychev's inequality,

$$
\begin{aligned}
Pr\{|\widetilde{M}_{np} - np(n)| > \sqrt{np(n)\log(np(n))}\} &\leq \frac{np(n)}{np(n)\log(np(n))} \\
&= \frac{1}{\log(np(n))} \to 0, \text{ as } np(n) \to \infty.
\end{aligned}
$$

□

Consider $\mathcal{G}_p(n, R(n))$. Denote the number of active nodes in the unit square by M_n^a, which is a random variable. Denote the number of active nodes in square S_i^{np} by N_i^a.

Lemma 2.2. $\lim_{np(n)\to\infty} Pr\{|M_n^a - np(n)| \leq \sqrt{np(n)\log(np(n))}\} = 1.$

Proof. Since M_n^a is a Binomial random variable with mean $= np(n)$ and variance $= np(n)(1-p(n))$, by Chebychev inequality,

$$
\begin{aligned}
Pr\{|M_n^a - np(n)| > \sqrt{np(n)\log(np(n))}\}\} &\leq \frac{np(n)(1-p(n))}{np(n)\log(np(n))} \\
&= \frac{1-p(n)}{\log(np(n))} \to 0, \text{ as } np(n) \to \infty.
\end{aligned}
$$

□

Let n be sufficiently large and p be small. When its product $np(n)$ is of moderate magnitude, the poisson approximation of binomial distribution has been proven in the literature, see e.g., [11]. In this chapter, we need it in the case of $np(n) \to \infty$ as $n \to \infty$, which is proven in the following lemma.

Lemma 2.3. *Suppose that $p(n) \to 0$ and $np(n) \to \infty$ as $n \to \infty$. For any nonnegative $j \leq n$ and sufficiently large n, $Pr\{M_n^a = j\}$ is approximated by $Pr\{\widetilde{M}_{np} = j\}$, i.e., in the limit their difference goes to zero.*

Proof. We have that

$$Pr\{M_n^a = j\} = \binom{n}{j} p(n)^j (1-p(n))^{n-j} \ ,$$
$$Pr\{\widetilde{M}_{np} = j\} = \frac{(np(n))^j e^{-np(n)}}{j!}.$$

As $Pr\{M_n^a = j\}$ is a binomial distribution determined by n and $p(n)$, we will denote it by $b(j; n, p(n))$. Thus

$$b(0; n, p(n)) = (1-p(n))^n. \tag{2.3}$$

By the definition of the derivative of function $\log x$, we have

$$\lim_{\delta \to 0} \frac{\log x - \log(x-\delta)}{\delta} = \frac{1}{x} \ . \tag{2.4}$$

Since $p(n) \to 0$ as $n \to \infty$, Eqn. (2.4) can be written as

$$\lim_{n \to \infty} \frac{\log x - \log(x - p(n))}{p(n)} = \frac{1}{x}.$$

For $x = 1$, we have

$$\lim_{n \to \infty} \frac{-\log(1-p(n))}{p(n)} = 1.$$

In other words, $\forall \epsilon_1 > 0$, there exists $N_1 > 0$ such that $n > N_1$ implies $|\frac{-\log(1-p(n))}{p(n)} - 1| < \epsilon_1$. Let $\Delta(n) \equiv \frac{-\log(1-p(n))}{p(n)} - 1$, such that $\Delta(n) \in [-\epsilon_1, \epsilon_1]$. For all $\epsilon_1 > 0$ and $\epsilon_2 > 0$, there exists $N_2 > 0$ such that $n > \max\{N_1, N_2\}$ implies

$$\begin{aligned}
|(1-p(n))^n - e^{-np(n)}| &= |(1-p(n))^{-\frac{1}{p(n)}\cdot(-np(n))} - e^{-np(n)}| \\
&= |e^{\frac{-1}{p(n)}\log(1-p(n))\cdot(-np(n))} - e^{-np(n)}| \\
&= |e^{(1+\Delta(n))\cdot(-np(n))} - e^{-np(n)}| \\
&= |e^{-np(n)}(e^{-np(n)\cdot\Delta(n)} - 1)|. \qquad (2.5)
\end{aligned}$$

Because $|\Delta(n)|$ is bounded by ϵ_1, $|e^{-np(n)\cdot\Delta(n)} - 1|$ is bounded by some $N_3 > 0$. Therefore, Eqn. (2.5) $\leq |e^{-np(n)}| \cdot N_3 < \epsilon_2$. Thus for sufficiently large n we have

$$b(0; n, p(n)) \approx e^{-np(n)}.$$

Furthermore, for any fixed j we have

$$\frac{b(j;n,p(n))}{b(j-1;n,p(n))} = \frac{np(n)-(j-1)p(n)}{j(1-p(n))}.$$

Therefore for sufficiently large n, we have

$$b(j;n,p(n)) \approx \frac{(np(n))^j}{j!}e^{-np(n)} = Pr\{\widetilde{M}_{np} = j\}.$$

$\square$

Lemma 2.4. (Lemma 3.2.5 (iii) in [14]) *Suppose* Y *is a Poisson random variable with parameter* λ*, then for any* $K > \frac{1}{\log(4/e)}$*, we have*

$$\lim_{\lambda\to+\infty} \frac{1}{\lambda}e^{\frac{1}{K}\lambda} \cdot Pr\{|Y-\lambda| \geq \mu\lambda\} = 0, \forall \mu \in (\mu^*, 1),$$

where μ^* *is the root of* $-\mu^* + (1+\mu^*)\log(1+\mu^*) = \frac{1}{K}$*.*

Lemmas 2.5 and 2.6 below are based on Lemma 3.1 in [14], with a slight modification by using $np(n)$ instead of n.

Lemma 2.5. *For any* $K > \frac{1}{log(4/e)}$*,*
$\lim_{np(n)\to\infty} Pr\{\max_i |\widetilde{N}_i^{np} - K\log(np(n))| \leq \mu K\log(np(n))\} = 1, \forall\mu \in (\mu^*, 1),$
where $\mu^* \in (0,1)$ *is the sole root of the equation* $-\mu^* + (1+\mu^*)\log(1+\mu^*) = \frac{1}{K}$*.*

Proof. Recall $M_{np} \triangleq \frac{np(n)}{K\log(np(n))}$. By invoking the independence property of the Poisson process for the random variables $\widetilde{N}_1^{np}, \widetilde{N}_2^{np}, \cdots, \widetilde{N}^{np}_{\frac{np(n)}{K\log(np(n))}}$, we have

$$\begin{aligned}
&Pr\{\max_{1\leq i\leq M_{np}} |\widetilde{N}_i^{np} - K\log(np(n))| \leq \mu K\log(np(n))\} \\
&\qquad = \prod_{i=1}^{M_{np}} Pr\{|\widetilde{N}_i^{np} - K\log(np(n))| \leq \mu K\log(np(n))\} \\
&\qquad = (Pr\{|\widetilde{N}_1^{np} - K\log(np(n))| \leq \mu K\log(np(n))\})^{M_{np}} \\
&\qquad = (1 - Pr\{|\widetilde{N}_1^{np} - K\log(np(n))| > \mu K\log(np(n))\})^{M_{np}} \\
&\qquad = e^{\frac{np(n)}{K\log(np(n))}\cdot\log(1-Pr\{|\widetilde{N}_1^{np} - K\log(np(n))| > \mu K\log(np(n))\})}.
\end{aligned}$$

If we let $\rho_{np} \triangleq K\log(np(n))$, which is the mean value of $\widetilde{N}_1^{np}$, then

$$Pr\{\max_{1\le i\le M_{np}} |\widetilde{N}_i^{np} - K\log(np(n))| \le \mu K\log(np(n))\}$$
$$= \exp\{\frac{e^{\frac{\rho_{np}}{K}}}{\rho_{np}} \cdot \log(1 - Pr\{|\widetilde{N}_1^{np} - \rho_{np}| > \mu\rho_{np}\})\}.$$

Since by Chebychev's inequality,

$$Pr\{|\widetilde{N}_1^{np} - \rho_{np}| > \mu\rho_{np}\} \le \frac{\mathrm{var}(\widetilde{N}_1^{np})}{(\mu\rho_{np})^2} = \frac{\rho_{np}}{(\mu\rho_{np})^2} = \frac{1}{\mu^2\rho_{np}} \to 0, \text{ as } np(n) \to \infty,$$

and $\log(1-x)$ is approximated by $-x$ for small x, we have

$$Pr\{\max_{1\le i\le M_{np}} |\widetilde{N}_i^{np} - K\log(np(n))| \le \mu K\log(np(n))\}$$
$$= e^{-\frac{e^{\frac{\rho_{np}}{K}}}{\rho_{np}} \cdot Pr\{|\widetilde{N}_1^{np} - \rho_{np}| > \mu\rho_{np}\}\cdot(1+o(1))}.$$

Hence, by Lemma 2.4, we deduce that

$$Pr\{\max_{1\le i\le M_{np}} |\widetilde{N}_i^{np} - K\log(np(n))| \le \mu K\log(np(n))\} \to 1, \text{ as } np(n) \to \infty.$$

□

Consider now the disk tessellation $\tau_D^{np}(a,b)$ in a unit square with nodes deployed as a Poisson point process with intensity $np(n)$. Similarly to the square tessellation, let the number of nodes that fall into disk D_i^{np} be denoted as $\widetilde{N}_{D,i}^{np}$. The next few lemmas bound the number of nodes in a disk tessellation. The proofs of these lemmas are essentially the same as that in [14] and are thus not included.

Lemma 2.6. *For any $K > \frac{1}{\log(4/e)}$ and any point sequence $\{(a_n,b_n) \in \mathbb{R}^2, n = 1,2,\cdots\}$, we have that*

$$Pr\{\widetilde{N}_{D,i}^{np} \le (1+\mu)K\log(np(n)), \text{ for any disk } D_i^{np} \text{ in tessellation } \tau_D^{np}(a_n,b_n)\} \to 1,$$

as $np(n) \to \infty, \forall \mu \in (\mu^*, 1)$, where $\mu^* \in (0,1)$ is the sole root of the equation $-\mu^* + (1+\mu^*)\log(1+\mu^*) = \frac{1}{K}$.

Let us consider graph $\mathcal{G}^{\lambda}(np(n), R(n))$. Let $P^{\lambda;(1)}(np(n), R(n))$ be the probability that $\mathcal{G}^{\lambda}(np(n), R(n))$ has at least one isolated node (i.e, one with no neighbors) and $P_d^{\lambda}(np(n), R(n))$ be the probability that $\mathcal{G}^{\lambda}(np(n), R(n))$ is disconnected. From *continuum percolation* [28], we know that $P_d^{\lambda}(np(n), R(n))$ is asymptotically the same as $P^{\lambda;(1)}(np(n), R(n))$. Consider $\mathcal{G}(np(n), R(n))$, the network with exactly $np(n)$ number of nodes. Let $P_d(np(n), R(n))$ be the probability that $\mathcal{G}(np(n), R(n))$ is disconnected.

Lemma 2.7. (Lemma 3.1 in [14]) *If* $\pi R^2(n) = \frac{\log(np(n))+c(n)}{np(n)}$, *then*

$$\limsup_{np(n)\to\infty} P^{\lambda;(1)}(np(n), R(n)) \leq e^{-c},$$

where $c = \lim_{n\to\infty} c(n)$.

Lemma 2.8. (Theorem 2.1 in [14]) *If* $\pi R^2(n) = \frac{\log(np(n))+c(n)}{np(n)}$, *then*

$$\liminf_{np(n)\to\infty} P_d(np(n), R(n)) \geq e^{-c}(1 - e^{-c}),$$

where $c = \lim_{n\to\infty} c(n)$.

The following theorem is proven using intermediate results in [14].

Theorem 2.2. *The network* $\mathcal{G}^{\lambda}(np(n), R(n))$ *with* $\pi R^2(n) = \frac{\log(np(n))+c(n)}{np(n)}$ *is connected with probability one as* $np(n) \to \infty$ *and* $n \to \infty$ *if and only if* $\lim_{n\to\infty} c(n) = \infty$.

Proof. (Sufficiency) From percolation theory, for any $\epsilon > 0$ and for all sufficiently large $np(n)$, we have

$$\begin{aligned} P_d^{\lambda}(np(n), R(n)) &\leq (1+\epsilon) P^{\lambda;(1)}(np(n), R(n)) \\ &\leq (1+\epsilon) e^{-c(n)} \ , \end{aligned}$$

where the second inequality is from Lemma 2.7. Since $\epsilon > 0$ is arbitrary, we have

$$\limsup_{np(n)\to\infty} P_d^\lambda(np(n), R(n)) \le e^{-c}.$$

(Necessity) From Eqn. (1.21) in [14],

$$\begin{aligned} P_d^\lambda(np(n), R(n)) &\ge P_d(np(n), R(n))(\frac{1}{2} - \epsilon) - \frac{e^{-np(n)\pi R^2(n)}}{\pi R^2(n)} \\ &\ge P_d(np(n), R(n))(\frac{1}{2} - \epsilon) - \frac{e^{-c(n)}}{\log(np(n)) + c(n)}. \end{aligned}$$

Based on Lemma 2.8, since $\epsilon > 0$ is arbitrary,

$$\liminf_{np(n)\to\infty} P_d^\lambda(np(n), R(n)) \ge \frac{1}{2} e^{-c}(1 - e^{-c}).$$

□

2.5 Proof of Theorem 2.1

In this section, we prove both parts of Theorem 2.1. For simplicity we will ignore edge effect in our discussion, but note that edge effect does not alter the main theorem (see also [51, 14]). The proof of each part consists of three steps.

2.5.1 Part I

In part I, the proof proceeds as follows:

(1) Given $np(n)R(n)^2 = k_2 \log(np(n))$ for some $k_2 > 0$, we show $\mathcal{G}_p(n, R(n))$ is asymptotically connected if $\mathcal{G}^\lambda(np(n), R(n))$ is asymptotically connected.

(2) It is shown that if there exists c_2 such that $\mathcal{F}^\lambda(np(n), c_2 \log(np(n)))$ is asymptotically connected, then there exists k_2 such that $\mathcal{G}^\lambda(np(n), R(n))$ is asymptotically connected with $np(n)R(n)^2 = k_2 \log(np(n))$.

(3) We show that $\mathcal{F}^\lambda(np(n), c_2 \log(np(n)))$ is asymptotically connected for some $c_2 > 0$.

For the first step, note that $R(n)$ is bounded and that $n \to \infty$ implies $np(n) \to \infty$. Let us denote by $\mathcal{M}$ such that $\forall j \in \mathcal{M}$ satisfies $|j - np(n)| \leq \sqrt{(np(n)\log(np(n)))}$. For sufficiently large n,

$$
\begin{aligned}
&Pr\{\mathcal{G}_p(n, R(n)) \text{ is connected}\} \\
&\quad = \sum_{j=0}^{n} Pr\{\mathcal{G}_p(n, R(n)) \text{ is connected}|M_n^a = j\} \cdot Pr\{M_n^a = j\} \\
&\quad = (\sum_{j\in\mathcal{M}} + \sum_{otherwise})Pr\{\mathcal{G}_p(n, R(n)) \text{ is connected}|M_n^a = j\} \cdot Pr\{M_n^a = j\} \\
&\quad = \sum_{j\in\mathcal{M}} Pr\{\mathcal{G}_p(n, R(n)) \text{ is connected}|M_n^a = j\} \cdot Pr\{M_n^a = j\} + o(1) \\
&\quad = \sum_{j\in\mathcal{M}} Pr\{\mathcal{G}^\lambda(np(n), R(n))\text{is connected}|\widetilde{M}_{np} = j\} \\
&\qquad\qquad \cdot Pr\{\widetilde{M}_{np} = j\} \cdot (1 + o(1)) + o(1) \ ,
\end{aligned}
\tag{2.6}
$$

where the third equality is based on Lemma 2.1. The fourth equality is based on Lemma 2.3 and the fact that $\mathcal{G}_p(n, R(n))$ given j active nodes is the same as $\mathcal{G}^\lambda(np(n), R(n))$ given j nodes are in the network. From Lemma 2.2 we have that Eqn. (2.6) can be written as

$$(1 + o(1)) \cdot (Pr\{\mathcal{G}^\lambda(np(n), R(n)) \text{ is connected}\} + o(1)) + o(1).$$

Therefore if

$$\lim_{n\to\infty} Pr\{\mathcal{G}^\lambda(np(n), R(n)) \text{ is connected}\} = 1,$$

then

$$\lim_{n\to\infty} Pr\{\mathcal{G}_p(n, R(n)) \text{ is connected}\} = 1 \ ,$$

thus completing the first step.

In Step 2 we show that if there exists c_2 for $\mathcal{F}^\lambda(np(n), c_2\log(np(n)))$ to be asymptotically connected, then there exists k_2 for $\mathcal{G}^\lambda(np(n), R(n))$ to be asymptotically connected with $np(n)R(n)^2 = k_2\log(np(n))$. To prove this, let us tessellate $\mathcal{G}^\lambda(np(n), R(n))$

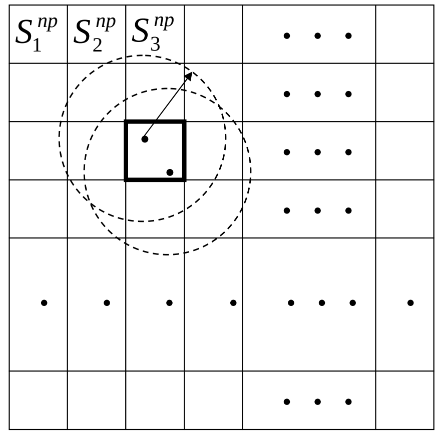


Figure 2.4: Nodes with radius of transmission $R(n) = \sqrt{\frac{2K\log(np(n))}{np(n)}}$ on τ_S^{np}.

by τ_S^{np}, with K, μ satisfying Lemma 2.5. Consider some nodes whose radius is $R(n) = \sqrt{\frac{2K\log(np(n))}{np(n)}}$ on τ_S^{np}, as shown in Figure 2.4. Every circle contains a small square. From Lemma 2.5, we know that each circle contains more than or equal to $K(1-\mu)\log(np(n))$ nodes with high probability, where $\mu \in (\mu^*, 1)$. We construct another graph by connecting each node with its nearest $K(1-\mu)\log(np(n)) - 1$ neighbors, which is $\mathcal{F}^\lambda(np(n), K(1-\mu)\log(np(n))-1)$. If $\mathcal{F}^\lambda(np(n), K(1-\mu)\log(np(n))-1)$ is asymptotically connected, then $\mathcal{G}^\lambda(np(n), R(n))$ with $np(n)R(n)^2 = 2K\log(np(n))$ is asymptotically connected. Thus there exists $k_2 = 2K$ when $c_2 = K(1-\mu)$. This completes the second step.

Finally, in the third step we want to prove that $\mathcal{F}^\lambda(np(n), c_2\log(np(n)))$ is asymptotically connected for $c_2 > \frac{2}{\log(4/e)}$. It suffices to show that for some $\delta > 0$,

$$\lim_{n\to\infty} Pr\{\mathcal{F}^\lambda(np(n), (2/\log(4/e) + \delta)\log(np(n))) \text{ is connected}\} = 1.$$

This proof is similar to that in [51] and is as follows.

According to Lemma 2.6, $\mu^* \to 1$ as $K \to (1/\log(4/e))^+$. So for any $\delta > 0$, there is a constant $\delta' > 0$ such that

$$K = 1/\log(4/e) + \delta' \Rightarrow (1+\mu^*)K < 2/\log(4/e) + \delta. \tag{2.7}$$

For the rest of this proof, we fix the parameter K in the Disk tessellation to be the one in Eqn. (2.7), and fix μ such that

$$1 > \mu > \mu^* \text{ and } (1+\mu)K < 2/\log(4/e) + \delta.$$

Let $r_{np} \triangleq \sqrt{\frac{K\log(np(n))}{\pi np(n)}}$ be the radius of the disks in the Disk tessellation. Then choose two positive constants ϵ,$\eta \in (0,1)$ such that

$$\pi(r_{np} - \epsilon r_{np})^2 > \frac{(1+\eta)\log(np(n))}{np(n)}. \tag{2.8}$$

Now let us tessellate the unit square by a collection of several disk tessellations:

$$\tau_\epsilon^{np} \triangleq \{\tau_D^{np}(i \cdot \epsilon r_{np}, j \cdot \epsilon r_{np}), i, j = 0, 1, 2, \cdots, 2 \cdot [\frac{1}{\epsilon}] + 1\}.$$

This collection of tessellations has the following property: For any point (a, b) in the unit square, there is a disk in τ_ϵ^{np} whose center is within a distance of ϵr_{np} from the point (see Figure 2.3). Since the number of tessellations in τ_ϵ^{np} is finite, by Lemma 2.6, we know that

$$\begin{aligned} &Pr\{\text{Every disk of } \tau_\epsilon^{np} \text{ contains no more than} \\ &\quad (2/\log(4/e) + \delta)\log(np(n)) \text{ nodes}\} \to 1, \text{ as } n \to \infty. \end{aligned}$$

By the choice of r_{np}, ϵ and τ_ϵ^{np}, any disk with radius $(1-\epsilon)r_{np}$ and centered in the unit square will be contained in a disk in the collection of tessellations τ_ϵ^{np} (see Figure 2.4). So if any of the disks of the tessellation collection τ_ϵ^{np} contains no more than $(2/\log(4/e)+\delta)\log(np(n))$ nodes, then each node of $\mathcal{F}^\lambda(np(n), c_2\log(np(n)))$ will be connected to every node that is within distance of $(1-\epsilon)r_{np}$. So if we define $B_{np} \triangleq$ {Every disk of τ_ϵ^{np} contains no more than $(\frac{2}{\log(4/e)} + \delta)\log(np(n))$ nodes}, then

$$Pr\{\mathcal{F}^\lambda(np(n), (2/\log(4/e) + \delta)\log(np(n))) \text{ is connected}|B_{np}\} \to 1, \text{ as } n \to \infty.$$

Therefore

$$
\begin{aligned}
&Pr\{\mathcal{F}^{\lambda}(np(n), (2/\log(4/e)+\delta)\log(np(n))) \text{ is connected}\}\\
&= Pr\{B_{np}\}\cdot Pr\{\mathcal{F}^{\lambda}(np(n), (2/\log(4/e)+\delta)\log(np(n))) \text{ is connected}|B_{np}\}\\
&\quad + Pr\{B_{np}^{c}\}\cdot Pr\{\mathcal{F}^{\lambda}(np(n), (2/\log(4/e)+\delta)\log(np(n))) \text{ is connected}|B_{np}^{c}\}\\
&= (1+o(1))\cdot(1+o(1))+o(1)\to 1,\ \text{ as } n\to\infty.
\end{aligned}
$$

Hence we proved that $\mathcal{F}^{\lambda}(np(n), c_2\log(np(n)))$ is asymptotically connected for $c_2 > \frac{2}{\log(4/e)}$, completing the third step.

If $\mu^* < 0.4808$ and $\mu \in (\mu^*, 0.4808)$, $K(1-\mu) > \frac{2}{\log(4/e)}$. Therefore, the choices $K > 9.9635$ and $k_2 > 19.9217$ satisfy the condition $c_2 > \frac{2}{\log(4/e)}$.

2.5.2 Part II

The proof of the second part of Theorem 2.1 follows a very similar procedure, consisting of three steps:

(1) Given $np(n)R(n)^2 = k_1\log(np(n))$ for some $k_1 > 0$, we show $\mathcal{G}_p(n, R(n))$ is asymptotically disconnected if $\mathcal{G}^{\lambda}(np(n), R(n))$ is asymptotically disconnected.

(2) It is shown that if there exists c_1 such that $\mathcal{F}^{\lambda}(np(n), c_1\log(np(n)))$ is asymptotically disconnected, then there exists k_1 such that $\mathcal{G}^{\lambda}(np(n), R(n))$ is asymptotically disconnected with $np(n)R(n)^2 = k_1\log(np(n))$.

(3) We show that $\mathcal{F}^{\lambda}(np(n), c_1\log(np(n)))$ is asymptotically disconnected for some $c_1 > 0$.

In the first step, similar to part I we will use the fact that $n \to \infty$ implies $np(n) \to \infty$. With slight modification from connectivity to disconnectivity on the argument used in part I of the proof given early, one can easily show that if

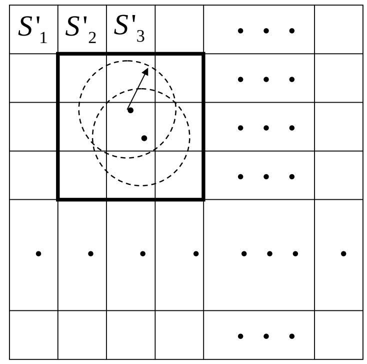


Figure 2.5: Nodes with radius of transmission $R(n) = \sqrt{\frac{K' \log(np(n))}{np(n)}}$ on $\tau_{S'}^{np}$.

$\mathcal{G}^{\lambda}(np(n), R(n))$ is asymptotically disconnected with probability 1, then $\mathcal{G}_p(n, R(n))$ is also asymptotically disconnected with probability one. This completes the first step of the proof of part II.

In the second step we show that if there exists c_1 such that $\mathcal{F}^{\lambda}(np(n), c_1 \log(np(n)))$ is asymptotically disconnected, then there exists k_1 such that $\mathcal{G}^{\lambda}(np(n), R(n))$ with $np(n)R(n)^2 = k_1 \log(np(n))$ is asymptotically disconnected. To prove this, we tessellate $\mathcal{G}^{\lambda}(np(n), R(n))$ by τ_S^{np}, with K, μ satisfying Lemma 2.5. Furthermore, we split each square into $\lceil\sqrt{\frac{9 \cdot 21(1+\mu)}{1-\mu}}\rceil^2$ smaller squares. Denote by $\tau_{S'}^{np}$ the new tessellation with $\lceil\sqrt{\frac{np(n)}{K \log(np(n))}}\rceil^2 \cdot \lceil\sqrt{\frac{9 \cdot 21(1+\mu)}{1-\mu}}\rceil^2$ squares and let $\widetilde{N}_i^*$ be the number of nodes in each smaller square S_i'. Thus $\widetilde{N}_i^*$ is a Poisson random variable with mean $\frac{K(1-\mu)}{9 \cdot 21(1+\mu)} \log(np(n))$. Similarly to Lemma 2.5, for $K' > \frac{1}{\log(4/e)}$, we have

$$\lim_{n\to\infty} Pr\{\max_i \widetilde{N}_i^* \leq (1+\mu)K' \log(np(n))\} = 1, \forall \mu \in (\mu^{**}, 1), \tag{2.9}$$

where $K' = \frac{1-\mu}{9 \cdot 21(1+\mu)} K$ and μ^{**} is the root of $-\mu^{**} + (1+\mu^{**}) \log(1+\mu^{**}) = \frac{1}{K'}$.

Consider some nodes with radius $R(n) = \sqrt{\frac{K' \log(np(n))}{np(n)}}$, the side length of each smaller square on $\tau_{S'}^{np}$ as shown in Figure 2.5. Every circle is included in a group of at most 9 small squares. From Eqn. (2.9), each circle contains less than or equal to $\frac{K(1-\mu)}{21} \log(np(n))$ nodes with high probability. We can thus construct another graph by connecting each node with its nearest $\frac{K(1-\mu)}{21}\log(np(n))-1$ neighbors, which results in $\mathcal{F}^\lambda(np(n),\frac{K(1-\mu)}{21}\log(np(n))-1)$. Consequently, if $\mathcal{F}^\lambda(np(n),\frac{K(1-\mu)}{21}\log(np(n))-1)$ is asymptotically disconnected, $\mathcal{G}^\lambda(np(n),R(n))$ with $np(n)R(n)^2=\frac{1-\mu}{9\cdot 21(1+\mu)}K\log(np(n))$ is asymptotically disconnected. Note that for large $np(n)$, $\frac{K(1-\mu)}{21}\log(np(n))\gg 1$. Thus there exists $k_1=\frac{1-\mu}{9\cdot 21(1+\mu)}K$ when $c_2=\frac{K(1-\mu)}{21}$. This completes the second step of the proof.

Finally, we want to prove that $\mathcal{F}^\lambda(np(n),c_1\log(np(n)))$ is asymptotically disconnected for $c_1<\frac{(1-\mu)K}{21}$. It suffices to show that for some $\epsilon>0$,

$$\lim_{n\to\infty} Pr\{\mathcal{F}^\lambda(np(n),\epsilon\log(np(n))))\text{ is connected}\}=0.$$

This proof is similar to that in Part I, and is included below for completeness.

According to Lemma 2.5,

$$\lim_{n\to\infty} Pr\{\max_i|\widetilde{N}_i^{np}-K\log(np(n))|\leq \mu K\log(np(n))\}=1.$$

Therefore if we let

$$\begin{aligned}
A_i^{np} &\triangleq \{\text{ No }(\epsilon\log(np(n))+1)\text{-filling event occursin the trap of }S_i^{np}\},\\
Q^{np} &\triangleq \{(k_1,k_2,\cdots,k_{M_{np}}):k_1+k_2+\cdots+k_{M_{np}}=\widetilde{M}^{np},\\
&\qquad\qquad\qquad\qquad\text{where }\widetilde{M}^{np}\geq 0\text{ and }k_i\geq 0,\forall i\},
\end{aligned}$$

then we have

$$\begin{aligned}
&Pr\{\mathcal{F}^\lambda(np(n),\epsilon\log(np(n))))\text{ is connected}\}\\
&\quad\leq Pr\{A_i^{np},\forall i\}\\
&\quad=\Sigma_{(k_1,k_2,\cdots,k_{M^{np}})\in Q^{np}} Pr\{A_i^{np},\forall i;\widetilde{N}_i^{np}=k_i,\forall i\}\\
&\quad=\Sigma_{(k_1,k_2,\cdots,k_{M^{np}})\in Q^{np}} Pr\{A_i^{np},\forall i|\widetilde{N}_i^{np}=k_i,\forall i\}\cdot Pr\{\widetilde{N}_i^{np}=k_i,\forall i\}
\end{aligned}$$

The last step of this proof is the same as the proof of the necessity part in [51], replacing n with $np(n)$.

Note that $K = 1973.9$ and $\mu = 0.032$ was one set of feasible choices from [51]. Therefore, $k_1 < 9.7962$ satisfies $c_1 < 0.074$.

2.6 Asymptotic Connectivity with Coverage

Suppose that *inactive* nodes turn off both the radio transceiver and the sensory device. And let $r(n)$ be the sensing radius. We are interested in a condition for asymptotic connectivity while we give a certain level of coverage.

We consider $\mathcal{G}_p(n, R(n))$. Given that the long term average sleep ratio of a node is $q(n) := 1 - p(n)$, regardless of the distribution of on and off periods (assuming they are both of finite mean which is desirable for coverage purpose), the probability that a given point in a unit square is not covered by any active node in a given time slot is

$$P_u = \sum_{j=0}^{n} q(n)^j \binom{n}{j} (\pi r^2(n))^j (1 - \pi r^2(n))^{n-j} \ .$$

where $n\pi r^2(n)$ is the expected number of nodes deployed within a circle of radius $r(n)$ around the point. The associated joint probability of the point being uncovered and at least one node being within a circle of radius $r(n)$ is

$$P_{u,c} = \sum_{j=1}^{n} q(n)^j \binom{n}{j} (\pi r^2(n))^j (1 - \pi r^2(n))^{n-j} \ .$$

$$\begin{aligned}
\lim_{n\to\infty} \text{ Eqn. } (2.10) &= \lim_{n\to\infty} \sum_{j=1}^{n} q(n)^j \frac{(n\pi r^2(n))^j e^{-n\pi r^2(n)}}{j!} \\
&= \lim_{n\to\infty} e^{-n\pi r^2(n)(1-q(n))}(1 - e^{-n\pi r^2(n)q(n)}) \\
&\quad - \lim_{n\to\infty} e^{-n\pi r^2(n)(1-q(n))} \sum_{j=n+1}^{\infty} \frac{(n\pi r^2(n)q(n))^j e^{-n\pi r^2(n)q(n)}}{j!} \\
&= \lim_{n\to\infty} e^{-n\pi r^2(n)(1-q(n))}(1 - e^{-n\pi r^2(n)q(n)}).
\end{aligned}$$

First equality is from Lemma 2.3. The last equality is because

$$\sum_{j=n+1}^{\infty} \frac{(n\pi r^2(n)q(n))^j e^{-n\pi r^2(n)q(n)}}{j!} \to 0, \text{ as } n \to \infty.$$

Furthermore, we can obtain $q(n) = \frac{\log(V+e^{-n\pi r^2(n)})}{n\pi r^2(n)} + 1$ for fixed $P_{u,c} = V$ when n goes to infinity.

To achieve asymptotic connectivity for active nodes, $np(n)\pi R^2(n) = k_2 \log n$ from Theorem 2.1. This relationship gives us a boundary condition for asymptotic connectivity. Furthermore, a random sleep schedule is $p(n) = -\frac{\log(V+e^{-n\pi r^2(n)})}{n\pi r^2(n)}$ to achieve $P_{u,c} = V$. Therefore, to achieve both,

$$\begin{aligned}
& n[-\frac{\log(V + e^{-n\pi r^2(n)})}{n\pi r^2(n)}]\pi R^2(n) = k_2 \log[-\frac{n\log(V + e^{-n\pi r^2(n)})}{n\pi r^2(n)}] \\
& \Rightarrow -\frac{R^2(n)}{r^2(n)} \log(V + e^{-n\pi r^2(n)}) = k_2 \log[-\frac{1}{\pi r^2(n)} \log(V + e^{-n\pi r^2(n)})].
\end{aligned} \tag{2.10}$$

First, for fixed n and when V is very small, Eqn. (2.10) becomes

$$\begin{aligned}
& -\frac{R^2(n)}{r^2(n)} \log e^{-n\pi r^2(n)} = k_2 \log(-\frac{1}{\pi r^2(n)} \log e^{-n\pi r^2(n)}) \\
& \Rightarrow n\pi R^2(n) = k_2 \log n.
\end{aligned}$$

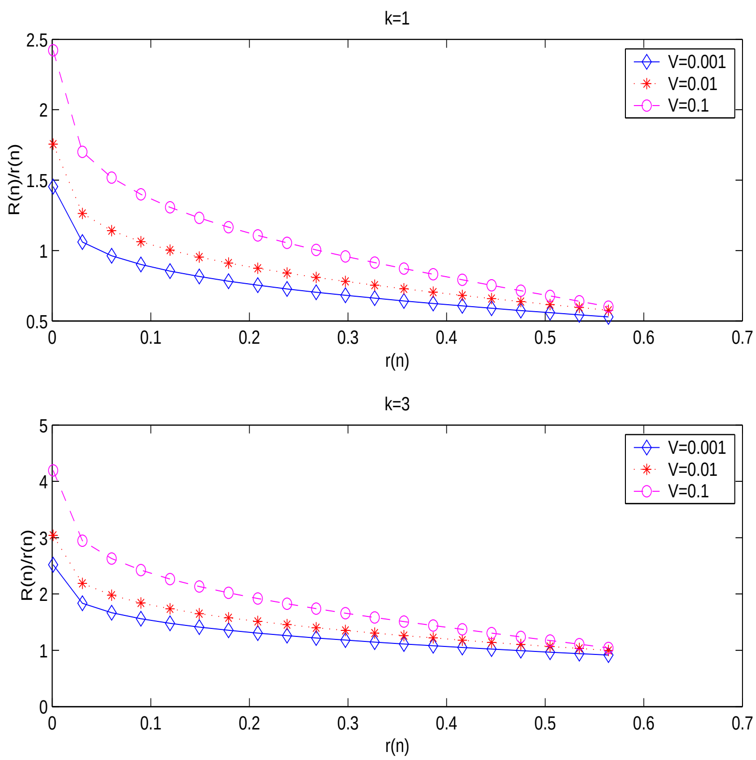


Figure 2.6: Boundary conditions for $\frac{R(n)}{r(n)}$ for p-asymptotic connectivity with some levels of coverage V when $k_2 = 1$ and 3.

Second, for fixed V and when n is very large, Eqn. (2.10) becomes

$$-\frac{R^2(n)}{r^2(n)}\log V = k_2 \log\left(-\frac{1}{\pi r^2(n)}\log V\right)$$

$$\Rightarrow \frac{R^2(n)}{r^2(n)}\log\frac{1}{V} + k_2 \log r^2(n) = k_2\left(\log\log\frac{1}{V} - \log \pi\right) \tag{2.11}$$

We evaluate Eqn. (2.11) to give numerical results in Figure 2.6. It shows boundary conditions for $\frac{R(n)}{r(n)}$ for asymptotic connectivity with some levels of coverage V. Each graph gives a ratio of $R(n)$ to $r(n)$ as $r(n)$ increases for $k_2 = 1$ and $k_2 = 3$.

2.7 Discussion

2.7.1 Generalized Connectivity

Connectivity defined in this chapter is strong in a sense that it requires the existence of a route from each node to every other node in the network. What if we only require any node to be connected to a set of nodes $\mathcal{A}$?

As a simple case, suppose $\mathcal{A}$ contains only one node n_A in the unit square. The network is connected if any node has a route to n_A. Let us construct a spanning tree $\mathbf{T}$ with a root n_A. If the network is connected, every node must be a part of $\mathbf{T}$. This implies that there exists a route from each node to every other node in $\mathbf{T}$. Hence, in this case the condition for connectivity is still $\Theta(\log n)$ neighbors.

Next, supposes that $\mathcal{A}$ contains more than one node in the unit square. Let $\mathcal{A}$ contains k nodes, $n_{A1}, n_{A2}, \cdots, n_{Ak}$. Again, we construct spanning trees $\mathbf{T_i}$ with a root n_{Ai}, and $i = 1, 2, \cdots, k$. Denote by n_i the number of nodes in $\mathbf{T_i}$, where $n_1 + n_2 + \cdots + n_k \geq n$. If they have any common nodes between trees, those trees can be integrated. Eventually, we can find trees which is isolated, $\mathbf{T_i^f}$, $i = 1, 2, \cdots, k_f$. Denote by n_i^f the number of nodes in $\mathbf{T_i^f}$, where $n_1^f + n_2^f + \cdots + n_{k_f}^f = n$. Area in the unit square can be divided into small areas. Since we assumed that nodes are uniformly placed, small areas have same density. Therefore, the sufficient and necessary conditions for this network to be connected remains $\Theta(\log n)$ neighbors.

2.7.2 Phase Transition

[14] showed that probability of connectivity has zero-one transitions for large n in the *random* network with fixed radius $R(n)$. In this chapter, we showed it for large n in the *random* network with fixed radius $R(n)$ and active probability $p(n)$. First, we experiment zero-one transitions in probability of connectivity in the *random* network when $p(n)R^2(n) = \Theta(\frac{\log n}{n})$. Figure 2.7 depicts probability of connectivity when $p(n)R^2(n) = c\frac{\log n}{n}$ with respect to c for $n = 20, 50, 100$ and fixed $p(n) = 0.8$. When n gets larger, the transitions become shaper.

Figure 2.8 depicts the probability of connectivity when $p(n)R^2(n) = c\frac{\log n}{n}$ with respect to c for $n = 20, 50, 100$ and $p(n) = \frac{1}{\sqrt{n}}$. In this case, $R(n)$ decreases to zero as n goes to infinity. Therefore, $np(n)$ goes to infinity and $R(n)$ is bounded. On

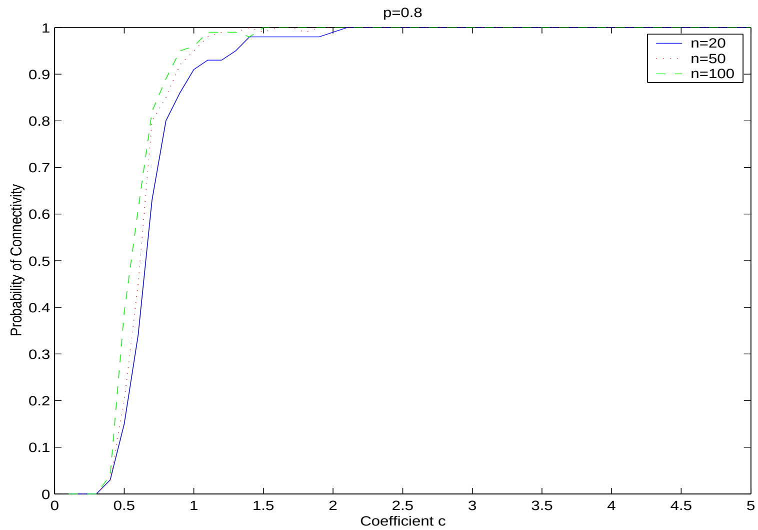


Figure 2.7: Phase transition in probability of connectivity when $p(n)R^2(n) = c\frac{\log n}{n}$ and $p(n) = 0.8$.

the other hand, if $p(n)$ decreases faster than or equal to $1/n$, $np(n)$ does not go to infinity and $R(n)$ becomes unbounded. Therefore, such cases are unable to be satisfied in this network. This is illustrated in Figure 2.9, which depicts probability of connectivity when $p(n)R^2(n) = c\frac{\log n}{n}$ with respect to c for $n = 20, 50, 100$ and $p(n) = \frac{1}{n}$. As c increases, the probability of connectivity never reaches to one.

2.8 Chapter Summary

In this chapter we studied the asymptotic connectivity of a low duty-cycled wireless sensor network where sensor nodes are randomly duty-cycled according to a fixed active probability. We derived the sufficient and necessary conditions for the network to be connected as the number of node grows to infinity. These conditions are in the form of the joint scaling behavior of the number of nodes in the network as well as the active probability. Thus such results reveal how duty-cycling should be scaled as the network gets denser in order to maintain network connectivity.[1]

[1]The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government.

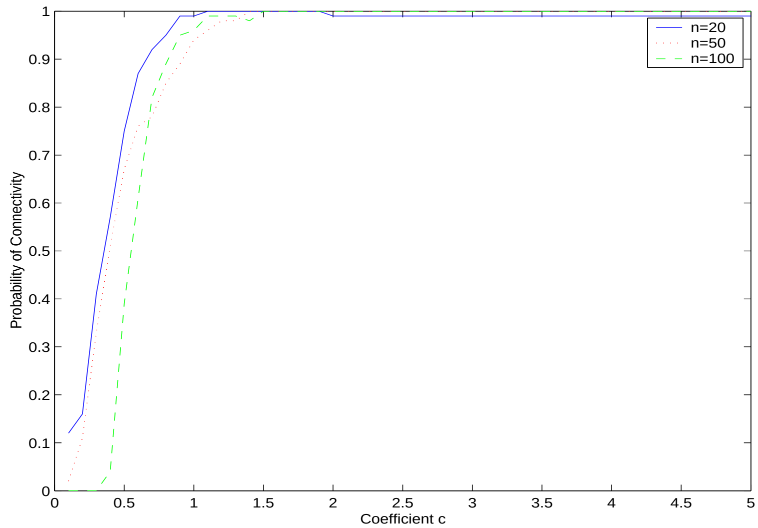


Figure 2.8: Phase transition in probability of connectivity when $p(n)R^2(n) = c\frac{\log n}{n}$ and $p(n) = \frac{1}{\sqrt{n}}$.

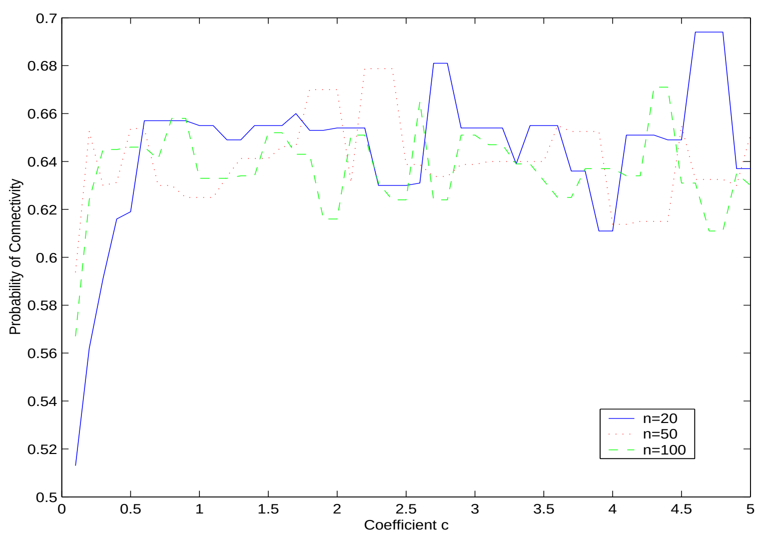


Figure 2.9: Phase transition in probability of connectivity when $p(n)R^2(n) = c\frac{\log n}{n}$ and $p(n) = \frac{1}{n}$.

CHAPTER 3

Optimal Stochastic Routing in Low Duty-cycled Wireless Sensor Networks

3.1 Introduction

For the past decade or so, wireless sensor networks have been extensively studied for a variety of applications: military, environmental, and scientific. In many of these application scenarios, sensors are deployed in large quantities, sometimes in remote areas. Each sensor has the ability to measure and wirelessly transmit data. In order to operate them remotely and autonomously, they are required to be reliable, robust, scalable, and secure among other things. In particular, since they are operated on battery power and are not always easily accessible or maintained in general, energy conservation is critical in keeping such networks long-lasting and useful. As a result, energy efficient design of such networks at all levels, from material to circuit to protocol, has long been a key subject of research and engineering. Low duty-cycling has been widely considered as one of the most effective ways of conserving energy, by periodically turning off sensors that are not actively in use. There are many challenges in designing low duty-cycled wireless sensor networks. The temporary unavailability of sensors can adversely affect both the coverage and connectivity of the network. In addition, duty-cycling causes all kinds of delays, in sensing, detection, and packet delivery (routing).

In this study, we are interested in designing good routing algorithms (measured by low delay) for wireless sensor networks in the presence of very low duty cycles. In particular, we will consider a class of random sleep schedules where sensors go to sleep independent of each other and for a random duration given by a certain probability distribution. In such a scenario, when a node does not have future information on other nodes' sleep schedules but only which of its neighbors are *currently* available, its routing decision (the selection of a neighbor to relay a packet) must properly balance the immediate availability of a node against the future performance of the corresponding route. For instance, we may pre-determine a best route based on average performance (delay) using prior statistics, and at each hop of this route the upstream node simply waits for the downstream node to become available. Alternatively, we can make a state-dependent decision depending on which set of neighboring nodes are available. An extreme example of this latter method is to forward the packet to the earliest available neighbor.

This duty-cycle-related uncertainty is further compounded by the uncertainty in packet transmission. That is, a transmission may succeed or fail depending on channel conditions, which is in general time varying. Again, here a node must weigh the pros and cons of using a pre-determined route and wait at each hop till a transmission succeeds, or can make a forwarding decision depending on which down stream nodes have successfully received the packet (this is possible due to the wireless broadcast medium).

We see that in both cases, one could either choose to perform routing in a deterministic way by selecting a route independent of the sleep state or the success/failure state of the network, or one could try to utilize information available to the nodes in making a closed-loop routing decision. Traditionally, most routing algorithms fall under the former category, see for instance [30, 34, 20, 58, 19, 29, 39, 12], and thus do not react to transmission failure actively. More recently, there have been a number of stochastic routing (also referred to as opportunistic routing) algorithms proposed in the literature [24, 5, 57] to address the uncertainty in transmission. The key idea underlying this latter category is to make routing decisions *after* having observed the outcome of an earlier transmission, i.e., after knowing which down stream nodes have or have not successfully received the transmission. Given different realizations of these transmission events, the actual route taken by a packet can be different, thus the term *event-based routing* or *sample-path dependent routing* [24]. This type of routing algorithms has a clear advantage over traditional deterministic routing in that it takes into account state information available to the nodes.

Compared to the above cited work, our problem also considers the uncertainty due to sleep scheduling, in addition to that due to transmission failure. In this chapter we will adopt the opportunistic routing idea and extend it to the case of low duty-cycle. In particular, we will follow closely the stochastic decision framework developed in [24]. It was shown in [24] that there exists an optimal Markov policy for the above cited problem with time-invariant transmission success probabilities, in the form of a priority policy. For a network of time-varying success probabilities, [24] found necessary and sufficient conditions for a priority policy to be optimal. As we will show, optimal policies for the problem considered in [24] are not in general optimal for low duty-cycled sensor networks because they do not take into account the current sleep state of nodes. In particular, a sender may be forced to wait when a subset of its neighbors are asleep.

The model used in this chapter is an extension to [24] in that it captures the randomness of topology caused by duty-cycling in addition to the randomness in channel conditions. The objective is to seek an optimal routing policy in such networks with respect to performance metrics such as transmission cost and delay, and to resolve the trade-off between these two performance metrics. In subsequent sections we will formally define this optimization problem. Various policies are then explored and characterized for optimality. The main contributions of this chapter are as follows.

1. As a benchmark we develop and analyze a centralized optimal stochastic algorithm for randomly duty-cycled wireless sensor network.

2. We develop a centralized stochastic routing algorithm with reduced state space which performs near-optimal when local sleep/wake states of neighbors are available.

3. We further develop a distributed algorithm utilizing local sleep/wake states of neighbors which performs better than some existing distributed algorithms such as ExOR, etc.

This chapter is organized as follows. In section 3.2, we discuss most relevant work in literature and motivation of our work. Section 3.3 provides the description of the network model with assumptions and definitions. Based on the specified model, we consider the centralized stochastic routing problem with the information of duty-cycles of nodes in the network in Section 3.5. In Section 3.6, we present a centralized stochastic routing algorithm without such duty-cycling information of the entire network to complement weak scalability of the optimal algorithm given in the previous section. We develop a distributed algorithm to compute a policy that resembles the near-optimal centralized algorithm shown in Section 3.7. The performance of algorithms is extensively evaluated in Section 3.8 by self-comparison and cross-comparison. Finally, we conclude in Section 3.9.

3.2 Literature Review

The single most relevant work is by Lott and Teneketzis [24]. As we briefly mentioned in the introduction, this work proposed a general framework for resource allocation and routing in wireless networks. Their network model captures (1) the broadcast nature of wireless communication, and (2) the uncertainty of channel condition via a transmission success probability obtained using analysis or measurement. The problem under consideration is that of anycast, where the goal is for a given packet to reach anyone of a set of destination nodes. Transmissions are costly and a certain reward is obtained if the packet reaches a certain node. The objective is to find a routing algorithm that maximizes the total reward less the total cost.

Two cases were considered in [24]. In the first case the transmission success probabilities are time-invariant, and the resulting optimal routing policy was found to be an index policy, and there is a priority-ordering of nodes that can be computed off-line and their priorities used in routing decisions. Specifically, a node continues to transmit till a higher-priority neighbor receives the packet successfully and becomes the next relay. In addition, this algorithm was shown to lend itself to Djikstra-type of distributed implementation. As we pointed out earlier, the biggest difference between this model and the problem considered in this chapter is nodes' sleep schedules. In the second case considered in [24] the transmission success probabilities are assumed to be time-varying and known ahead of time. One might think that the problem studied in this chapter could be viewed as a special case of time-varying success probability, i.e., in our problem the success probability essentially switches between a positive quantity and zero over time. However, there are some major differences between the model considered in [24] and our model. Firstly, in our case if we were to model on/off as part of the success probability that switches over time, then we do not know the value of these probabilities ahead of time due to the random nature of the duty cycling. Secondly, in our case we actually get to observe the availability of nodes at the current time. As this part of the uncertainty can be eliminated for the current time, we cannot really bundle it together with channel uncertainty. In other words, a single probability quantity, even if time-varying, cannot adequately capture our scenario.

In addition to the more analytical approach discussed above, there are also practical routing algorithms aimed at finding the best possible relay for each transmission. ExOR by Biswas et al [5] is a routing scheme that exploits the broadcast nature of wireless medium by selecting the next forwarder among those which successfully received data after data transmission. This was called *opportunistic* forwarding in [5], and conceptually a very similar idea that that studied in [24] but with a different relay selection criterion. Specifically, while the latter selects a relay based on the priority index, in the former a sender node selects a relay among its neighbors based on a metric called estimated transmission count (ETX) which is the smallest sum of inverse of packet delivery probabilities of links along possible paths to the destination (i.e, this is the smallest estimated number of transmissions it takes to reach the destination along any possible path). The candidates are prioritized by their ETX values. The sender includes the list of candidates with their priorities in a data message. Once some candidates in the list received the data message, they decide whether to forward it by themselves. The fact that such decision is not made by the sender but by receivers is one of the most critical part of the opportunistic routing. The election of a forwarder is performed as follows. If a node in the list receives a data message, it waits for its forwarding timer (which is set according to its priority) to expire and piggybacks the data message while indicating the highest priority of acknowledgements (ACK) it has seen. After all ACKs are sent, each node which received the date message decides to forward it if there is no ACK from the higher priority nodes.

Zhong et al. improved ExOR in two ways [57]. Instead of ETX, the authors defined a new metric called expected any-path transmissions (EAX). EAX captures the expected number of transmissions needed to successfully deliver a data message to destination under opportunistic routing whereas ETX is the expected number of transmissions along a best path with largest delivery probability. In addition, robust acknowledgement mechanism (RACK) increases ACK delivery probability significantly by sending an ACK a certain number of times. This work may be considered as an alternative implementation of Lott and Teneketzis' algorithm.

Our goal is to extend the above works by considering nodes' duty-cycling. In particular, we will use the stochastic optimization framework developed in [24] to derive key properties of an optimal routing algorithm in our problem, and will also borrow from [57] in developing a practical decentralized algorithm.

3.3 Description of the Model and Problem Formulation

We consider a static wireless ad-hoc or sensor network where nodes are duty-cycled independently from one another. Our model is defined to capture some substantial features at the network layer with physical and link layer features included but simplified. We will limit out attention to the delivery of a single packet/message from a source node to a destination node.

3.3.1 A High Level Description

At a high level, the central problem is to find a good (in terms of delay or certain cost measure) route from a source node to a destination node. In a non-duty cycled static network, a typical method is to associate a measure/cost with each link in the network and perform shortest path routing. For instance, if such a cost is unit, then one ends up with a minimum hop-count route; if such a cost is given by the expected number of transmissions over a link (by using a predefined transmission success probability), then the resulting route has the minimum number of expected transmissions. Similar measures can also be defined to take into account factors such as energy consumption.

In our scenario, these nodes are not always available due to duty-cycling, and not available all at the same time. Since a node can potentially obtain the information on whether each of its neighbors are available when a packet needs to be transmitted, a routing decision (i.e., the selection of the next hop relay node) must be made as to whether one should select the node, among all wake nodes, that leads to a minimum cost path, or to wait for a particular node to wake up, who leads to the minimum cost path among all nodes (wake and asleep), or some variations/combinations of these. In this context, it is not immediately clear what principles a good routing algorithm should employ.

In this chapter, we will start by considering a centralized system, where at each instance of time (we assume discrete time) some central agent has the full knowledge of which subset of nodes have already received the message, and which subset of nodes are currently awake. The central agent cannot foresee future sleep state of the nodes, but knows the current sleep state. The routing decision (note that we only consider single-path routing) at each time step then reduces to the question of (1) among the set of nodes that have already received the message, which one should be selected as the relay node to retransmit the message, and (2) whether we should simply do nothing, wait for one time step, and reconsider the decision at the next time. This in essence is the routing decision problem we seek to address in this chapter. For this centralized version of the problem we will derive the structural properties of the optimal routing policy and construct an algorithm that computes such a policy. To reduce the computational complexity we will further propose a sub-optimal routing algorithm and is considerably simpler.

We then consider a distributed implementation of the above sub-optimal algorithm, whereby each node only has access to local information, effectively resulting in a decentralized routing algorithm. Specifically, in this case a node only knows who among its neighbors have received the message, and who among its neighbors are currently awake or asleep. A node then must decide on its own, based on such local information whether it should serve as a relay for the message it receives. Because such decisions are made by individual nodes in a decentralized fashion, it is possible that multiple nodes may decide to relay the same message. This distributed implementation is accomplished via packet exchange and certain local information update procedure; details are provided in subsequent sections.

Below we state formally the assumptions and notations used in this chapter.

3.3.2 Assumptions

- We will focus on the routing of a single message originated from somewhere in the network and has a single destination node. Under the stochastic routing framework, since the routing is sample-path dependent, each message may follow a different path. Thus by this assumption we are ignoring possible interaction or interference introduced by simultaneous message transmissions or subsequent messages in the same stream.

- We consider a discrete time system, where in each time step (or time slot) a node is active/awake with a time-invariant probability, independent of other time slots and other nodes. For simplicity in our derivation we will assume that this *active* probability is the same for all node, though they need not be. The complement of active probability is also called the *sleep* probability.

- We further assume in addition to the previous assumption, that any node that has successfully received the message will remain awake. This assumption is adopted for simplicity in presentation in our analysis. In practice, we only need to ensure that the node who is designated as the relay should stay awake till the next hop/relay receives the message successfully.

- A transmission between a sender i and a receiver j has a time-invariant probability p_{ij} of being successful, independent of other transmission attempts. If this success probability is nonzero, then j is called a "neighbor" of i. This probability does not have to be symmetric between two nodes.

- A transmission and its acknowledgment (ACK) from successful receivers occur within a single time slot.

- Our routing problem is classified as anycast: there is a set of nodes to one of which the message needs to reach, but it may not matter which node gets it as long as one of them does. This reflects the situation where a message from a sensor needs to be delivered to one of several gateway nodes.

3.3.3 Notations

A summary list of notations used in this chapter is as follows.

N is the number of nodes in the network.

$\Omega = \{1, \cdots, N\}$ is the set of all nodes. So, $|\Omega| = N$.

I is a nonexistent node which represents the idle action.

q_{ij} is the transmission success probability from node i to node j, given that both nodes are awake. As stated earlier, j is called a neighbor of i if $q_{ij} > 0$.

p is the active probability for all nodes.

(W, A) refers to a state of the system, where $W \subseteq \Omega$ and $A \in \{0,1\}^N$. W is defined as the set of nodes that have received the message. A is defined as the sequence of sleep(0)/active(1) status of all nodes. In particular, node i is awake if it has received a message as stated in assumptions: Given $A = \{a_1, a_2, \cdots, a_N\}$, $a_i = 1$ for all $i \in W$.

$F(W)$ denotes a feasible set of all possible sleep/active states A induced by W so that A is consistent with W. More specifically, given W, there are a total of $2^{N-|W|}$ sets of A's in $F(W)$ where $a_i = 1$ for all $i \in W$ and $a_i \in \{0,1\}$ for all $i \in \Omega - W$.

$F(W|W', A')$ for $W \subset W', A' \in F(W')$ denotes a subset of sleep/active states $A \in F(W)$, such that that A is identical to A' except that $a_i \in \{0,1\}$ for all $i \in W' - W$.

$F(W|W', A')$ for $W \supset W', A' \in F(W')$ denotes a subset of sleep/active states $A \in F(W)$, such that that A is identical to A' except that $a_i = 1$ for all $i \in W - W'$. We see that there is only one such A in this set.

$T : 2^\Omega \rightarrow 2^N$ is defined as a mapping from W to a vector $T(W) = \{w_1, w_2, \cdots, w_N\}$, $W \subseteq \Omega$ where each element $w_i = 1$ if node i has received the message, and 0 otherwise.

$P^i(W', A'|W, A)$ indicates the probability of state (W', A') reached from state (W, A) by choosing i for transmission, $i \in W$. Let $T(W) = \{w_1, w_2, \cdots, w_N\}$

and $A = \{a_1, a_2, \cdots, a_N\} \in F(W)$. Also, $T(W') = \{w'_1, w'_2, \cdots, w'_N\}$ and $A' = \{a'_1, a'_2, \cdots, a'_N\} \in F(W')$. If a node i is chosen for transmission, the transition probability is given by

$$P^i(W', A'|W, A) = \left(\prod_{\forall j: w_j=0, a_j=1, w'_j=1} q_{ij}\right) \cdot \left(\prod_{\forall j: w_j=0, a_j=1, w'_j=0} 1 - q_{ij}\right) \cdot \left(\prod_{\forall j: a_j=0, w'_j=1} 0\right) \cdot p^{I_{\overline{a}'} - I_{\overline{w}'}} (1-p)^{N - I_{\overline{a}'}}, \tag{3.1}$$

for $\forall i \in W$,

where $I_{\overline{w}'}$ is the number of 1's in $T(W')$, and $I_{\overline{a}'}$ is the number of 1's in A'. If the idle node I is chosen,

$$P^I(W', A'|W, A) = \begin{cases} p^{I_{\overline{a}'} - I_{\overline{w}'}} (1-p)^{N - I_{\overline{a}'}}, & \text{if } W' = W \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

$R : 2^{\Omega} \to \mathbb{R}$ is the reward functions. Specially, we denote $R_i = R(\{i\})$.

π is a Markov policy such that π depends only on the current state (W, A). We write $\pi(W, A) = i$ to indicate that policy π transmits at node i when in state (W, A), $i \in W$. We write $\pi(W, A) = I$ to indicate policy π choose the idle/wait action. We write $\pi(W, A) = r$ to indicate policy π retires and receives reward $R(W) = r$ when in state (W, A).

$V^{\pi}(W, A)$ is the expected reward when starting in state (W, A) under policy π.

3.3.4 Problem Formulation

Problem 3.1. We consider the transmission of a packet in a low duty-cycled wireless network of N nodes, where each node is active with probability p, described above. At each time instant the central controller chooses among three actions: (1) select a node among nodes that have the packet for the next transmission; (2) wait for the next time step; and (3) terminate the routing process. It acts at the beginning of each time slot with the knowledge of the set of nodes which have received the message and the set of current active nodes in the network. The transmission from a node i costs $c_i > 0$ and is the local broadcast to its active neighbors. The idle action, denoted by $i = I$, costs $c_i = \alpha \geq 0$, a penalty on idle waiting. This transmission is successfully received by a neighbor j with a time-invariant probability p_{ij} given node j is active during that time slot. Each transmission event is assumed to be independent of another. The objective is to choose the right action at each time step and the right time to terminate the process so as to maximize the total expected reward less cost:

$$E\{R(S_f) - \sum_{t=1}^{\tau-1} c_{i(t)}\}, \tag{3.3}$$

where τ is the stopping time when the transmission process is terminated, S_f is the state at τ, and $i(t)$ is the node (including idle action) chosen by the policy at time t.

3.4 Preliminaries

Below we present a number of definitions that will be helpful in exploring important properties of an optimal Markov policy for the problem outlined above. When nodes are always awake (i.e., $p = 1$), which is a special case of Problem 3.1, the authors of [24] have shown that an optimal Markov policy is both a priority policy and an index policy. The first few definitions below are reproduced from [24] for this thesis to be self-contained. These explain what a priority or an index policy is. We then present an example to illustrate they are not able to capture the extra dynamics introduced by node sleeping. This motivates us to define generalized versions of priority policies and index policies, respectively.

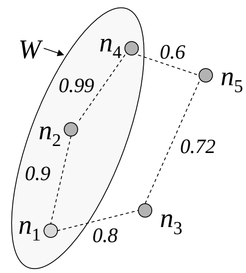


Figure 3.1: System for an Example 1.

Definition 3.1. [24] A Markov policy π is a priority policy if there is a strict priority ordering of the nodes s.t. $\forall i \in \Omega$ we have $\pi(S \cup \{i\}) = \pi(\{i\}) = i$ or r_i, $\forall S \subseteq \Omega_i$, where Ω_i is the set of nodes of priority lower than i.

Definition 3.2. [24] A function $f : 2^{\Omega} \to \mathbb{R}$ is an index function on Ω if f satisfies

$$f(S) = \max_{i \in S} f(\{i\}), \quad \forall S \subseteq \Omega. \tag{3.4}$$

Definition 3.3. [24] A priority policy π is called an index policy if $V^{\pi}(\cdot)$ is an index function on Ω.

Below we use a simple example to show that the above definitions cannot be directly applied to Problem 1; in other words, an optimal policy may not be found in the class of priority policies for Problem 1.

Example 3.1. A Case where an Optimal Markov Policy cannot be a Priority Policy

We consider a system depicted in Figure 3.1, where $\Omega = \{1, 2, 3, 4, 5\}$ and nonzero transmission success probabilities between nodes. Assume that $R_i = 0$ except node 5 which has a reward $R_5 > 0$. For simplicity we also assume that $c_i = 1$ for $i \in \Omega$. Let us consider first the case where nodes are not duty cycling. An optimal policy can be found by applying Lott's algorithm. For instance, when $W = \Omega$, it is trivial to see that the optimal action is to retire and receive R_5. Any W that includes node 5 results in the same decision as above; node 5 will thus be considered to have the *highest priority* among all nodes. When $W = \{1, 2, 3, 4\}$, the optimal decision is for node 3 to transmit. Similarly, the optimal decision given any W that includes node 3 is always to select node 3 for transmission. Node 3 thus has the highest priority among all nodes except for 5. If we take nodes 5 and 3 away from the set W, then node 4 becomes the optimal decision, with the next highest priority, regardless of the membership of the rest of the set. Eventually, by repeating this process until W becomes empty, we end up with an ordered list of nodes, in descending order of their priorities. For this particular example, the priorities are such that the ordered list is nodes 5, 3, 4, 1, 2 from the highest to the lowest. The result is called a priority policy because there exists such a priority list that is independent of the actual state of the system, and that the optimal decision is based on this priority list: choose the highest priority node among W for the next transmission.

Now, we consider the case where nodes are duty-cycling with active probability $p = 0.1$. In addition, we assume that the idling cost is 1, i.e., $c_I = 1$. In this example, an active node i is denoted by ia and a sleeping node i by is. As mentioned in the previous subsection, nodes in W are assumed to be awake. Therefore, only nodes to be concerned for on/off states are the nodes in $\Omega - W$, i.e., $A \in F(W)$. Let $W = \{1, 2, 4\}$ as shown in the Figure 3.1. Let π^* to be an optimal Markov policy. We have $\pi^*(W, \{3a, 5a\}) = 4$, $\pi^*(W, \{3a, 5s\}) = 1$, $\pi^*(W, \{3s, 5a\}) = 4$, and $\pi^*(W, \{3s, 5s\}) = I$ based on the calculation given in Appendix A. Let us focus on $A = \{3a, 5s\}$. In this case, node 1 seems to be the highest priority node among nodes $1, 2$, and 4. Now, suppose $W = \{1, 2\}$. For the sleep/wake states in $F(W|\{1, 2, 4\}, \{3a, 5s\})$, we obtain $\pi^*(W, \{3a, 4a, 5s\}) = 2$ and $\pi^*(W, \{3a, 4s, 5s\}) = 1$ by the calculation similarly done in Appendix A. When node 4 is in sleep, node 1 is the highest priority node as expected. On the other hand, when node 4 is active, node 2 is the highest priority node among node 1 and node 2. In other words, node 1 is not always the highest priority node among nodes $1, 2$, and 4 but nodes' priorities may change with sleep states of nodes.

Remark 3.1. As can be seen from the above example, removing a node like 4 from the set $W = \{1, 2, 4\}$ has a significant impact on the resulting optimal policy, even though it is not the highest priority node given $A = \{3a, 5s\}$. This is because node 4 is the highest priority node in W given other sleep/wake states such as $\{3a, 5a\}$ and $\{3s, 5a\}$. To summarize, given W, if a node i is the highest priority node in W for some feasible sleep/wake state, then the priority ordering in $W - \{i\}$ are not always preserved under other sleep/wake states. Thus if we remove node i, then we need to recalculate the priority ordering of nodes in $W - \{i\}$. By contrast, in the case when $p = 1$, this priority ordering is preserved no matter which node we remove from the set W. This is the primary difference between Problem 1 and that considered in [24] both from a conceptual and a computational point of view.

Motivated by the above example, it is necessary to generalize the preceding definitions in the context of our problem.

Definition 3.4. Consider a Markov policy π such that $\pi(W, A_i) = n_i \in W \cup \{I\}, \forall i \in \{1, \cdots, m\}$ for $W \subseteq \Omega$ and $\forall A_i \in F(W)$ where $m = 2^{N-|W|}$. This policy is called a Generalized(G)-priority policy if the following condition holds: Define $N_W = \bigcup_{i=1}^{m} n_i - \{I\}$ and for $\forall S \subseteq W - N_W$, we have

$$\pi(W, A_i) = \pi(S \cup N_W, A) = n_i, \forall A \in F(S \cup N_W | W, A_i), \quad \forall i \in \{1, \cdots, m\},$$

where the condition on A is simply to ensure that the sleep state A is consistent with state A_i (it is identical to A_i except for nodes in $W - S - N_W$ what are unspecified). What this definition says is that a policy is a G-priority policy if there exists a set

N_W of priority nodes within W whose priorities are strictly higher than the rest regardless of the sleep state, but whose priority ordering among themselves can only be determined for a specific sleep state. This set consists of nodes that would have been selected in at least one sleep state.

Definition 3.5. A function $f : 2^\Omega \times 2^N \to \mathbb{R}$ is an Generalized(G)-index function on 2^Ω if f satisfies

$$f(W, A) = \max_{\tilde{W} \subseteq W, \tilde{A} \in F(\tilde{W}|W,A)} f(\tilde{W}, \tilde{A}), \quad \forall W \subseteq \Omega, \forall A \in F(W) \ . \tag{3.5}$$

Definition 3.6. A priority policy π is called an Generalized(G)-index policy if $V^\pi(\cdot)$ is an G-index function on Ω.

3.4.1 Special Cases of Problem 3.1

There are two special case interpretations of Problem 1 depending on what we use as costs.

The case of $c_I = 0$

If the idle cost is zero, there is no penalty on waiting. In this case, there is no loss of optimality to always wait till all nodes are awake (a positive probability event) and then make a decision on who is to transmit. If we only consider the problem in this particular sleep state (all awake), i.e., we wait in other states, then the problem becomes identical to the one studied and solved in [24].

The case of $c_i = c_I = c$

If all costs are the same, the problem can be regarded as finding a policy which minimizes delay. Assuming the transmission of a packet consumes a certain amount of time and so does waiting, each cost can be translated into a time unit. Therefore, the problem is to find a policy that minimizes the sum of the time slots taken.

3.5 Analysis of Problem 3.1

In this section, we analyze Problem 3.1 and derive structural properties of an optimal policy π^*. As mentioned earlier, we will take a centralized point of view and assume that at each time instant, a decision-maker has complete information on the time-invariant transition probabilities and the current sleep/wake state. We will then use these properties to construct optimal and sub-optimal routing policies. In a later section we will discuss distributed implementations of these.

Our system of Problem 3.1 can be modeled by a two-dimensional finite state Markov chain. That is, each decision is made based on current state (W, A) where state space is finite. Hence, we limit our attention to Markov policies. One may use stochastic dynamic programming to find an optimal Markov policy. However, its computational complexity is high. For instance, suppose that the number of nodes in the network is N and $|W| = n$. Given W, there are 2^{N-n} A's in $F(W)$ and $n+1$ actions, one for each node in W plus I. For each pair (W, A_i), $A_i \in F(W)$, its optimal value function requires the optimal value functions for other sleep/wake states $(W, A_j), \forall A_j \in F(W)$. All these optimal value functions are solved simultaneously by setting the action for each (W, A_j). Thus, the number of such combinations is $(n+1)^{2^{(N-n)}}$ for given W. And there are $\frac{N!}{n!(N-n)!}$ W's for $|W| = n$. Therefore, the total number of calculations is

$$\sum_{n=1}^{N} \frac{N!}{n!(N-n)!} (n+1)^{2^{(N-n)}}. \tag{3.6}$$

As N grows, the complexity grows rapidly. For this reason, instead of applying stochastic dynamic programming directly, we will investigate the structural properties of an optimal Markov policy, which are then used to construct algorithms with lower complexity.

We next show that there exists an optimal G-index policy for Problem 3.1 in Theorem 3.1. To summarize, the proof of Theorem 3.1 is to show that an optimal Markov policy with certain properties is a G-priority policy, which is in turn a G-index policy by proving that the expected reward function is a G-index function. We then propose an algorithm to find an optimal G-index policy and discuss its computational complexity. It should be noted that this method follows closely the framework developed in [24] although there are technical differences due to the introduction of sleep states.

Unless otherwise noted, all missing proofs may be found in the appendix.

The proof of Theorem 3.1 utilizes some useful lemmas presented in the following. Lemma 3.1 below is essentially the same as given in [24], but adapted to our notation. It shows that an optimal Markov policy has the property that if all supersets that can be reached from a state have optimal expected reward values and the actions at the state for all sleep states are optimal, then the expected reward value at the state is optimal.

Lemma 3.1. *Let π^* be an optimal Markov policy for Problem 3.1. Suppose we are given W_1 and $A_1 \in F(W_1)$, and let π be a Markov policy with the following properties:*

$$V^{\pi}(W, A) = V^{\pi^*}(W, A), \quad \forall W \supset W_1, \forall A \in F(W), \tag{3.7}$$

$$\pi(W_1, A_1) = \pi^*(W_1, A_1), \quad \forall A_1 \in F(W_1). \tag{3.8}$$

Then

$$V^{\pi}(W_1, A_1) = V^{\pi^*}(W_1, A_1). \tag{3.9}$$

The following lemma shows the monotonicity of an optimal Markov policy.

Lemma 3.2. *In Problem 3.1, let π^* be an optimal Markov policy. Let $W_1, W_2 \subseteq \Omega$ and $W_2 \subseteq W_1$. Then, for $A_1 \in F(W_1)$, $V^{\pi^*}(W_2, A_2) \leq V^{\pi^*}(W_1, A_1)$ where $A_2 \in F(W_2|W_1, A_1)$.*

In the next lemma, we show the properties of an optimal Markov policy, specifically the G-priority structure.

Lemma 3.3. *Let π^* be an optimal Markov policy for Problem 3.1. Then, there exists a Markov policy π which has the following properties.*

1. *For all $W \subseteq \Omega$ where $|W| \geq 2$ and all possible $A_i \in F(W) = \{A_1, \cdots, A_m\}$, $m = 2^{N-|W|}$,*

$$\pi(W, A_i) = n_i \in W \cup \{I\} \Rightarrow \pi(W - \{j\}, A) = n_i,$$
$$\forall j \in W - \cup_{i=1}^m n_i, \forall A \in F(W - \{j\}|W, A_i), \tag{3.10}$$
$$\pi(W, A_i) = r_{n_i}, n_i \neq I \Rightarrow \pi(W - \{j\}, A) = r_{n_i},$$
$$\forall j \in W - \cup_{i=1}^m n_i, \forall A \in F(W - \{j\}|W, A_i). \tag{3.11}$$

2. *For all $W \subseteq \Omega$ where $|W| \geq 2$ and all possible $A_i \in F(W)$, and $\pi(W, A_i) = n_i \in W \cup \{I\}$ or $r_{n_i}, n_i \neq I$ for $i \in \{1, \cdots, m\}$,*

$$V^{\pi}(W-\{j\}, A) = V^{\pi}(W, A_i) = V^{\pi^*}(W, A_i) = V^{\pi^*}(W - \{j\}, A),$$
$$\forall j \in W - \cup_{i=1}^m n_i, \forall A \in F(W - \{j\}|W, A_i). \tag{3.12}$$

3. *π is an optimal Markov policy.*

In the following lemma, we show that an optimal markov policy has the expected reward that is a G-index function.

Lemma 3.4. *For any optimal Markov policy π^*, $V^{\pi^*}(\cdot)$ is a G-index function on $\Omega \cup \{I\}$.*

Theorem 3.1. *There is an optimal Markov policy π^* for Problem 3.1 which is a G-index policy.*

Proof. By Lemma 3.3, there exists a Markov policy π^* which is an optimal Markov policy. $V^{\pi^*}(\cdot)$ is a G-index function by Lemma 3.4. This says that the optimal decision on the resulting set after removing some nodes that are not in $\bigcup_i n_i$ from W remains the same. Thus the conditions in Definition 3.4 are satisfied. Thus π^* is a G-priority policy. Since π^* is a G-priority policy and its $V^{\pi^*}(\cdot)$ is a G-index function, π^* is a G-index policy according to Definition 3.6. □

3.6 Optimal and Sub-Optimal Routing Algorithms

3.6.1 An Optimal Centralized Algorithm for Problem 3.1

We present an algorithm to compute the optimal G-index policy for Problem 3.1. Compared to the brute-forth dynamic programming, our algorithm utilizes the properties of G-index policy stated in Lemma 3.3 to reduce the number of computations. Let node d be the destination. The procedure starts with $W = \Omega$ and $A = \{1, \cdots, 1\}$. Its optimal action and reward value are straight-forward, which are

$$V(\Omega, A) = R_d \text{ and } \pi(\Omega, A) = r_d.$$

From the properties 1 and 2 in Lemma 3.3, we know

$$V(\Omega - \{j\}, A) = R_d \text{ and } \pi(\Omega - \{j\}, A) = d,$$

for $\forall A \in F(\Omega - \{j\})$ if $j \neq d$. Thus, we only need to calculate $V(\Omega - \{d\}, A)$ for $\forall A \in F(\Omega - \{d\})$.

By solving the associated set of linear equations, we obtain $\pi(\Omega - \{d\}, A)$ for $\forall A \in F(\Omega - \{d\})$. Suppose $\pi(\Omega - \{d\}, A_i) = n_i$ for each i s.t. $A_i \in F(\Omega - \{d\})$. Let us denote by $D(\Omega - \{d\}) = \cup_i \{n_i\}$ the set of highest priority nodes in W. Again, by

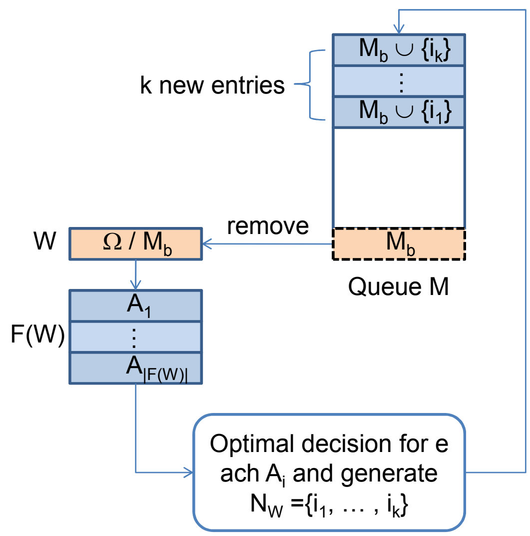


Figure 3.2: The diagram of Algorithm 3.1.

the properties Lemma 3.3, we have

$$\pi(S \cup D(\Omega - \{d\}), A) = n_i, \forall S \subset \Omega - \{d\}, A \in F(S \cup D(\Omega - \{d\})|\Omega - \{d\}, A_i).$$

Therefore, the reward functions that need to be calculated are $V(\Omega - \{d\} - \{n_i\}, A)$, for $\forall A \in F(\Omega - \{d\} - \{n_i\})$. The subsequent steps are done similarly as above.

We now formally describe the above procedure in Algorithm 3.1. Figure 3.2 illustrates how Algorithm 3.1 works. Note that this algorithm is presented for a single destination, but can be easily extended to the case of multiple destinations.

Algorithm 3.1. Define sets W, $F(W)$, N_W and a queue M, as follows.

Each entry in queue M contains the set of nodes $S \in \Omega$ which have not received the packet. Specially, denote by M_b the head of line of M. W is the complement of M_b with respect to Ω, which is $W = \Omega - M_b$ meaning the set of nodes which have received packet. $F(W)$ is the set of all feasible active(1)/sleep(0) states of the nodes in M_b and all ones for the nodes in W. That is, $F(W) = \{A_1, A_2, ..., A_k\}$ where $k = 2^{|M_b|}$. N_W is the set of highest priority nodes in W for every $A_i \in F(W)$.

Since the case where $W = \Omega$ is trivial, we start with $W = \Omega - \{d\}$. Initially, the queue $M = \{M_b\} = \{\{d\}\}$ contains a destination d; the action taken by an optimal G-index policy π on the destination d is to retire and receive R_d regardless of sleep states. $F(W)$ contains two sets which include ones for all nodes except for d which is zero in one set and one in the other. N_W is initially empty.

The algorithm proceeds as follows.

1. For each $i \in W$ and each $A_j \in F(W)$, let π_i^j be an G-index policy with the same priority list as π for the nodes in M_b, with i as the next highest priority node after M_b, and with the priority of the nodes, W- $\{i\}$ arbitrary, but lower than i. Compute $V_i^{\pi_i^j}(W, A_j)$ for all j, $1 \leq j \leq k$ from

   (3.13)
   $$V_i^{\pi_i^j}(W, A_j) = \max\{-c_i + \sum_{W' \supseteq W} \sum_{A' \in F(W')} P^i(W', A'|W, A_j) V^{\pi_i^j}_{\pi_i^j(W', A')}(W', A'), R_i\}.$$

2. For an idle node I to choose, let π_I be an index policy which is similarly defined as in step 1 except no actual transmission to take place. Thus,

   $$(3.14)\quad V_i^{\pi_I^j}(W, A_j) = \max\{-\alpha + \sum_{A' \in F(W')} P^I(W, A'|W, A_j) V^{\pi_I^j}_{\pi_I^j(W, A')}(W, A'), R_i\}.$$

3. For each set of choices of a node $i_j \in W \bigcup \{I\}$ for A_j, $1 \leq j \leq k$, denoted by $\mathbf{i} = \{i_1, i_2, ..., i_k\}$, $V_{i_j}^{\pi_{i_j}^j}(W, A_j)$ are solved by k linear equations. Choose $\mathbf{i}$ with the highest values of $V_{i_j}^{\pi_{i_j}^j}(W, A_j)$'s. Ties are broken with more Is in $\mathbf{i}$, otherwise arbitrarily.

4. N_W includes all distinct $y \in \mathbf{i}$, which is not equal to I. For each node in N_W, append it to the set M_b and place the resulting set on top of the queue M.

5. Finally, remove M_b from the bottom of the queue M. If M is empty, stop. Otherwise, go to step 1).

We now prove the optimality of Algorithm 3.1 for Problem 3.1 in the following theorem.

Theorem 3.2. *For Problem 3.1, Algorithm 3.1 produces an optimal G-index policy.*

Proof. We prove this theorem by induction. Let π be an optimal G-index policy. The base case holds because Algorithm 3.1 initially assigns nodes with highest reward into M which is optimal. Suppose Algorithm 3.1 is running at the point where $|M_b| = L$. Then, $F(W) = \{A_1, A_2, ..., A_k\}$ where $k = 2^L$. Let $i \in W \cup \{I\}$ be actual $(L+1)$th highest priority node for A_j according to π whether retiring or not. Consider $l \in W, l \neq i$. Let π_l denote the priority policy that has the same priority as π in the first L nodes and have node l be the $(L+1)$th highest priority node for A_j. Then, we want to show

$$V_i^{\pi_i^j}(W, A_j) = V_i^{\pi}(W, A_j) \geq V_l^{\pi}(W, A_j) \geq V_l^{\pi_l^j}(W, A_j). \tag{3.15}$$

The first equality comes from the assumption that node i has $(L+1)$th highest priority for A_j according to π and the definition that π_i^j has the same priorities for the first L nodes as π. The first inequality comes from the assumption that the priority of node i is higher than the one of node l. The second inequality is because π is optimal. Eqn. (3.15) shows that for each A_j, node $i \in W$ maximizing $V_i^{\pi_i^j}(W, A_j)$ is assigned as the $(L+1)$th highest priority node. This completes the induction step and the proof. □

It is worth noting that utilizing the structure of an optimal Markov policy reduces the computational complexity required in finding an optimal policy for Problem 3.1. Whereas the computational complexity of directly using stochastic dynamic programming is given by Eqn. (3.6), the complexity of Algorithm 3.1 is upper bounded

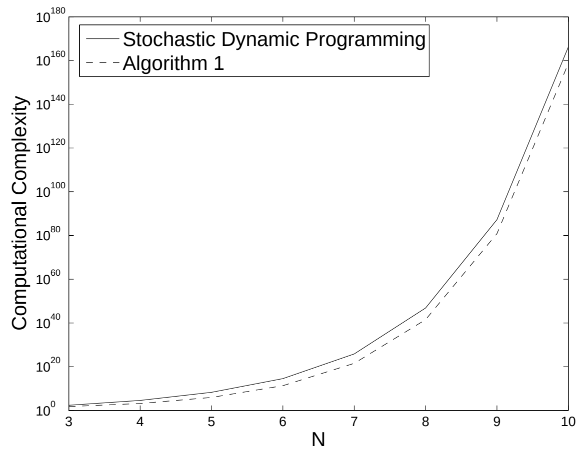


Figure 3.3: Computational complexity of a stochastic dynamic programming algorithm.

by

$$\sum_{n=1}^{N} (n+1)^{2^{N-n}} \prod_{m=n}^{N} \min(2^{N-m}, m).$$

In above equation, $\prod_{m=n}^{N} \min(2^{N-m}, m) \leq N!/n!(N-n)!$. Figure 3.3 shows the computational complexity of Algorithm 3.1. As you can see, its complexity is still too high.

3.6.2 A Sub-Optimal Algorithm

Algorithm 3.1 is not very scalable. In this section we modify Model (M) to maintain a simpler state of the system (i.e., W only) rather than (W, A). Accordingly, a change to the assumptions in Subsection 3.3.2 is made with respect to the information available to the decision-maker. In this section, it is thus assumed that the decision-maker has the knowledge of the nodes which received a message and time-invariant transition probabilities but no information on the sleep/wake status of all nodes. In the following, we redefine some notations for Model (M) while others remain the same as given in 3.3.2.

The state of the system is determined by W only.

$P^i(W'|W, A)$ indicates the probability of state W' reached from state W by choosing $i \in W$ for transmission, when nodes' sleep/wake status is A at the moment.

If a node i is chosen for transmission, the transition probability is defined as

$$P^i(W'|W,A)$$

(3.16)

$$= \left(\prod_{\forall j: w_j=0, a_j=1, w'_j=1} q_{ij} \right) \cdot \left(\prod_{\forall j: w_j=0, a_j=1, w'_j=0} 1 - q_{ij} \right) \cdot \left(\prod_{\forall j: a_j=0, w'_j=1} 0 \right), \quad \text{for } \forall i \in W,$$

where q_{ij} is the probability that j receives the message from i if both awake and $I_{\overline{w}'}$ is the number of 1's in $T(W')$. Note that the idle node I is never be chosen.

π is a Markov policy such that π depends only on the current state W. We write $\pi(W) = i$ to indicate policy π transmits at node i when in state W, $i \in W$. We write $\pi(W) = r$ to indicate policy π retires and receives reward $R(W)$ when in state W. $\pi(W) = r_i$ is written as shorthand that policy π retires and receives reward $R_i(W)$, $i \in W$.

$V^\pi(W)$ is the expected reward when starting in state W under policy π.

Given the modified model described above (i.e., without nodes' active/sleep information), the problem is reduced to the one studied in [24] with a modification to the state transition probability. This is because under the above assumptions the decision-maker cannot differentiate transmission failures caused by channel errors from the ones by duty-cycling. Hence, sleep/wake activity of nodes is reflected in transition probability measured on average, i.e., $P^i(W'|W) = \sum_{A \in F(W)} P^i(W'|W,A)P(A)$. Given such transition probabilities, [24] presented an algorithm which produces an optimal index policy under this model. In other words, the algorithm, referred to in this chapter as *Lott's Algorithm*, is optimal in the case where the sleep/wake states of nodes are unobservable. However, it is not hard to see that Lott's algorithm may not be optimal for Problem 3.1 because it uses less information. This was also demonstrated in Example 3.1 which highlights the possibility that a priority policy cannot be optimal for our problem (Note that Lott's algorithm produces an index policy which is a priority policy as well). Under Lott's Algorithms, the expected reward given W when i is transmitting is calculated by

$$(3.17)\quad V_i^{\pi_i}(W) = \max\{-c_i + \sum_{W' \supseteq W} \left(\sum_{A \in F(W)} P^i(W'|W,A)P(A) \right) V^{\pi_i}_{\pi_i(W')}(W')\}, R_i\}.$$

In the following, we present an algorithm that outperforms Lott's Algorithm for our problem while maintaining the simple state W (compared to (W,A)) as in Lott's Algorithm. Specifically, the decision maker has access to the sleep/wake states A at the time of transmission, but its calculation of the expected reward is based only on W. This significantly simplifies the computation.

Algorithm 3.2. The sets W, $F(W) = \{A_1, A_2, ..., A_k\}$, N_W, M_b and the queue M are defined the same way as in Algorithm 3.1.

The algorithm consists of two parts: an off-line part and an on-line part. The off-line part obtains the expected reward values $\tilde{V}(W)$ for all $W \subseteq \Omega$ by Lott's Algorithm. The on-line part of the algorithm proceeds as follows.

1. For each $i \in W$, let π be a policy with the same priority list as the policy generated by Lott's Algorithm for the nodes of M_b with i as the next highest priority node after M_b, W - $\{i\}$ arbitrary, but lower than i. Compute $V_i^{\pi}(W, A_j)$ for all j, $1 \leq j \leq k$ from

$$(3.18)\qquad V_i^{\pi}(W, A_j) = \max\{-c_i + \sum_{W' \supseteq W} P^i(W'|W, A_j)\tilde{V}(W'), R_i\}.$$

2. When selecting the idle action its value is computed as:

$$(3.19)\qquad V_I^{\pi}(W, A_j) = \max\{-\alpha + P^I(W|W, A_j)\tilde{V}(W'), R_I\}.$$

3. For A_j, choose a node $i_j \in W \bigcup \{I\}$ with highest values of $V_i^{\pi}(W, A_j)$, $1 \leq j \leq k$, denoted by $\mathbf{i} = \{i_1, i_2, ..., i_k\}$. Ties are broken arbitrarily.

4. For each distinct $y \in \mathbf{i}$, which is not equal to I, append $\{y\} \bigcup M_b$ at the top of M. Remove M_b from the bottom of M.

5. If M is empty, stop. If not, go to step 1.

Unlike Lott's Algorithm, Algorithm 3.2 takes an action dependent on A. It re-computes the priorities of nodes in W with consideration of sleep/wake status at the time of transmission and chooses a node with highest modified priority for the next transmission. This algorithm cannot perform better than Algorithm 3.1 by definition. However, below we show it does at least as good as Lott's Algorithm in the following corollary.

Corollary 3.1. *Algorithm 3.2 performs at least as good as Lott's Algorithm for Problem 3.1.*

Proof. For notational simplicity, R_i is not included in the following expected reward equations. In addition, we simply denote by $\tilde{V}(W)$ the expected reward calculated by Lott's Algorithm. In particular, $F(W) = \{A_1, A_2, \cdots, A_{|F(W)|}\}$. Suppose node i^* achieves the maximum of the expected reward given in Eqn. (3.17). In order words, node $i^* \in W$ is chosen by Lott's Algorithm. Then,

$$
\begin{aligned}
i^* &= \arg\max_i \{-c_i + \sum_{W' \supseteq W} \left(\sum_{j=1}^{|F(W)|} P^i(W'|W, A_j) P(A_j) \right) \tilde{V}(W')\}\} \\
&= \arg\max_i \{ \sum_{j=1}^{|F(W)|} \left(-c_i + \sum_{W' \supseteq W} P^i(W'|W, A_j) \tilde{V}(W') \right) P(A_j)\}. \qquad (3.20)
\end{aligned}
$$

From Algorithm 3.2, we obtain $i_j \in W \cup \{I\}$ which achieves the maximum of the expected reward equation given in Eqn. (3.18) for each A_j. For each j,

$$i_j = \arg\max_i \{-c_i + \sum_{W' \supseteq W} P^i(W'|W, A_j)\tilde{V}(W')\}. \tag{3.21}$$

For those A_j such that $i_j = i^*$,

$$-c_{i^*} + \sum_{W' \supseteq W} P^{i^*}(W'|W, A_j)\tilde{V}(W') = -c_{i_j} + \sum_{W' \supseteq W} P^{i_j}(W'|W, A_j)\tilde{V}(W').$$

And for those A_j such that $i_j \neq i^*$,

$$-c_{i^*} + \sum_{W' \supseteq W} P^{i^*}(W'|W, A_j)\tilde{V}(W') \leq -c_{i_j} + \sum_{W' \supseteq W} P^{i_j}(W'|W, A_j)\tilde{V}(W').$$

Therefore, we have

$$\begin{aligned}
\tilde{V}(W) &= \sum_{j=1}^{|F(W)|} \left(-c_{i^*} + \sum_{W' \supseteq W} P^{i^*}(W'|W, A_j)\tilde{V}(W') \right) P(A_j) \\
&\leq \sum_{j=1}^{|F(W)|} \left(-c_{i_j} + \sum_{W' \supseteq W} P^{i_j}(W'|W, A_j)\tilde{V}(W') \right) P(A_j) = \sum_{j=1}^{|F(W)|} V(W, A_j)P(A_j).
\end{aligned}$$

Hence, Algorithm 3.2 performs better than Lott's Algorithm on average. □

3.7 Distributed Implementation

In this section, we develop a practical routing protocol that implements Algorithm 3.2 in a distributed way. We will adopt opportunistic-like forwarding used in [5] in our algorithm where nodes are not assumed to have perfect information on W and A. Specifically, nodes periodically exchange a HELLO (also referred to as a beacon packet in the sequel) packet when they are awake. From these exchanges nodes infer about their neighbors' sleep status when making a decision on whether they should forward a received packet.

Our stochastic routing protocol, referred to as SRP below, consists of two elements: priority update and forwarder selection. In priority update a node has the option of recalculating the priorities of its neighbors. Recall that in Algorithm 3.2

we first compute the nodes' priorities off-line, ignoring the current sleep state, using Lott's algorithm. These will be referred to as the *off-line priorities.* In SRP, nodes can choose to update these off-line priorities and recalculate as they obtain their neighbors' sleep state via the HELLO packets. In the forwarder selection step a node decides for itself whether it should become a forward and retransmit the packet it received based on current priorities. Below we present these two elements in more detail.

3.7.1 Priority Update Procedure

In this subsection, we describe how the offline priorities are set and updated in SRP.

An active node i transmits a short HELLO packet periodically[1]. This HELLO packet contains explicit information on measured channel quality and implicitly conveys the fact that the sender of the HELLO packet is active. In addition, it contains an updated value of node i's priority $V^n(i)$, calculated as follows.

Initially, $V^0(i)$ for all i is obtained based on Lott's Algorithm off-line. Recall that the optimal policy obtained by Lott's Algorithm is an index policy (i.e., $\tilde{V}^\pi(W) = \tilde{V}^\pi(\{i\})$ if i is the highest priority node under π in W). As part of initialization, we assign $V^0(i) = \tilde{V}^\pi(\{i\})$ to node i at the start of the algorithm; $\tilde{V}^\pi(\{i\})$ is also written as $\tilde{V}_i^\pi$ below for simplicity.

This quantity is then updated before node i sends out each beacon within a single wake period, and is reset to $V^0(i) = \tilde{V}^\pi(\{i\})$ upon waking up from a sleep period. Specifically, right before the n-th beacon transmission at time t_n^i, node i updates $V^n(i)$ and includes its value in the beacon packet. Note that the transmission times

[1]HELLO packets are commonly used for neighborhood discovery, a mechanism employed by virtually all routing protocols to maintain fresh information on which nodes are one's neighbors. In this sense our protocol simply utilizes an existing mechanism and the exchanged state information gets a free ride.

of the beacon packets are unsynchronized among nodes in the network; a node's beacon transmission times are only relevant to its latest wake-up time. Thus, t_n^i for node i might be different from t_n^j for node j. Node i recalculates $V^n(i)$ based on updates received from active neighbors during the time interval $[t_{n-1}^i, t_n^i]$. In addition, node i maintains a candidate set denoted as C_i, which is a subset of neighbors of node i that contains all possible forwarders, e.g., nodes whose current priorities are higher than i's. Initially, C_i contains the nodes with higher initial priorities (determined by $V^0(\cdot)$) than i's. This set may change over time depending on the priority updates.

The more precise details are given in the following description of the priority update procedure, followed by a particular node i. We will assume that the off-line computation of $\{\tilde{V}_i^\pi\}$ by Lott's Algorithm is completed, such that each nodes has its own $\tilde{V}_i^\pi$ as well as $\tilde{V}_j^\pi$ for all nodes j in its neighbor set N_i. This can be accomplished using the Dijkstra-like distributed algorithm proposed in [24], in which case this computation is off-line only in the sense that this computation is done prior to the execution of SRP.

1. When node i goes to sleep, it turns off the radio and does nothing.

2. Upon waking up, node i sets the beacon counter n to zero, the beacon transmission time t_0^i to current time, and immediately transmits a beacon packet containing value $V^0(i)$ which is set to $\tilde{V}_i^\pi$. $V_i^0(j)$ is initialized to $\tilde{V}_j^\pi$ for all $j \in N_i$; the set A_i that contains all active neighbors is initialized to be an empty set. The set C_i of forwarder candidates contains the set of neighbors j's who have $\tilde{V}_j^\pi > \tilde{V}_i^\pi$.

3. Node i then increments n by one, and set the next beacon transmission time t_n^i to $t_{n-1}^i + T$, where T is the (constant) beacon interval.

4. Between t^i_{n-1} and t^i_n, if node i receives a beacon packet from some neighbor j, it updates $V_i^{n-1}(j)$ with the new value contained in the packet and records its update time. Also, node j is added to A_i if it is not already in the set.

5. Right before the n-th beacon transmission, at time t^i_n, node i recalculates the priorities as follows. If a beacon packet from node j was last received at a time earlier than $t^i_n - \beta T$, where β a constant multiplier and βT sets a threshold on how long a neighbor has not been heard from before assuming it's asleep, then node j is assumed to be in sleep mode and is removed from A_i. For those nodes in A_i, set $V_i^n(j) = V_i^{n-1}(j)$. Otherwise, set $V_i^n(j) = \tilde{V}_j^\pi$ for a sleep node j. Include in C_i all neighbors that qualify as a possible forwarder and their current priorities. Denote by $q^*_{ij|C_i,A_i}$ the probability that node j receives successfully while nodes with higher priorities in $A_i \bigcap C_i$ fail. Denote nodes with higher priorities than node j by $\{A_i \bigcap C_i\}_j^+ \subset A_i \bigcap C_i$. Then,

$$q^*_{ij|C_i,A_i} = q_{ij} \prod_{k \in \{A_i \bigcap C_i\}_j^+} (1 - q_{ik}).$$

   Using this probability, node i updates $V^n(i)$ as follows.

$$V^n(i) = \frac{-c_i + \sum_{j \in A_i \bigcap C_i} q^*_{ij|C_i,A_i} V_i^n(j)}{1 - \sum_{j \in A_i \bigcap C_i} (1 - q_{ij})}.$$

   Node i then transmits a beacon packet with $V^n(i)$ to its neighbors.

6. While node i continues to be awake, repeat steps 3-5.

Remark 3.2. Relationship between T and an "on" duration: We assume that an on duration is larger than a beacon interval T. The length of an on duration obviously affects the accuracy of recalculation of $V^n(i)$.

3.7.2 Forwarder Selection Procedure

When an upstream forwarder or relay, say node k, sends out the message, it contains a list of potential forwarders C_k. When node i receives the message within its n-th beacon interval, $[t^i_{n-1}, t^i_n]$, it first checks to see if it is included in the set C_k. If it is, it waits for a certain time period to see if it hears any ACKs from higher priority nodes. This time period is randomly chosen but inversely related to its own priority position in C_i. If it does, then node i will not transmit the message. If it fails to get any ACK from higher priority nodes during the period, it transmits the message containing the list of candidates as the next forwarders in the message. The details of this forwarder selection procedure are provided in the following. This algorithm is performed whenever node i generates a message or receives it from one of its neighbors.

1. Recall that $V(i)$ and $\{V_i(j)\}_{j \in N_i}$ are set to current priority values calculated by the priority update procedure. The current active neighbors of node i, A_i, is also given in priority update.

2. When node i receives a message, it obtains the list of candidate forwarders. If it is on the list, go to step 3. Otherwise, it does not forward the message and returns to the receiving mode.

3. If node i is listed as a potential forwarder, it calculates a time period D based on its priority on the list. If it is the k-th highest priority node on the list with a total of M nodes on the list, it randomly selects D as proportional to $k - 1$. Or an ACK is repeated like the multiple duplicated ACKs as robust acknowledgement introduced by [57].

4. If node i receives ACKs from higher priority nodes, it transmits an ACK with the

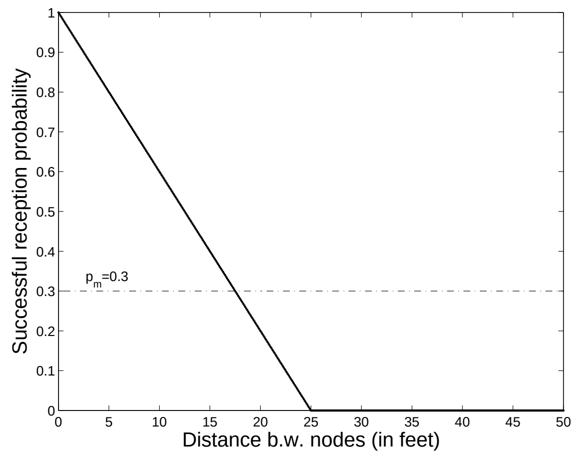


Figure 3.4: Delivery success probability w.r.t. distance.

identity (ID) of the highest priority node, and it does not forward the message. During the period D, if node i does not receive an ACK from any of the higher priority nodes, node i decides to forward and transmits an ACK with its own ID. The message contains the priority list of the next forwarders according to $V(i)$, $\{V_i(j)\}_{j \in N_i}$, A_i.

5. If node i decides not to forward under the policy π and receives no ACK during $M \cdot T_s$ period, it goes to step 3, unless it was already repeated for R times. If so, the message is removed.

6. If node i has transmitted the message, it waits ACKs from neighbors for at most $R \cdot T_s$. If it receives no ACK, it retransmits the message.

3.8 Performance Evaluation

We have performed extensive MATLAB simulation to evaluate the performance of the proposed algorithms. The simulated system closely follows the set of assumptions listed earlier in this chapter. Here we reiterate some of the more relevant ones. The lossy channel model we adopted in the simulation is based on pair-wise distance. Specifically, we assume that the success probability that a node receives a message from any node is given by a linear function of the distance between the nodes as shown in Figure 3.4. This distribution is based on the measurements on Rene Motes using medium transmission power reported by Ganesan et al in [13]. In general, a node with non zero reception probability is regarded as a neighbor. However, we also eliminate nodes with poor reception probability (those lower than a threshold p_m) from a neighboring set. Each sensor node is duty-cycled with a sleep probability p_s, and the discrete time unit is chosen large enough for a transmission and ACKs to occur. A source and a destination are randomly selected among nodes in the network. A node that has received a message does not go back to sleep again till the simulation ends. We assume that the network is connected when all nodes are awake, thus in time any destination may be reached from any source.

Throughout this section, we consider three different scenarios depending on how the transmission cost and idle penalty are determined.

1. *Unit cost for both transmission and idle action*: Under this scenario the problem reduces to finding a delay-optimal path from a source to a destination. Note that the term *delay* used in this chapter accounts for the number of time units taken to reach the destination considering hop counts and retransmissions caused by channel errors. With this cost scenario we may also find a path that minimizes energy consumption, given that the normalized energy consumption in transmission is roughly the same as that in idle waiting.

2. *Random cost for transmission and nonzero cost for idle action*: With this cost scenario the problem finds a path that minimizes the total cost. Because both transmissions and waiting are costly, there may be a tradeoff between minimizing the number of transmissions and minimizing delay. For instance, a path may incur the smallest number of transmissions (e.g., a shortest path when all

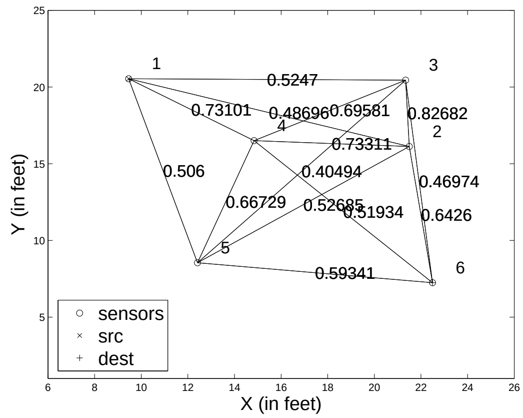


Figure 3.5: Topology 1: an example of a network topology with 6 nodes.

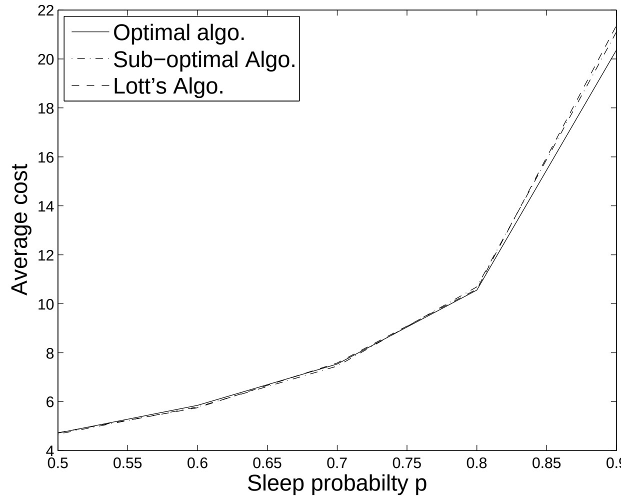
transmission success probabilities are equal) but may involve a large amount of waiting. The combined cost may render this path not as desirable. The tradeoff between transmission energy consumption and delay can be adjusted through setting the respective costs. The intention of using a random transmission cost is so that this cost may represent the fact that some transmissions are more costly if the transmitting node has relatively low residual energy, or if all its neighbors are located far away thereby physically requiring more energy.

3. *Random cost for transmission and zero cost for idle action*: In this case the problem looks for a cost-efficient path without having to worry about penalty on waiting. Since there is no penalty on waiting, there is no loss of optimality for a policy to simply wait till all nodes are awake and then make the decision on who is to relay. In this sense Lott's Algorithm would be an optimal algorithm, i.e., it is optimal to wait till all nodes are awake and then apply Lott's Algorithm.

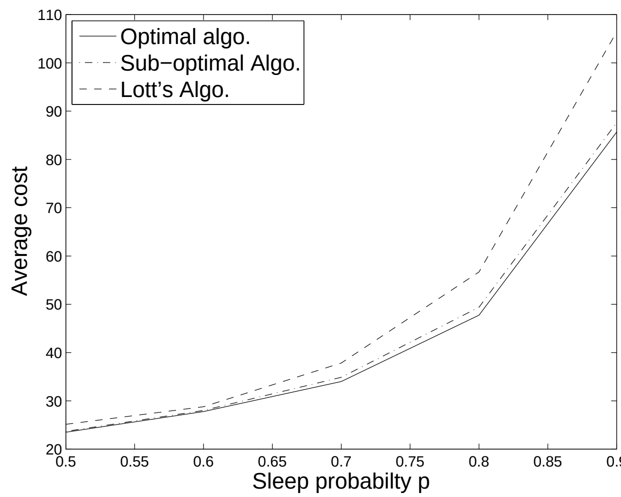
3.8.1 The effect of sleep information on optimality

In the previous sections, it was shown that Algorithm 3.1, referred to as the Optimal Algorithm in the remainder of this section, generates an optimal G-index policy for Problem 3.1. Unfortunately, its computational complexity is extremely high and thus is not really usable even for a small network. We did manage for sizes up to $N = 6$. The network topology under consideration is a small network of 6 sensor nodes with average node degree 4.6 and $p_m = 0.3$ as shown in Figure 3.5, referred as Topology 1. Based on this topology, we first examine how much performance degradation will result if we ignore sleep information. In Figure 3.6 we compare Algorithm 3.1, Lott's Algorithm which requires no sleep information, and Algorithm 3.2 (also referred to as the sub-optimal algorithm in the remainder of this section) that utilizes the current sleep state in making forwarding decisions.

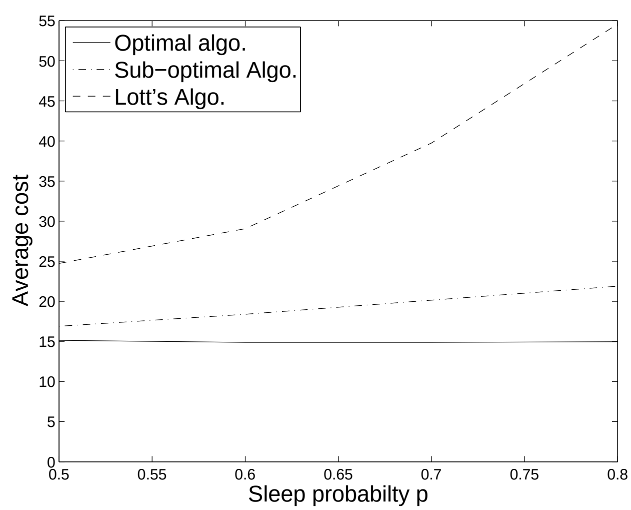
Figure 3.6 depicts the average costs of paths taken by these algorithms when different cost distributions are applied. When all costs are the same and normalized to unit as in the first cost scenario, average path cost is identical to average delay. When p is relatively small up to 0.8, average delays of all three algorithms are virtually indistinguishable as illustrated in Figure 3.6(a). As p becomes very high (0.9), the Optimal Algorithm shows a slight advantage. In the second cost scenario, nodes' costs are uniformly generated over $[1, 7]$ while idle cost is fixed at 4. As shown in Figure 3.6(b), it is remarkable that the Sub-optimal Algorithm performs as good as the optimal one. This indicates that the Sub-optimal Algorithm which is much simpler and requires only local sleep/wake information than the optimal algorithm works sufficiently well in such a small network. In the third cost scenario, nodes' transmission costs are generated by the same distribution as above but no costs are imposed on the idle action. The third scenario is meant for the applications that are extremely delay-tolerant. Figure 3.6(c) shows that the average costs of Optimal Algorithm and Sub-optimal Algorithm are almost unaffected by the increase in sleep probability by taking a large number of idle actions and waiting for the right moment to transmit. The average cost of Optimal Algorithm is exactly the same while waiting



(a) Scenario 1 with unit costs.



(b) Scenario 2 with random cost and nonzero idle action penalty.



(c) Scenario 3 with random cost and zero idle action penalty.

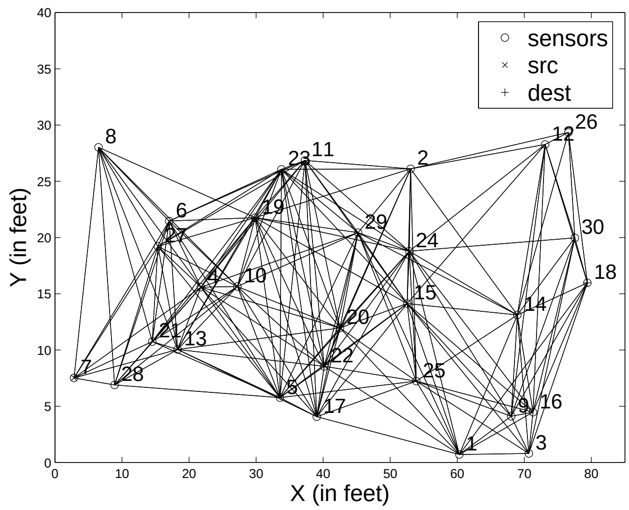
Figure 3.6: Performance comparison of the centralized algorithms on Topology 1

delay increases exponentially as p increases. Note that the average cost of Lott's Algorithm is invariant to the change in idle cost since it never selects the idle action.

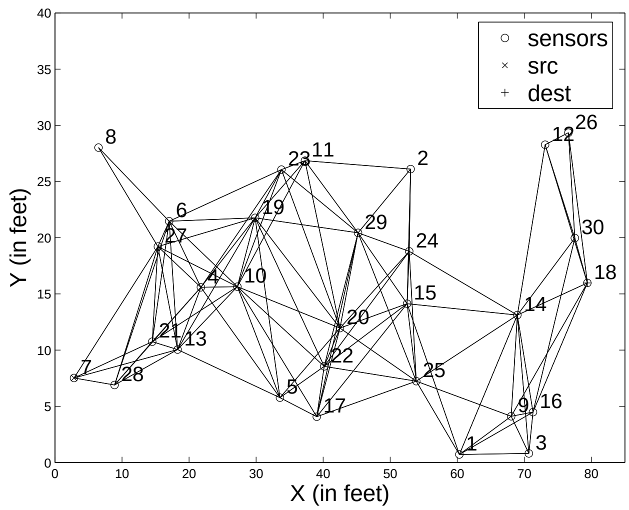
3.8.2 The effect of node degree

If a node has more neighbors, given a sleep probability it is more likely to have more wake neighbors. However, even in a highly connected network, a best neighbor is not always on. Thus, whether to transmit now or wait for better neighbors to be on is not straight-forward depending on which neighbors are awake at the time of transmission. We focus on the performance comparison of Lott's Algorithm and Sub-optimal Algorithm when increasing the average node degree in the next set of results. We consider a network where $N = 30$ sensor nodes are deployed with different $p_m = \{0, 0.3, 0.5\}$ as shown in Figure 3.7. p_m determines the set of neighbors and so does node degree.

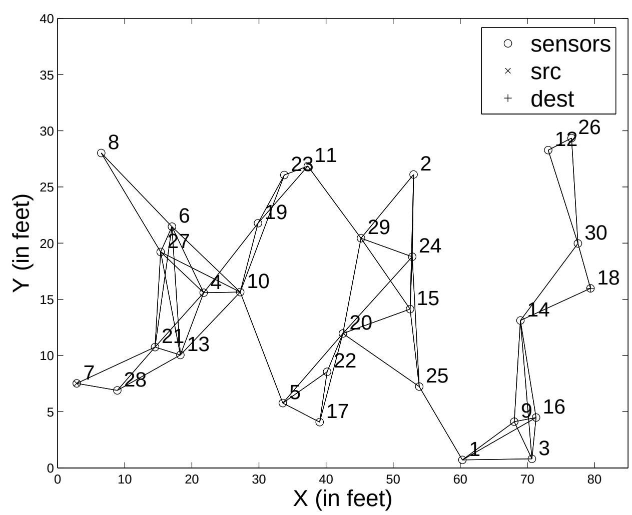
Using the third cost scenario, as the degree of nodes increases, Figure 3.8(a) shows Sub-optimal Algorithm improves significantly compared to Lott's Algorithm, and the improvement is more pronounced as p increases. That is, the Sub-optimal Algorithm is more effective when duty-cycling is heavy and less so otherwise. This is because there are sufficient number of wake neighbors around, which makes idle action unnecessary. Figure 3.8(b) depicts that delay performance of Sub-optimal Algorithm is slightly better than that of Lott's Algorithm, which is desirable in many applications. As depicted in Figure 3.8(c), Lott's Algorithm takes no idle action while Sub-optimal Algorithm takes more idle actions as p increases or node degree reduces. That is, Lott's Algorithm took more hops to reach the destination whereas Sub-optimal Algorithm waited for better neighborhood to wake up but not too long while taking less hops instead.



(a) Topology 2 with average node degree = 12.33 when $p_m = 0$.

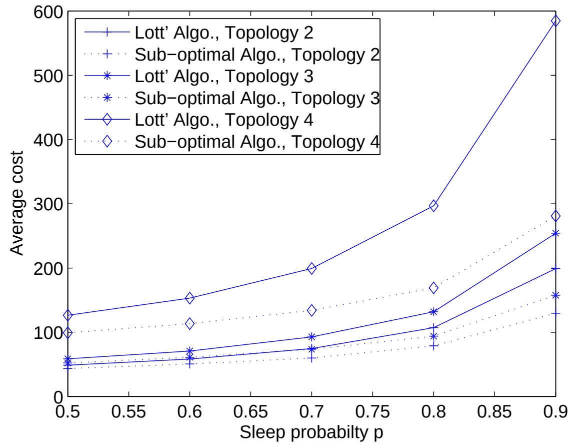


(b) Topology 3 with average node degree = 7.13 when $p_m = 0.3$.

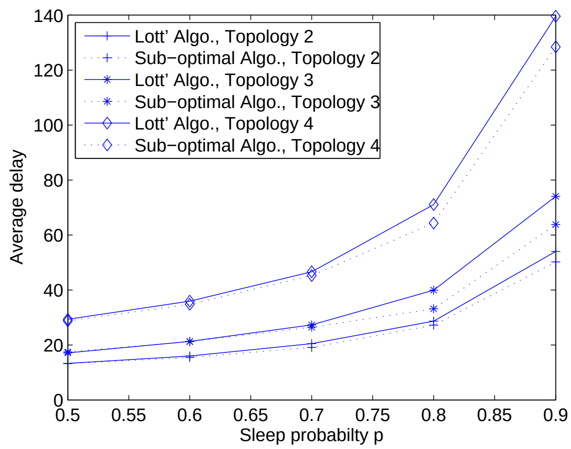


(c) Topology 4 with average node degree = 4.13 when $p_m = 0.5$.

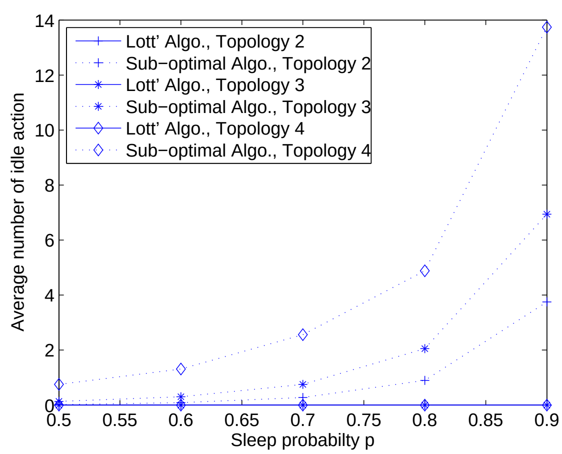
Figure 3.7: Topologies with 30 sensor nodes.



(a) Average cost.



(b) Average delay.



(c) Average number of idle actions.

Figure 3.8: The effect of average degree of nodes on the performance of Sub-optimal and Lott's Algorithms (scenario 3).

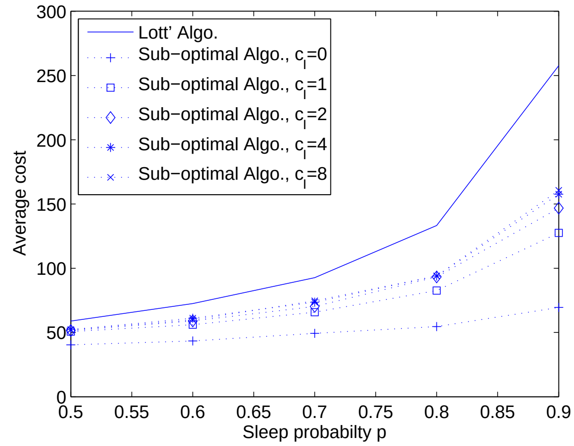
3.8.3 The role of idle costs

As described above, Lott's Algorithm is invariant to changes in idle cost. In this subsection, we examine more clostly Sub-optimal Algorithm on Topology 3 while varying the idle cost by selecting it from the set $c_I = \{0, 1, 2, 4, 8\}$.

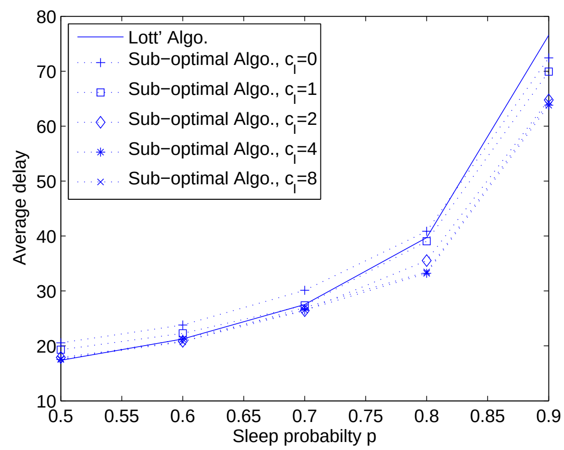
Figure 3.9 depicts the average cost of Algorithm 4 with various c_I. As c_I grows, the average cost tends to increase but the average delay decreases. Thus, there is a trade off between cost and delay. c_I is a design parameter to be adjusted for the intended applications. In Figure 3.9(c), note that when $c_I = 8$ no idle action is taken. This means that there will be no difference in performance if c_I increases beyond 8. One may try to find c_I^* to satisfy its cost efficiency and delay constraint (e.g., over the range of $[0, 8]$ in this particular example). In this example, the worst delay performance where $c_I = 8$ was not very bad due to the high average node degree. However, in a less connected network, it could be an important parameter that may need to be chosen with care.

3.8.4 The performance of the distributed protocol SRP

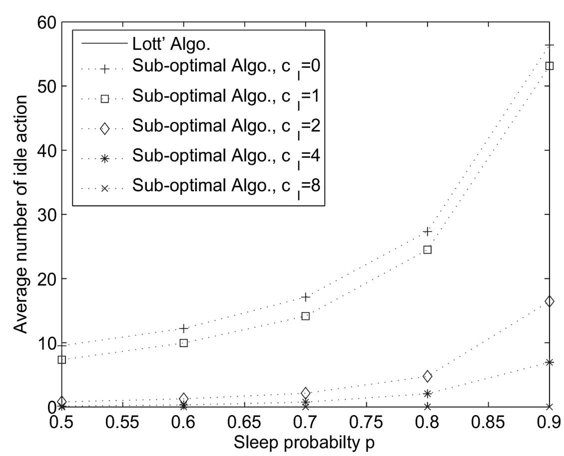
We evaluate the performance of SRP on Topology 3 with 30 nodes and $p_m = 0.3$ as illustrated in Figure 3.7(b). As described in Section 3.7, the distributed algorithm's access to sleep state is limited to a node's 1-hop neighbors, which is obtained from the beacons broadcasted by neighbors every T time unit. In our simulation, T is set to 2. Each node's sleep schedule is generated by a geometric distribution with mean length of on periods of 4. Given the scenarios of cost distributions introduced earlier, we examine the performance of SRP described in Section 3.7 comparing with one of the most promising algorithms in the literature. Specifically, we consider a few variations of ExOR with different forwarder selection metrics: 1) the number



(a) Average cost.



(b) Average delay.



(c) Average number of idle actions.

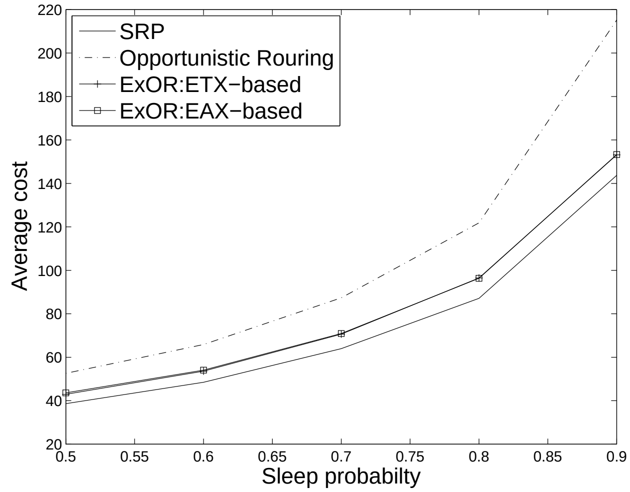
Figure 3.9: The role of wait cost on the performance of Sub-optimal Algorithm on Topology 3.

of hops to best-path and loss rate [4], 2) ETX [5], and 3) EAX [57]. We provide cross-comparison between our algorithm and three different versions of ExOR. For the simulation, 300 packets are randomly generated in the network during 3000 time units. Each node has a finite queue so that the total delay takes into account queueing delay in addition to hop counts and the number of waiting decisions.

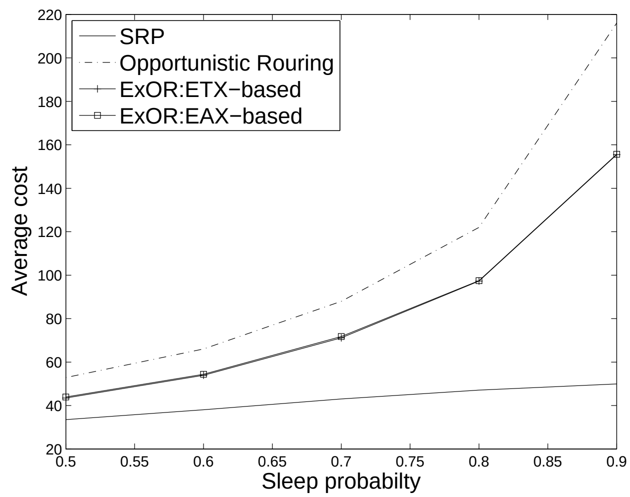
Figure 3.10(a) depicts the average cost of these algorithms when nodes' costs are distributed uniformly with a mean 4 and idle cost is 4. ExOR, which is known to outperform traditional routing where packets are sent to the pre-computed path with the smallest costs, performs the worst among them in the figure. Other versions of ExOR using ETX and EAX metrics performs better than the original ExOR. On the other hand, under scenario 3 with the same distribution for nodes' costs and no wait cost, Figure 3.10(b) shows that the average cost of SRP is the minimum with the largest delay. In Figure 3.10(c) SRP exhibits the same performance of delay as ExORs with ETX and EAX. Overall, our algorithms outperform ExORs in terms of average cost with reasonable delay performance.

3.9 Chapter Summary

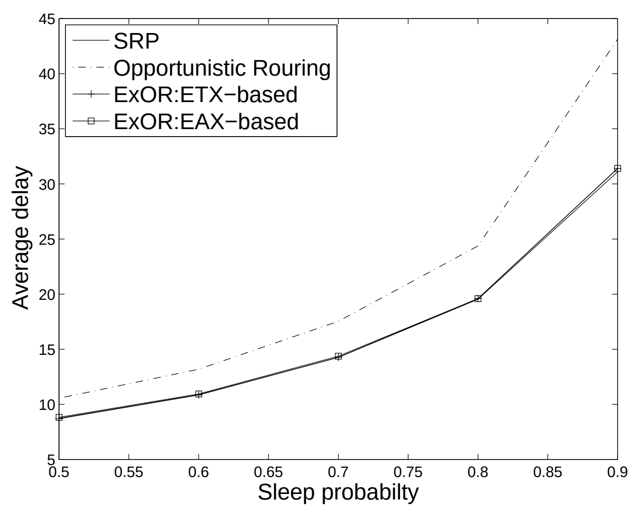
We studied a routing problem in wireless sensor networks where sensors are randomly duty-cycled. We developed an optimal stochastic routing framework in the presence of duty-cycling as well as unreliable wireless channels. Using this framework, we presented and analyzed an optimal centralized stochastic routing algorithm, and then simplified the algorithm when only local sleep/wake states of neighbors are available. We further developed a distributed algorithm utilizing local sleep/wake states of neighbors which performs better than some existing distributed algorithms such as ExOR.



(a) Average cost. (scenario 2)



(b) Average cost. (scenario 3)



(c) Average delay.

Figure 3.10: Performance comparison between the decentralized algorithms and ExORs (scenario 3).

CHAPTER 4

Opportunistic vs. Non-opportunistic Routing: A Delay Analysis

4.1 Introduction

Routing has long been a subject of extensive research for wireless ad hoc and sensor networks. Traditionally, routing algorithms may be classified as proactive [30, 34], reactive or on-demand [20, 58, 19], and hybrid [16, 22], by differentiating whether a route is established before or after communication needs arise. They may be classified as single-path and multi-path, judged by how many routes are simultaneously maintained. Routing algorithms can also be classified by the underlying selection metric for a route: greedy geographical routing [29, 39, 12] selects for each hop the relay that is the closest to the destination; shortest path routing selects the one with the smallest number of hops; lifetime-maximizing routing [7, 6, 40, 54] selects an energy efficient one, etc.

Due to interference and random fading, a certain amount of uncertainty in wireless data communication is a reality that cannot be ignored. In this context, all the above routing algorithms share a common feature: in making routing decisions they either do not take into account link quality (e.g., in the form of a transmission success probability), or do so through the expectation, e.g., by calculating the expected number of transmissions needed on a given link [8]. These methods are inherently *deterministic* in the sense that the routing decisions are typically made regardless of the actual outcome of data transmissions (i.e., success or failure), or more precisely, they are made *before* data transmissions are performed and observed.

By contrast, during the last few years we have seen a number of studies proposing an *opportunistic* routing approach [24, 5, 57], that exploits two features of wireless communication: the uncertainty in packet transmission, and the broadcast nature of transmissions. Conceptually, the fundamental idea of opportunistic routing is to make routing decisions (selecting the next hop relay node) based on the actual realization of the preceding packet transmission that has to be observed *posteriori*, rather than on average statistics that can be computed *a priori*. Specifically, since the transmission is broadcast, multiple next hop nodes may have received it successfully. With such information, we can opportunistically select a relay from this set, instead of attempting retransmission to a pre-fixed next hop relay, who happened to have failed to receive the previous transmission. While the advantage (or lack thereof) of such an approach cannot be precisely determined without specifying the criteria used for relay selection, it is not hard to see intuitively why it should have an advantage in general. Consider the following simplistic, albeit revealing, analogy: if we toss multiple coins then the fastest way of getting a heads is to pick a coin that actually came up heads (or continue tossing till this happens), rather than sticking to a pre-selected coin and wait for it to come up heads.

In [24], a cost-optimal opportunistic stochastic routing algorithm in the form of an index policy was presented. It was shown that there exists a strict priority ordering among nodes, and that the optimal routing and transmission strategy is such that the packet transmission follows a path of increasing priorities. In [5] and [57] opportunistic routing algorithms were proposed and the relay selection metrics include its closeness to destination, EAX [57], etc.

In essence, the advantage of opportunistic routing is one of multi-receiver diversity gain. It is stochastic in nature since the routing decisions are event-based or equivalently, sample-path dependent. They depend on the actual realization of the system, and are made *after* observing the outcome of transmission. For this reason, we will also call this class of routing algorithms stochastic routing or *event-based routing*, a term first used in [24] in the context of ad hoc routing to the best of our knowledge, and refer to the more traditional routing algorithms discussed earlier as *non-event based routing* or *non-opportunistic routing*.

While opportunistic routing is intuitively appealing, its performance is not easy to quantify. For instance, even when the structure of the cost-optimal routing algorithm is precisely known [24] and may be implemented efficiently, its performance is nonetheless hard to model and quantify. One often has to resort to simulation for quantitative performance comparison. The objective of this chapter is to perform a quantitative comparison study on the routing delay of opportunistic and non-opportunistic routing algorithms. To make our analysis tractable and to obtain insight, we examine the scaling behavior of routing delays in the limiting regime as the network becomes large. This is a method widely used to study wireless networks due to their complexity, for example in the context of network throughput capacity [15], path length [41], etc. We follow the same approach in the present chapter.

Specifically, we consider a network of a fixed area with increasing node density. Each pair of nodes is associated with a transmission success probability, whose value is drawn from a given distribution. In subsequent sections we examine the routing delay induced by these two types of routing algorithms, and identify conditions under which it is finite (or infinite) as the number of nodes goes to infinity. Our main contributions are summarized as follows.

1. We show that when the transmission success probabilities are not bounded away from zero, non-opportunistic routing results in infinite routing delay while opportunistic routing has the same order as a straight-line error-free routing when no packet loss is assumed.

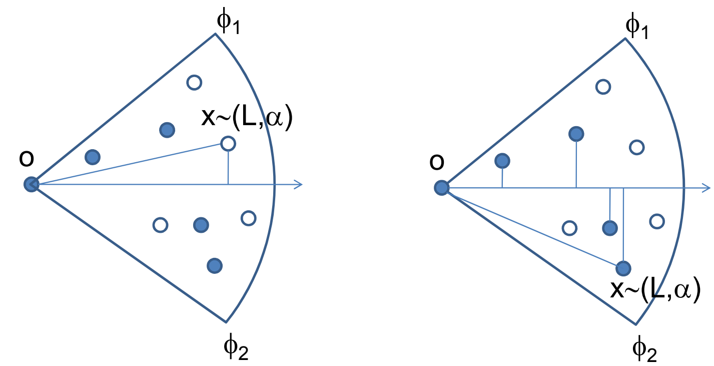
2. In the case where non-opportunistic routing has infinite delay, we show that combining it with multi-path routing is sufficient to turn the delay finite, albeit at the expense of increased transmission overhead.

The remainder of the chapter is organized as follows. Section 4.2 describes our network model and assumptions. Sections 4.3 and 4.4 analyze the routing delay for non-opportunistic and opportunistic routing methods, respectively. More practical scenarios are discussed in Section 4.5. Numeric simulation results are shown in Section 4.7 and Section 4.8 summarizes the chapter.

4.2 Network Model and Assumptions

Consider a wireless network where n nodes are randomly and uniformly deployed in a unit square. Each node is assumed to have a maximum transmission range $R(n)$, and any node within a circle of radius $R(n)$ of the transmitting node will receive the packet with a probability subsequently referred to as the (transmission) success probability. These nodes are called *neighbors* of the transmitting node. It is assumed that $R(n)$ is sufficiently large to ensure asymptotic network connectivity and straight-line routing [51]. In what follows we will assume $R(n) = K\sqrt{\frac{\log n}{n}}$ for some $K > 1$.

The success probability p, which is a pair-wise quantity, is used as a means to capture the fundamental uncertainty nature of wireless communication. This includes



(a) Non-opportunistic routing (b) Opportunistic routing

Figure 4.1: An illustration of choosing next hops by different routing methods.

failures due to fading, interference, as well as temporary unavailability caused by duty-cycling or self-recovery. Due to the dynamic nature of wireless channels, and due to the variety of causes for transmission failure, we will assume that the success probability is not a fixed quantity but rather given by a probability distribution f. More precisely, the transmission success probability p between any pair of two neighboring nodes is randomly drawn from a distribution f which may be pair-dependent. These probabilities are assumed to be independent across node pairs.

In subsequent sections we will first consider two classes of success probability distributions, those that have support over $[0, 1]$ and in particular support containing point 0, and those over $(\epsilon, 1]$ for some small $\epsilon > 0$, i.e., bounded away from zero. We will then consider a distance-based success distribution where f depends on the distance between a node pair.

For analysis and comparison purposes, we adopt the following representative models for the routing protocols under consideration. For the non-opportunistic routing scheme, we employ a random routing algorithm based on a geographic routing scenario considered in [44] [1], assuming imprecise node location information but precise destination information. Specifically, a node (denoted by 'o' in Figure 4.1(a)) selects

[1]Note that the term *random* here refers to the random selection of a relay node *a-priori*, thus this is still a non-opportunistic approach in the present context.

a forwarding node arbitrarily among neighbors in the sector within angles $[\phi_1, \phi_2]$ shown in Figure 4.1(a) and transmits to that node. Once a relay is selected, a transmission success probability p is randomly drawn from the distribution f associated with the node pair. We will assume that if the transmission fails, the source node will continue to retransmit to the same relay node with the same success probability p, till it succeeds [2].

For the opportunistic routing, we consider the following decision rule. As depicted in Figure 4.1(b), a relay node (denoted by 'o') transmits and some nodes in the sector (denoted by white dots) successfully receive it and the others (denoted by dark dots) do not. The next relay is selected among all successful receivers, that makes the largest hop progress toward the destination, approximated by the projection to a straight-line between the current transmitter and the destination (also called hop projection) [44]. For simplicity, we will ignore the possibility of multiple relay nodes that can occur in practice due to decentralized decision making. In our analysis we will assume that there is only one relay node. It is also naturally assumed that if none of the neighbors receives the message, retransmissions occur until at least one of them receives successfully.

Since we are interested in understanding how the routing delay scales as the total number of nodes (or the density) becomes large, we will assume that for any given node deployment a node has at least two neighbors to choose from. If this is not true then opportunistic routing reduces to the same as non-opportunistic routing for lack of a relay choice.

Within the context of lossy packet reception, the routing delay refers to the total amount of time it takes for a packet to travel from the source to the destination.

[2]The algorithm in [44] does not consider transmission failure; what's presented here is a lossy-transmission adaptation of that algorithm.

The corresponding terminologies and notations are as follows.

Definition 4.1. Hop count is the number of hops (relays) from the source to the destination.

Definition 4.2. Hop delay (1-hop delay) is the number of transmissions occurred until the next relay is different from the previous transmitter.

Definition 4.3. Conditional hop progress is the progress toward the destination given that hop progress is nonzero.

Definition 4.4. Routing delay is the sum of hop delays from the source to the destination.

We denote by $H_k^{(n)}$ the conditional hop progress taken at the kth hop. The hop count $h(n)$ is calculated by

$$h(n) = \sup\{j : \sum_{k=1}^{j} H_k^{(n)} < d - \gamma(n)\}, \tag{4.1}$$

where d is the distance between the source and the destination and $\gamma(n)$ is the size of the ball around the destination. Within this $\gamma(n)$ ball, we assume that straight-line routing is employed with enough routing information. So, its size is assumed to be very small and order-wise negligible. Hop delay taken at the kth hop is denoted by $N_k^{(n)}$. Then, routing delay $\tau(n)$ is

$$\tau(n) = \sum_{k=1}^{h(n)} N_k^{(n)}. \tag{4.2}$$

Throughout this paper, we use the following notations to express the order of asymptotic routing delay. For two functions $x(n)$ and $y(n)$ defined on some subset of the real line, $x(n) = O(y(n))$ implies that there exist numbers n_0 and M such that $|x(n)| \leq M \cdot |y(n)|$. Then, $x(n) = \Theta(y(n))$ implies that $x(n) = O(y(n))$ and $y(n) = O(x(n))$.

As we derive asymptotic routing delay of opportunistic routing, we make use of some result from [44]. To summarize, one of their results is to show the order of hop count under geographic routing with imprecise node location information but precise destination location information. In this case, geographic routing acts as random routing, which is non-opportunistic routing in our classification, that chooses a random node in a sector within angles $[\phi_1, \phi_2]$ such that for a uniformly chosen point (L, α) in the sector, $E[L \cos \alpha] > 0$, where $L \in (0, R(n)], \alpha \in [\phi_1, \phi_2]$. In other words, the angles of the sector ensures the positive expected progress toward the destination. Then, it is shown that the routing delays are on the order of $\Theta(\frac{1}{R(n)})$ with the scaling constant inversely proportional to the expected progress toward the destination. From this result, we can infer that if hop delay is finite and fixed, hop count will be on the order of $\Theta(\frac{1}{R(n)})$. In turn, routing delay is going to be on the same order. However, if it is infinite, routing delay is also unbounded.

4.3 Non-Opportunistic Routing

In this section, the delay performance of non-opportunistic routing is analyzed. Let a node in the sector contained between the angles $[\phi_1, \phi_2]$ be expressed by the polar coordinate (L, α). For notational simplicity, we use the nodes' normalized coordinates $(\tilde{L}, \alpha)$ when $R(n)$ is assumed to be unity. We will also assume $E[\tilde{L} \cos \alpha] = \beta \in (0, 1]$. This assumption (which guarantees the expectation of hop progress to be positive) is necessary to show the hop count to be $\Theta(\frac{1}{R(n)})$ as in [44].

4.3.1 Transmission success probability density with support $[0, 1]$

We begin by considering the success probability p to be randomly distributed on $[0, 1]$. In addition, we require that $f(p) > 0$ for $p = 0$, i.e., there is non-zero density of generating a zero probability of success. This may loosely model a deep fading scenario where a node has extremely poor reception for possibly a long time relative to data transmission and the associated protocol timeout values.

The following lemma establishes that the hop delay of non-opportunistic routing under this type of success probability is unbounded. Consider a transmission at the kth hop. The number of neighbors in the sector of the kth hop is denoted by $M_k(n)$, also simplified as M_k. $E[N_k^{(n)}|M_k = m]$ represents the expected hop delay given m neighbors in the sector. $E[H_k^{(n)}|M_k = m]$ is the expected conditional hop progress given there are m neighbors in the sector.

Lemma 4.1. *When f is not bounded away from zero, the expected hop delay of non-opportunistic routing is infinite. That is,*

$$E[N_k^{(n)}] = \infty.$$

Proof. Given $m > 1$, $E[N_k^{(n)}|M_k = m]$ is calculated as follows.

$$E[N_k^{(n)}|M_k = m] = \sum_{j=1}^{\infty} j \cdot Pr(N_k^{(n)} = j|M_k = m). \tag{4.3}$$

The probability that the number of transmissions at the kth hop is exactly j is

$$\begin{aligned} Pr(N_k^{(n)} = j|M_k = m) &= \int_0^1 (1-p)^{j-1} p f(p) dp \\ &= \int_0^1 \tilde{p}^{j-1} f(1-\tilde{p}) d\tilde{p} - \int_0^1 \tilde{p}^{j} f(1-\tilde{p}) d\tilde{p}, \end{aligned} \tag{4.4}$$

where the second equality is obtained by a change of variables $\tilde{p} = 1 - p$. The event $\{N_k^{(n)} = j\}$ is independent of the event $\{M_k = m\}$ provided at least one node in the sector. Thus $E[N_k^{(n)}|M_k = m] = E[N_k^{(n)}]$. By applying Eqn. (4.4), Eqn. (4.3) becomes

$$E[N_k^{(n)}] = E[N_k^{(n)}|M_k = m]$$
$$= \sum_{j=1}^{\infty} j \cdot \left(\int_0^1 \tilde{p}^{j-1} f(1-\tilde{p}) d\tilde{p} - \int_0^1 \tilde{p}^j f(1-\tilde{p}) d\tilde{p} \right)$$
$$= \sum_{j=1}^{\infty} \int_0^1 \tilde{p}^{j-1} f(1-\tilde{p}) d\tilde{p}, \tag{4.5}$$

where the second equality is obtained by a change of variables. Now consider the vicinity of $\tilde{p} = 1$ in Eqn. (4.5). Suppose $\tilde{p} = \tilde{p}^*$ achieves the minimum of probability density $f(1-\tilde{p})$ over $[1-\delta, 1]$ for some small $\delta > 0$. Eqn. (4.5) is thus lower-bounded by

$$\sum_{j=1}^{\infty} \int_{1-\delta}^1 \tilde{p}^{j-1} f(1-\tilde{p}) d\tilde{p} \geq f(1-\tilde{p}^*) \sum_{j=1}^{\infty} \int_{1-\delta}^1 \tilde{p}^{j-1} d\tilde{p}$$
$$= f(1-\tilde{p}^*) \sum_{j=1}^{\infty} \left(\frac{1}{j} - \frac{(1-\delta)^j}{j} \right)$$
$$\geq f(1-\tilde{p}^*) \left(\sum_{j=1}^{\infty} \frac{1}{j} - \sum_{j=1}^{\infty} (1-\delta)^j \right) = \infty,$$

where the latter summation gives $\frac{1-\delta}{\delta}$ whereas the former summation yields infinity. The expected hop delay $E[N_k^{(n)}]$ is thus infinite. □

We see that when the transmission success probability is not bounded away from zero, the expected hop delay diverges regardless of n.

4.3.2 Transmission success probability density with support $[\epsilon, 1]$

We now consider the second case where p is a bounded random variable in $[\epsilon, 1]$ for some small $\epsilon > 0$.

Lemma 4.2. *When f is bounded away from 0 and defined over $[\epsilon, 1]$ forr $\epsilon > 0$, the expected hop delay of non-opportunistic routing is finite and independent of n, i.e.,* $E[N_k^{(n)}] < \infty$.

Proof. The expected hop delay at the kth hop is calculated using Eqn. (4.5). For a general distribution $f(p)$ over $[\epsilon, 1]$ where $\epsilon > 0$, hop delay is

$$\begin{aligned}
E[N_k^{(n)}|M_k = m] &= \sum_{j=1}^{\infty} \int_0^{1-\epsilon} \tilde{p}^{j-1} f(1-\tilde{p}) d\tilde{p} \\
&= \lim_{J\to\infty} \sum_{j=1}^{J} \int_0^{1-\epsilon} \tilde{p}^{j-1} f(1-\tilde{p}) d\tilde{p} \\
&= \lim_{J\to\infty} \int_0^{1-\epsilon} \frac{1-\tilde{p}^J}{1-\tilde{p}} f(1-\tilde{p}) d\tilde{p} \\
&= \lim_{J\to\infty} \int_\epsilon^{1} \frac{1-(1-p)^J}{p} f(p) dp \\
&= \lim_{J\to\infty} E\left[\frac{1-(1-p)^J}{p}\right] \\
&= E\left[\lim_{J\to\infty} \frac{1-(1-p)^J}{p}\right] = E\left[\frac{1}{p}\right],
\end{aligned}$$

where the second to last equality holds by the monotone convergence theorem because the function $\frac{1-(1-p)^J}{p}$ monotonically increases and converges to $1/p$ as J increases. Note that $E[N_k^{(n)}|M_k = m]$ is the same for all $m \geq 1$. Therefore, we have $E[N_k^{(n)}] = E[N_k^{(n)}|M_k = m] < \infty$. □

Theorem 4.1. *Let f be defined over $[\epsilon, 1]$ for $\epsilon > 0$. Then the routing delay $\tau(n)$ of non-opportunistic routing is on the order of $\Theta(\frac{1}{R(n)})$ as n increases.*

Proof. To derive $\tau(n)$ as defined by Eqn. (4.2), we first consider hop delay $N_k^{(n)}$. By Lemma 4.2, we have $E[N_k^{(n)}] < \infty$. On the other hand, the expected hop progress is obtained as follows. Similarly to hop delay, hop progress of non-opportunistic routing is also independent from the number of neighbors in the sector as long as there is at least one neighbor. $E[H_k^{(n)}|M_k = m]$ is equal to $E[H_k^{(n)}]$. From the results in [44], we have the following: given $E[\tilde{L}\cos\alpha] = \beta$ for some $\beta \in (0, 1)$, hop count $h(n)$ is on the order of $\Theta(\frac{1}{R(n)})$. Thus, we have $\frac{c_1}{R(n)} \leq h(n) \leq \frac{c_2}{R(n)}$ for positive c_1, c_2

such that $c_1 < \frac{1}{\beta} < c_2$. Assume $M_k \geq 1$. Then,

$$\sum_{k=1}^{\frac{c_1}{R(n)}} N_k^{(n)} \leq \tau(n) \leq \sum_{k=1}^{\frac{c_2}{R(n)}} N_k^{(n)},$$

where $N_k^{(n)}$'s are i.i.d. positive random variables with a finite mean $E[N_k^{(n)}]$. By the strong law of large numbers, the left and right sums converge to $\frac{c_1 E[N_k^{(n)}]}{R(n)}$ and $\frac{c_2 E[N_k^{(n)}]}{R(n)}$ almost surely as $n \to \infty$, respectively. Thus, for some positive constants $c_1^* \leq c_1 E[N_k^{(n)}]$ and $c_2^* \geq c_2 E[N_k^{(n)}]$,

$$\frac{c_1^*}{R(n)} \leq \tau(n) \leq \frac{c_2^*}{R(n)}.$$

□

To conclude, we see that when success probability is not bounded away from zero, non-opportunistic routing has an infinite expect delay per hop, and a constant hop delay when the success probability is bounded away from zero. In the latter case its routing delay is on the order of $\Theta(\frac{1}{R(n)})$; one naturally expects the routing delay in the former case to grow faster.

4.4 Opportunistic Routing

In this section, the delay performance of opportunistic routing is analyzed for any success probability distribution f over $[0, 1]$. Note that the two classes defined in the previous section are both special cases of this. We make the same assumption as in the previous section: $E[\tilde{L} \cos \alpha] = \beta \in (0, 1]$.

Lemma 4.3. *For any distribution f, the expected hop delay of opportunistic routing is finite: $E[N_k^{(n)}] < \infty$.*

Proof. Denote by m the number of neighbors in the section. Denote by p_i the success probability of node i. Given $m \geq 1$, $E[N_k^{(n)} | M_k = m]$ is calculated by Eqn. (4.3).

For opportunistic routing, we have:

$$Pr(N_k^{(n)} = j|M_k = m) = \int_0^1 \cdots \int_0^1 \left(1 - \prod_{l=1}^{m}(1-p_l)\right)\left(\prod_{l=1}^{m}(1-p_l)^{j-1} f(dp_l)\right).$$

By a change of variables $\tilde{p}_i = 1 - p_i, \forall i$, and noting that $\tilde{p}_i$'s are independent, Eqn. (4.3) can be written as

$$\begin{aligned} E[N_k^{(n)}|M_k = m] &= \sum_{j=1}^{\infty} j \left(\int_0^1 \tilde{p}^{j-1} f(1-\tilde{p})\, d\tilde{p}\right)^m - \sum_{j=1}^{\infty} j \left(\int_0^1 \tilde{p}^{j} f(1-\tilde{p})\, d\tilde{p}\right)^m \\ &= \sum_{j=1}^{\infty} \left(\int_0^1 \tilde{p}^{j-1} f(1-\tilde{p})\, d\tilde{p}\right)^m. \end{aligned} \tag{4.6}$$

We next show that Eqn. (4.6) is finite given $m \geq 2$.

Consider the vicinity of $\tilde{p} = 1$. Suppose $\tilde{p} = \tilde{p}^*$ achieves the maximum of $f(1-\tilde{p})$ over $[0, \delta]$ for some $\delta > 0$. Then, Eqn. (4.6) is upper-bounded by

$$\begin{aligned} \sum_{j=1}^{\infty} (\frac{1}{\delta} \int_{1-\delta}^1 \tilde{p}^{j-1} f(1-\tilde{p}^*) d\tilde{p})^m &= \sum_{j=1}^{\infty} \frac{f(1-\tilde{p}^*)^m}{\delta^m} \left(\int_{1-\delta}^1 \tilde{p}^{j-1} d\tilde{p}\right)^m \\ &= \frac{f(1-\tilde{p}^*)^m}{\delta^m} \sum_{j=1}^{\infty} \left(\frac{1}{j} - \frac{(1-\delta)^j}{j}\right)^m, \end{aligned}$$

which is finite because the geometric sum is finite for any $m \geq 2$. Then, Eqn. (4.6) is finite for any $m \geq 2$. Therefore, the unconditional expectation $E[N_k^{(n)}]$ is also finite for any distribution of M_k. □

Lemma 4.4. *For any distribution f, given $E[\tilde{L} \cos \alpha] = \beta$, the expected conditional hop progress given m neighbors in the sector, $E[\tilde{H}_k^{(n)}|M_k = m]$, is nondecreasing and bounded in $[\beta, 1]$, for $m > 1$.*

Proof. Consider that there are $m > 1$ neighbors in the sector. The conditional hop progress at the kth hop, $H_k^{(n)}$, given $m > 1$ is well-approximated by the maximum of hop projections among successful neighbors. Suppose that the number of successful receivers is exactly m_s after transmission. Let $\tilde{A}_i = \tilde{L}_i \cos\alpha$. Then, $\tilde{H}_k^{(n)}$ given m_s successful neighbors in the sector is obtained by $\max\{\tilde{A}_1, \tilde{A}_2, \cdots, \tilde{A}_{m_s}\}$. Note that $\tilde{A}_i$'s are i.i.d. random variables with mean β by assumption. We denote by $F_{\tilde{A}}(\cdot)$ the probability distribution function of $\tilde{A}_i, \forall i$. Then, the probability distribution function of $\tilde{H}_k^{(n)}$ given m_s successful neighbors is

$$F_{\tilde{H}_k}^{m_s}(h) = (F_{\tilde{A}}(h))^{m_s}.$$

We will obtain $E[\tilde{H}_k^{(n)}|M_k = m]$ first. We have

$$\begin{aligned} &Pr(m_s = i|m_s \geq 1, M_k = m) \\ &\qquad = \frac{\int_0^1 \cdots \int_0^1 Pr(m_s = i|p_1 \cdots p_m) Pr(p_1 \cdots p_m) dp_1 \cdots dp_m}{\int_0^1 \cdots \int_0^1 Pr(m_s \geq 1|p_1 \cdots p_m) Pr(p_1 \cdots p_m) dp_1 \cdots dp_m} \\ &\qquad = \frac{\int_0^1 \cdots \int_0^1 \binom{m}{i} \prod_{l=1}^{i} p_l \prod_{l=i+1}^{m} (1-p_l) f(dp_1) \cdots f(dp_m)}{\int_0^1 \cdots \int_0^1 (1 - \prod_{l=1}^{m}(1-p_l)) f(dp_1) \cdots f(dp_m)}. \end{aligned}$$

Using this probability, we obtain:

$$\begin{aligned} Pr(\tilde{H}_k^{(n)} \leq h|M_k = m) &= \sum_{i=1}^{m} F_{\tilde{H}_k}^{m_s}(h) Pr(m_s = i|m_s \geq 1, M_k = m) \\ &= \sum_{i=1}^{m} (F_{\tilde{A}}(h))^i \frac{\binom{m}{i} E[p]^i (1-E[p])^{m-i}}{1-(1-E[p])^m} \\ &= \frac{(F_{\tilde{A}}(h)E[p] + 1 - E[p])^m - (1-E[p])^m}{1-(1-E[p])^m}. \end{aligned}$$

Denote $E[p]$ by p^*. The expected value of $\tilde{H}_k^{(n)}$ given m neighbors in the sector is

$$E[\tilde{H}_k^{(n)}|M_k = m] = \int_0^1 \frac{1-(F_{\tilde{A}}(h)p^* + 1 - p^*)^m}{1-(1-p^*)^m}\, dh. \tag{4.7}$$

When $m = 1$, Eqn. (4.7) is β because $\int_0^1 (1 - F_{\tilde{A}})(h) dh = \beta$. As m goes to infinity, it approaches 1.

We next prove that $E[\tilde{H}_k^{(n)}|M_k = m]$ is nondecreasing in m. By applying Eqn. (4.7), we have

$$
\begin{aligned}
&E[\tilde{H}_k^{(n)}|M_k = m] - E[\tilde{H}_k^{(n)}|M_k = m-1] \\
&\qquad = \frac{\int_0^1 p^*(F_{\tilde{A}}(h)p^* + 1 - p^*)^{m-1}(1 - F_{\tilde{A}}(h)(1-(1-p^*)^{m-1}))dh}{(1-(1-p^*)^m)(1-(1-p^*)^{m-1})} \\
&\qquad\quad - \frac{p^*(1-p^*)^{m-1}}{(1-(1-p^*)^m)(1-(1-p^*)^{m-1})} \\
&\qquad = \frac{1-\tilde{p}}{(1-(1-\tilde{p})^m)(1-(1-\tilde{p})^{m-1})} \cdot \\
&\qquad\quad \left(\int_0^1 (\tilde{p} + F_{\tilde{A}}(h)(1-\tilde{p}))^{m-1}(1 + F_{\tilde{A}}(h)(1-\tilde{p}^{m-1})) - \tilde{p}^{m-1} dh\right), \qquad (4.8)
\end{aligned}
$$

where the second equality is obtained by a change of variables $\tilde{p} = 1 - p^*$. In Eqn. (4.8), the term within the integral is nonnegative because $(\tilde{p} + F_{\tilde{A}}(h)(1 - \tilde{p}))^{m-1}(1 + F_{\tilde{A}}(h)(1-\tilde{p}^{m-1}))$ is always greater than or equal to $\tilde{p}^{m-1}$ for any value of $F_{\tilde{A}}(h)$ and p. Therefore, Eqn. (4.8) is nonnegative. Hence, Eqn. (4.7) is nondecreasing and bounded in $[\beta, 1]$. □

The proofs for the next three lemmas can be found in the appendix.

Lemma 4.5. *Consider bounded i.i.d. random variables* $X_i^{(n)}, 1 \le i \le n$ *and* $E[X_i^{(n)}|M(n) = m]$ *is nondecreasing in* m*, where* $M(n)$ *is a random variable such that* $M(n) = \Theta(\log n)$*. Suppose* $\lim_{m\to\infty} E[X_i^{(n)}|M(n) = m] = \beta^*$*. Then, we have*

$$\lim_{n\to\infty} E[X_i^{(n)}] = \beta^*.$$

The next lemma is necessary to prove Theorem 4.7. It is comparable to the limit theorem for triangular arrays presented in [44]; the difference is that in our case the expectation of conditional hop progress is not a fixed value for all n as in theirs but a nondecreasing sequence of n.

Lemma 4.6. *For any fixed* $K > 1$*, let* $R(n) = K\sqrt{(\log n/n)}$*. Consider bounded i.i.d. random variables* $X_i^{(n)}, 1 \le i \le n$ *where* $E[X_i^{(n)}] = \beta^{(n)}$ *and* $\lim_{n\to\infty} \beta^{(n)} = \beta^*$.

Then,

$$\lim_{n\to\infty} R(n) \sum_{i=1}^{\frac{1}{R(n)}} X_i^{(n)} = \beta^* \text{ (almost surely).}$$

The next lemma shows that the hop count of opportunistic routing is on the same order as straight-line routing. The proof of this lemma is mostly identical to the proof of Theorem 3.1 in [44] where $E[\tilde{L}^{(n)} \cos(\alpha^{(n)})] = \beta$ which is a fixed value between 0 and 1. In our case, β is not a fixed value but a convergent sequence. By using Lemma 4.6, one can easily derive the results.

Lemma 4.7. *For any fixed $K > 1$, let $R(n) = K\sqrt{(\log n/n)}$. Let $(L^{(n)}, \alpha^{(n)})$ be the polar coordinates of a randomly chosen point from a sector and radius $R(n)$. And, let $(\tilde{L}^{(n)}, \alpha^{(n)})$ be corresponding coordinates when $R(n)$ is scaled to 1. Let $E[\tilde{L}^{(n)} \cos(\alpha^{(n)})] = \beta^{(n)}$ and $\beta^* = \lim_{n\to\infty} \beta^{(n)}$. Then, $\forall$ positive k_1 and k_2 such that $k_1 < \frac{1}{\beta^*} < k_2$, $h(n)$ satisfies*

$$\frac{k_1}{K}\sqrt{\frac{n}{\log n}} \le h(n) \le \frac{k_2}{K}\sqrt{\frac{n}{\log n}} \text{ (asymptotically a.s.).}$$

Theorem 4.2. *For any success distribution f, the routing delay $\tau(n)$ of opportunistic routing is on the order of $\Theta(\frac{1}{R(n)})$ as n increases.*

Proof. From Lemma 4.4, we know $E[\tilde{H}_k^{(n)} | M_k = m]$ is bounded in $[\beta, 1]$ for all $m > 1$ and nondecreasing. Let us denote by $\beta^{(n)}$ the unconditional expectation $E[\tilde{H}_k^{(n)}]$, which is also within $[\beta, 1]$. Then, $\lim_{n\to\infty} \beta^{(n)} = 1$ by Lemma 4.5. Finally, according to Lemma 4.7, we obtain hop count $h(n)$ is on the order of $\Theta(\frac{1}{R(n)})$. In other words, there are positive k_1 and k_2 such that for $k_1 < 1 < k_2$, we have $\frac{k_1}{R(n)} \le h(n) \le \frac{k_2}{R(n)}$. Of course, its scaling constants k_1 and k_2 are smaller than c_1 and c_2 for non-opportunistic routing.

From Eqn. (4.2), Routing delay $\tau(n)$ is expressed by

$$\sum_{k=1}^{\frac{k_1}{R(n)}} N_k^{(n)} \leq \tau(n) \leq \sum_{k=1}^{\frac{k_2}{R(n)}} N_k^{(n)},$$

where $N_k^{(n)}$'s are i.i.d. positive random variables with a finite mean $E[N_k^{(n)}]$. By the strong law of large numbers, the left and right sums converge to $\frac{k_1 E[N_k^{(n)}]}{R(n)}$ and $\frac{k_2 E[N_k^{(n)}]}{R(n)}$ almost surely as $n \to \infty$, respectively. Thus, for some positive constants $k_1^* \leq k_1 E[N_k^{(n)}]$ and $k_2^* \geq k_2 E[N_k^{(n)}]$,

$$\frac{k_1^*}{R(n)} \leq \tau(n) \leq \frac{k_2^*}{R(n)}.$$

$\square$

We conclude that the delay of opportunistic routing scales on the order of $\Theta(\frac{1}{R(n)})$ under any success probability distribution f over $[0,1]$.

4.5 Distance-based Success Probability Model

The results from the previous two sections indicate that in terms of the qualitative scaling property of routing delay, the difference between these two routing methods only exists when the event that certain transmission can fail deterministically ($p = 0$) can occur with non-zero probability. Intuitively, what this suggests is that the fundamental difference between the two is that opportunistic routing is more robust since it will not be stuck with a fixed, potentially very bad route, while non-opportunistic routing can. In reality, a link is not going to be extremely poor ($p = 0$) for infinitely long, which is the model used in the analysis. So even a non-opportunistic routing algorithm eventually will recover either because the link quality becomes better, or through other built-in recovery mechanisms in the routing protocol, e.g., via timeouts. What this result indicates is thus that non-opportunistic routing can experience very long delays in the presence of events like deep fading, long sleep periods, etc.

On the other hand, the quantitative advantage of opportunistic routing when both routing algorithms have finite hop delay is not shown through the scaling result; they have the same order. They only differ through the constant factor. While numerical results comparing the two are provided in a later section, below we use a case study, where the success probabilities are functions of the pair-wise distance, to illustrate the difference in routing delay under these two different routing methods.

Suppose $f(p|\tilde{L}) = \delta(p - 1 + \tilde{L})$ for $0 \le \tilde{L} \le \tilde{L}_{max} < 1$ and 0 otherwise. That is, transmission success probability is equal to $1 - \tilde{L}$ for nodes located at a distance less than or equal to $\tilde{L}_{max}$ while nodes that are located farther than $\tilde{L}_{max}$ have no success probability (these will not be considered as relays). By doing so, p is lower-bounded by a nonzero value which results in finite hop delays for both routing schemes.

4.5.1 Non-Opportunistic Routing

Consider the kth hop. A neighbor is randomly chosen among m neighbors in the sector. The expected hop delay of non-opportunistic routing given $M_k = m$ is

$$\begin{aligned} E[N_k^{(n)}|M_k = m] &= \sum_{j=1}^{\infty} j \cdot \int_0^{\tilde{L}_{max}} \frac{2l}{\tilde{L}_{max}} \int_0^1 (1-p)^{j-1} p\, \delta(p - 1 + l) dp\, dl \\ &= \sum_{j=1}^{\infty} \frac{2(\tilde{L}_{max})^j}{j+1}. \end{aligned} \tag{4.9}$$

where $\sum_{j=1}^{\infty} \frac{(\tilde{L}_{max})^j}{j}$ converges since $\tilde{L}_{max} < 1$ by the Ratio test.

4.5.2 Opportunistic Routing

Consider there are m neighbors at the kth hop. Denote by $(\tilde{L}_i, \alpha_i)$ the location of node i. The expected hop delay of opportunistic routing given m nodes in the sector is

$$
\begin{aligned}
&E[N_k^{(n)}|M_k = m] \\
&\quad = \sum_{j=1}^{\infty} j \int_0^1 \cdots \int_0^1 Pr(N_k^{(n)} = j|\tilde{L}_1 = l_1, \cdots, \tilde{L}_m = l_m) \\
&\qquad\qquad \cdot Pr(\tilde{L}_1 = l_1, \cdots, \tilde{L}_m = l_m) dl_1, \cdots, dl_m \\
&\quad = \sum_{j=1}^{\infty} j \left(\frac{2}{\tilde{L}_{max}}\right)^m \int_0^{\tilde{L}_{max}} \cdots \int_0^{\tilde{L}_{max}} l_1 \cdots l_m \\
&\qquad\qquad \cdot Pr(N_k = j|\tilde{L}_1 = l_1, \cdots, \tilde{L}_m = l_m)\, dl_1 \cdots dl_m,
\end{aligned}
$$

In the above equation, the probability that $N_k^{(n)} = j$ given nodes' locations is

$$
\begin{aligned}
&Pr(N_k^{(n)} = j|\tilde{L}_1 = l_1, \cdots, \tilde{L}_m = l_m) \\
&\quad = \int_0^1 \cdots \int_0^1 \left(\prod_{i=1}^{m} (1-p_i)^{j-1}\right) \left(1 - \prod_{i=1}^{m} (1-p_i)\right) f(p_1|l_1) \cdots f(p_m|l_m) dp_1 \cdots dp_m \\
&\quad = l_1^{j-1} \cdots l_m^{j-1} (1 - l_1 \cdots l_m).
\end{aligned}
$$

By applying this, we have

$$
\begin{aligned}
&E[N_k^{(n)}|M_k = m] \\
&\quad = \sum_{j=1}^{\infty} j \left(\frac{2}{\tilde{L}_{max}}\right)^m \int_0^{\tilde{L}_{max}} \cdots \int_0^{\tilde{L}_{max}} l_1^j \cdots l_m^j \cdot (1 - l_1 \cdots l_m) dl_1 \cdots dl_m \\
&\quad = \sum_{j=1}^{\infty} j \left(\frac{2}{\tilde{L}_{max}}\right)^m \left((\int_0^{\tilde{L}_{max}} l_1^j dl_1)^m - (\int_0^{\tilde{L}_{max}} l_1^{j+1} dl_1)^m \right) \\
&\quad = \sum_{j=1}^{\infty} \left(\frac{2(\tilde{L}_{max})^j}{j+1}\right)^m . \qquad (4.10)
\end{aligned}
$$

Eqn. (4.10) shows that the larger the number of neighbors in the sector, the smaller the hop delay of opportunistic routing. Thus, as the density n increases, it converges to 1. Compared to hop delay of non-opportunistic routing given in Eqn. (4.9), which is a constant, hop delay of opportunistic routing becomes significantly smaller as more neighbors are available. This of course is not at all surprising since the gain of opportunistic routing originates from multi-receiver diversity as we noted earlier.

If the success probability declines at the higher rate as distance increases, e.g., $p_i = 1 - \tilde{L}^d$, $d \geq 2$, the expected progress per hop gets smaller for both non-opportunistic routing and opportunistic routing while the number of retransmissions per hop gets larger. But, it still results in the same performance order-wise.

4.6 The Extension to Multi-Path Routing

In this section we show that the lack of robustness in non-opportunistic routing can be sufficiently compensated (i.e., can turn the unbounded delay to bounded) by using multiple paths (each is non-opportunistic), at the expense of increased overhead.

We consider a routing scheme that creates multiple paths simultaneously. The routing delay is taken to be the minimum of the delays along different paths. This scheme works as follows. At each hop, $B \geq 1$ copies are generated and forwarded to B randomly chosen neighbors. If at least one of B transmissions are successful, the transmission stops. This can be viewed as the case where B nodes are in the sector and if there is at least one successful receiver, all successful receivers will act as relays in the next hop. Thus, this scheme works better than opportunistic routing with exactly B neighbors in the sector at each hop. Therefore, it has finite hop delay and $\Theta(\frac{1}{R(n)})$ routing delay. Denote by $\mathcal{P}$ the set of paths and by P_i the ith path. Then, routing delay is defined by

$$\tau(n) = \min_{1 \leq i \leq |\mathcal{P}|} |P_i|. \tag{4.11}$$

where $|P_i|$ is defined as path delay along the path P_i. The case of $B = 1$ is trivial because it is exactly the same as the non-opportunistic routing we have considered.

Below we assume that $B \geq 2$. When B becomes large, delay is the same order-wise but the scaling constant gets smaller with higher overhead (i.e., the total number of transmissions in the network). Thus, the case of $B = 2$ gives us an upper bound on hop delay and a lower bound on overhead of this scheme.

Consider that node i receives a packet successfully with probability p_i given by a random distribution $f(p_i)$. For simplicity of analysis, we assume that p_i is uniformly distributed between 0 and 1 for all i. Consider a sender at the kth hop. Multi-path routing randomly picks two nodes among the neighbors in the sector. Then, transmission is repeated until at least one of those two is successful. Denote by P_f the probability of unsuccessful receptions at both neighbors. Then, the expected number of retransmissions at the kth hop is calculated as follows.

$$\begin{aligned}
E[N_k^{(n)}] &= \sum_{j=1}^{\infty} j \cdot P_f^{j-1}(1 - P_f) \\
&= \sum_{j=1}^{\infty} j \cdot (\int_0^1 (1-p_1)f(p_1)dp_1)^{2(j-1)} \cdot (1 - (\int_0^1 (1-p_1)f(p_1)dp_1)^2) \\
&= \sum_{j=1}^{\infty} j \cdot (\frac{1}{4})^{j-1}\frac{3}{4} \\
&= \frac{4}{3}.
\end{aligned}$$

Next, the expected number of branches at the kth hop, denoted by $E[\hat{B}_k^{(n)}]$, represents the number of copies $\hat{B}_k^{(n)}$ generated at the kth hop on average. Retransmissions at each hop ends with at least one successful reception. Thus, at the last trial of retransmissions at each hop, $\hat{B}_k^{(n)}$ takes a value of one or two. Denote by $P_{s,1}$ the probability of exactly one successful reception at one of two neighbors and by $P_{s,2}$

the probability of two successful receptions at both neighbors. Then,

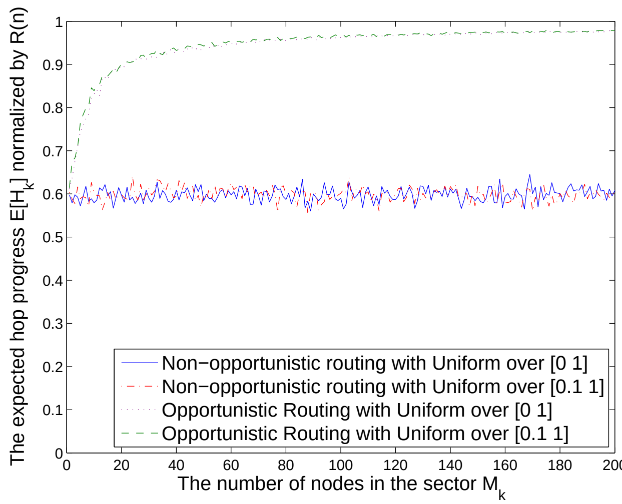
$$\begin{aligned}
E[\hat{B}_k^{(n)}] &= \sum_{j=1}^{2} j \cdot Pr(\hat{B}_k^{(n)} = j) \\
&= \sum_{j=1}^{2} j \cdot \frac{P_{s,j}}{1 - P_f} \\
&= \frac{\binom{2}{1} \int_0^1 pf(p)dp \int_0^1 (1-p)f(p)dp + 2(\int_0^1 pf(p)dp)^2}{1 - (\int_0^1 (1-p)f(p)dp)^2} \\
&= \frac{4}{3}.
\end{aligned}$$

The total number of transmissions is on the order of $\Theta(E[N_k^{(n)}] \cdot (E[\hat{B}_k^{(n)}])^{\frac{1}{R(n)}}) \simeq \Theta(\frac{4}{3}^{\frac{1}{R(n)}})$.

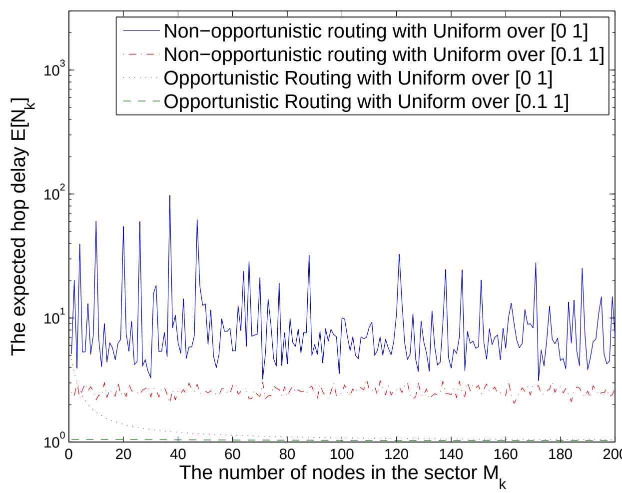
As we described earlier in this section, multi-path routing with B copies performs better than opportunistic routing where exactly B neighbors are in the sector at each hop. For comparison purpose, we evaluate hop delay of the latter case given in Eqn. (4.6) where f is a uniform distribution over $[0, 1]$ and $m = 2$. By calculation, we obtain $E[N_k^{(n)}] = \sum_{j=1}^{\infty} \frac{1}{j}$, which is Riemann zeta-function $\zeta(2) \approx 1.645$, whereas hop delay of multi-path routing is $\frac{4}{3}$. As we expected, multi-path routing with $B = 2$ achieves smaller hop delay than opportunistic routing with $B = 2$ neighbors at each hop.

4.7 Numerical Results

In previous section we analytically studied the order of routing delay of two examples of opportunistic and non-opportunistic routing methods. In this section, we perform MATLAB simulation to compare the performance of these routing schemes under a few transmission success/failure models. Emphasis here is on verifying the scaling property of routing delay as well as the scaling constants under different transmission success models.



(a) Hop progress

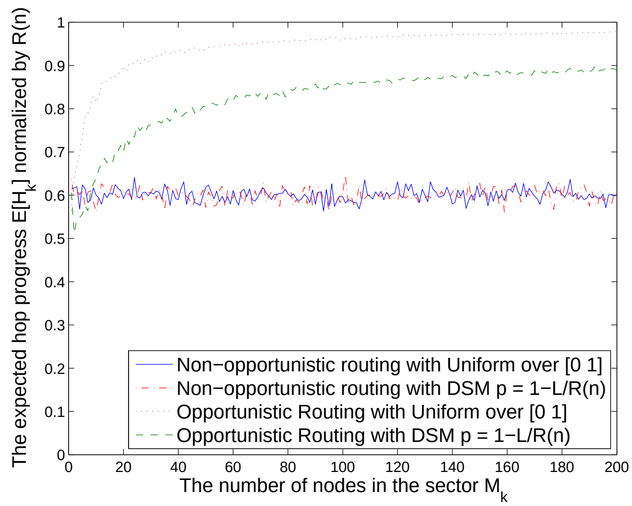


(b) Hop delay

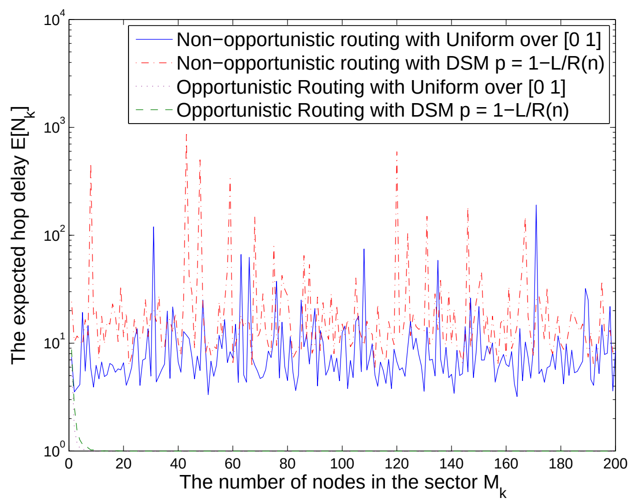
Figure 4.2: The 1-hop performance of non-opportunistic routing and opportunistic routing under uniform distributions.

First, we simulate hop progress and hop delay (the number of retransmissions). A relay node transmits a packet to neighbors in a sector with angles $[-\pi/4, \pi/4]$ and unit radius as shown in Figure 4.1. M_k nodes are generated within the sector and deployed uniformly. The next forwarding node is chosen according to both routing schemes (i.e., non-opportunistic routing and opportunistic routing), respectively. We consider two uniform distributions for the success probability: uniform over $[0, 1]$ and uniform over $[0.1, 1]$. Figure 4.2(a) shows that non-opportunistic routing has the same conditional hop progress regardless of M_k while conditional hop progress of opportunistic routing increases from the same value as the one of non-opportunistic routing when $M_k = 1$ close to 1 as M_k increases. However, there is no difference in hop delay between the distributions bounded by 0 and bounded by 0.1. In Figure 4.2(b), it is interesting to see that hop delay of non-opportunistic routing is extremely large and fluctuates when transmission success probability is not bounded away from 0, which gets much lower and consistent when it is bounded by 0.1. Like hop progress, hop delay is not affected by M_k. On the other hand, opportunistic routing maintains relatively very small hop delay for both distributions. Specially, hop delay reduces rapidly as M_k increases under the distribution with bounded by 0 whereas it is close to one with the distribution with bounded by 0.1.

Now, we examine hop progress and hop delay of both routing strategies under the distance-based success probability model (DSM) as well. As described in Section 4.5, we use $p = 1 - \tilde{L}$, where $\tilde{L} = L/R(n)$, equivalently speaking, transmission success probability linearly decreases with respect to to distance. Since the distance-based model generates more nodes with poor success probability than uniform distributed success probability model, hop delays of both routing schemes under the former case are greater than the ones under the latter as shown in Figure 4.3(b). Hop progress of

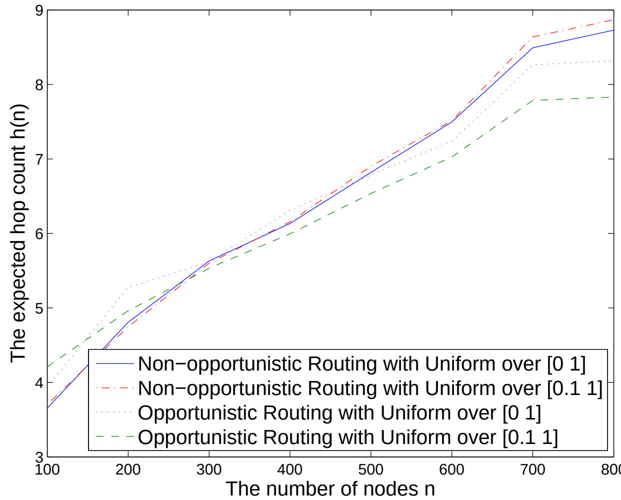


(a) Hop progress

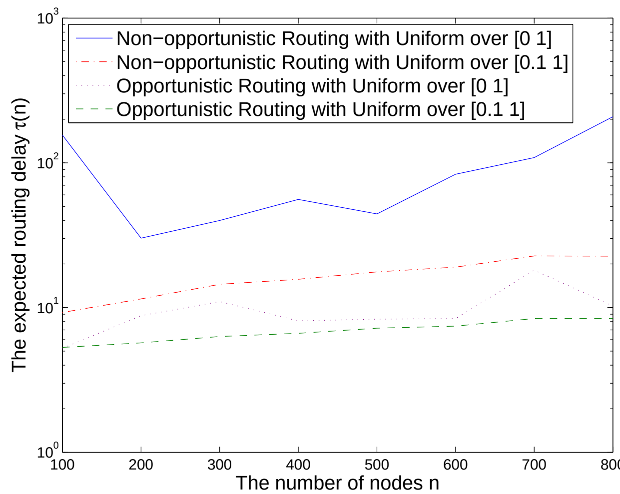


(b) Hop delay

Figure 4.3: Comparison of the 1-hop performance of non-opportunistic routing and opportunistic routing under uniform success transmission probability model vs. distance-based success transmission probability model (DSM).



(a) Hop count



(b) Routing delay

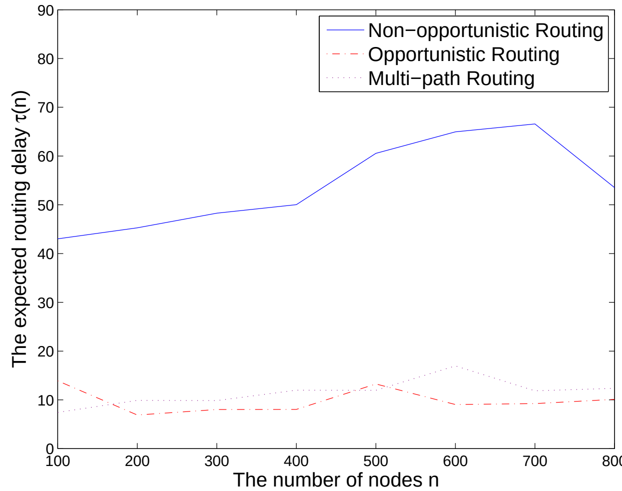
Figure 4.4: The performance of non-opportunistic routing and opportunistic routing under uniform distributions.

opportunistic routing under the distance-based success probability model converges to 1 at slower rate than the one under the uniform case. Overall, routing algorithms under the distance-based success probability model perform poorer than the uniform success probability model.

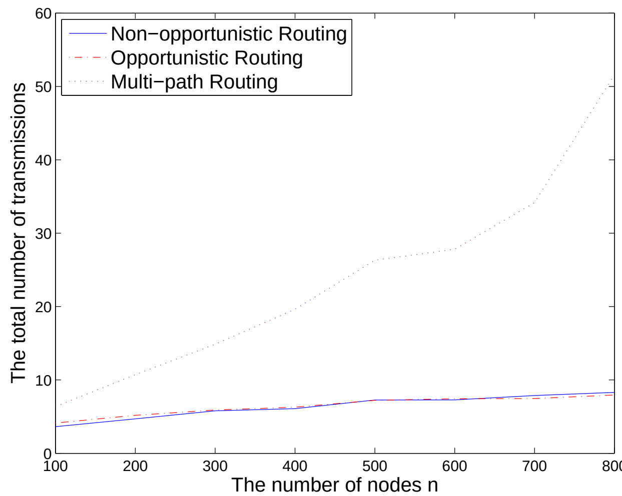
Next, we evaluate routing delay in the network where n nodes are randomly deployed in the unit square. At each run, Nodes are randomly relocated, and a source and a destination are randomly selected. Routing delay is averaged over 200 runs. Again, we compare opportunistic vs. non-opportunistic routing under the same distributions as above. Figure 4.4(a) shows hop counts which seem similar for all cases but those of opportunistic routing get slightly smaller as node density n gets higher. This is to be expected from the results in Figure 4.2(a) because hop progress of opportunistic routing increases as M_k increases. On the other hand, for small number of nodes n, opportunistic routing takes more number of hops to reach the destination because conditional hop progress is similar to that of non-opportunistic routing.

Overall, routing delays as shown in Figure 4.4(b) tend to increase as n increases. However, the ones under the distributions not bounded away from zero have much larger variation than the others. As clearly shown from the analysis, opportunistic routing is beneficial over non-opportunistic routing for both distributions.

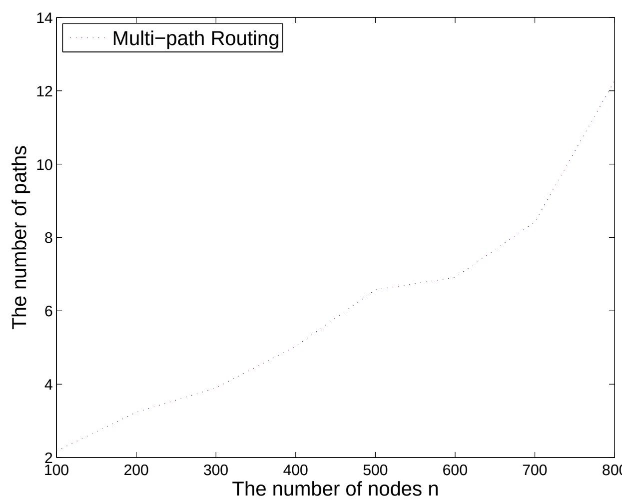
The final set of simulation shows the performance of multi-path routing together with non-opportunistic routing. In particular, we use uniform distribution not bounded away from zero to generate a transmission success probability. The comparison is made between multi-path routing, non-opportunistic routing, and opportunistic routing on the same network setting described right above. As shown in Figure 4.5(a), multi-path routing performs superior than non-opportunistic routing in terms of routing delay whereas the former requires increased overhead (i.e., the number of transmissions) as depicted in Figure 4.5(b). The expected number of paths created by multi-path routing linearly increase as n increases as shown in Figure 4.5(c). It is remarkable to see that opportunistic routing performs as good as multi-path routing with less overhead, which emphasize the merits of using such event-based routing in wireless networks.



(a) Routing delay



(b) The number of retransmissions



(c) The number of paths

Figure 4.5: The performance of multi-path routing under uniform distribution over $[0, 1]$.

4.8 Chapter Summary

In this chapter we have studied the asymptotic routing delay of opportunistic routing as well as non-opportunistic routing for wireless ad hoc and sensor networks where there are uncertainties involved in packet transmission due to fading, node failures, power saving, etc. These characteristics were captured by a transmission success probability randomly drawn from a given distribution. We have shown that when the transmission success probabilities are not bounded away from zero, non-opportunistic routing results in extremely large delay. It is because there is a possibility that a node with a very low success probability is chosen as a relay. By contrast, opportunistic routing is much more robust can easily get out of such a situation because nodes are always chosen among successful receivers. In addition, we have shown that maintaining multi-paths of non-opportunistic routing can overcome the infinite routing delay problem at the expense of significant overhead.

CHAPTER 5

Performance Evaluation of Broadcast Algorithms: An Analysis-Emulation Hybrid Model

5.1 Introduction

Broadcast algorithms have been widely studied in the context of wireless ad hoc and sensor networks where frequent topological changes occur due to either mobility or duty-cycling. Broadcast is one of the most fundamental operations required by most communication protocols for a variety of reasons. In the context of ad hoc and sensor networks, one of the most important functionality of broadcast is discovery, acquisition, and dissemination of topology information.

Broadcast in its simplest form is a non-discriminatory flooding, where each node transmits (also referred to as rebroadcasts) once for every unique received packet. The main drawback of this approach lies in its transmission redundancy and therefore high energy consumption, especially when node density is high. This redundancy also causes unnecessary transmission contention and packet collisions that lead to performance degradation, a phenomenon referred to as the broadcast storm problem in [31]. It is a particularly severe problem in energy-constrained sensor networks.

To mitigate this problem there have been many broadcast algorithms proposed that aim at reducing the amount of redundant transmissions by preventing some of the nodes from rebroadcasting. Depending on how these nodes are selected, these algorithms may be classified into to two categories. In the first category, these rebroadcasting nodes are determined dynamically based on the outcome of each transmission, see for example [31, 47, 17, 43]. One particular example of this class of algorithms is called *probabilistic broadcast* where each receiving node determines whether to rebroadcast with a fixed probability [31]. In other examples the rebroadcasting nodes are determined based on the number of received neighbors explicitly (e.g. [33, 23, 50, 9]) or implicitly (e.g. by counting the number of received packets, or by estimating the uncovered area, etc. [31]). In the second category, the approach is to pre-determine the set of rebroadcasting nodes in advance of actual transmission using the topological information [36, 48]. The main research challenge there is to choose such a set so as to minimize redundant transmissions while guaranteeing certain coverage requirement.

A good broadcast algorithm should have low redundancy (the amount of transmissions incurred) and high coverage (or reachability, the percentage of nodes that receive the broadcast at the end of the process), among other things. There have been many comparison studies on broadcast algorithms using such performance metrics [49, 56]. Due to the complexity of applying such algorithms to a network, these studies are primarily simulation based. While potentially highly accurate if done correctly, simulation can be very time consuming. A potentially bigger issue arises when we want to evaluate the performance of a larger system with many complex protocols and applications running, of which the broadcast algorithm is only one component. In this case, it is highly desirable to have a performance model that is not only computationally more efficient, but also has well-defined model input and output so that it can potentially be linked and integrated with other performance models that evaluate other components of the same system.

All these desired features point to an analytical performance model. Unfortunately analytical models are very difficult to obtain in this case due to the dynamic and random nature of the system. A common way to get around this problem is to consider large network asymptotics as we have done in Chapter 4, as well as others (see references cited in Chapter 4). Results of this type, while instructive and insightful in the right context, do not always apply to a finite network. To the best of our knowledge to date there has not been a comprehensive yet computationally efficient mathematical framework to evaluate the performance of broadcast algorithms.

The goal of this chapter is to make some progress in this direction. We will start by considering a full state-space model that tries to capture all possible states that a network can encounter during the course of a broadcast process. With this the model also captures all possible sample paths the network can follow, along with the estimated performance along each sample path. Finally, the average performance of the broadcast algorithm can be estimated by averaging over all possible sample paths.

This method is unfortunately not very scalable due to the large state space. We thus next consider methods that do not require the full characterization of the state space. In particular, we observe that there are sample paths that are more *representative* than others; these are sample paths such that the performance along which follow more closely the average performance of the algorithm. Therefore if we could identify these representative sample paths (as little as a single one), then we only need performance computation over a potentially very small set of sample paths, thereby significantly reducing the amount of computation. Motivated by this, in this chapter we present a hybrid model whereby (1) we only focus on a selected subset of sample paths and estimate what happens along these sample paths, and (2) the selection of this subset is based on simple analytical models as well as heuristics. This method can thus be viewed as a combination of analysis and emulation in the hope of obtaining the desired modeling accuracy and computational efficiency. The essential idea is to emulate the representative behavior of the broadcast scheme.

The remainder of this chapter is organized as follows. Section 5.3 and Section 5.4 present the state-space based model and the analysis-emulation model, respectively. In each of both sections, we apply each model to some of exemplary broadcasting strategies as a case study. Their numerical results are shown in Section 5.5. Finally, we conclude in Section 5.6.

5.2 Network Model and Assumptions

We consider a network where n nodes are deployed arbitrarily over a field. Let V be the set of nodes. The set of neighbors of node i, denoted by N_i, is defined as the collection of nodes that are able to receive a packet transmitted from node i with a nonzero probability. Transmission success probability is denoted by a n-by-n matrix $\mathbf{Q}$ where each element q_{ij} represents transmission success probability between nodes i and j. For simplicity of presentation, we will ignore duty-cycling, but noting that including it in this modeling framework is quite straightforward.

We will ignore the coupling of more than one broadcast packets and focus instead on just one packet. The broadcast is assumed to occur in discrete time, or in *rounds*, where in each round a set of nodes that have already received the packet decide whether to retransmit, and if so they retransmit, and by the end of the same round some other nodes will receive the packet. Then in the next round another set of nodes will decide whether to retransmit, and so on, and the same process repeats, till no more nodes transmit, which marks the end of the broadcast process.

Given a particular broadcast scheme π and a topology described above, the performance metrics of interest are as follows: the expected delay to accomplish the broadcast, denoted by $D^{\pi}(n, K, \mathbf{Q})$, the expected number of total transmissions, denoted by $T^{\pi}(n, K, \mathbf{Q})$, and the expected number of nodes which successfully receive the broadcast, denoted by $R^{\pi}(n, K, \mathbf{Q})$.

To illustrate our performance model and make the discussion concrete, we will focus on two example broadcast algorithms in this chapter. However, we emphasize that the proposed modeling framework can be more broadly applied. These two broadcast algorithms were both proposed in [31] as possible ways to mitigate the broadcast storm problem: the probabilistic scheme and the counter-based scheme. The probabilistic scheme is a simple method that suppresses nodes from rebroadcast with a prefixed retransmission probability p. Each node after receiving a packet decides to rebroadcast with this given probability. The exception is the source node, who broadcasts with probability 1. If a node receives the same packet from more than two nodes with the same transmission round, we will assume that the message will be lost due to collision. We assume there is no retransmission in case of collision since in general there is no built-in ACK mechanisms for a broadcast transmission.

The counter-based scheme is slightly more sophisticated where each node counts the number of times that it has received the packet for a randomly chosen length of period (an integer multiple of rounds). The number C_i is initially set to 1 when node i receives the packet for the first time. And node i starts a random timer T_i. During the period T_i, every time it receives a duplicate of the same packet, it increases C_i by one. When the timer expires, its counter C_i is compared to a threshold C_{th} which is greater than equal to 2. If C_i is less than the threshold C_{th}, it rebroadcasts at the start of the next round. Otherwise, it does not. Similar to the probabilistic scheme, if a node receives messages from two more neighbors within the same round, the node cannot receive any of those messages due to collision.

5.3 The State-Space Model

In this section, we describe the state-space model. The general framework is described first, followed by its application to the probabilistic broadcast scheme.

5.3.1 The General Framework

Let us define the state of system S as follows. The state S comprises two parts: the first part is common to all broadcast strategies whereas the second part is algorithm-specific. The common part of S, denoted by S_c, contains a vector $\mathbf{r}$ that represents the nodes which have received the packet, and a vector $\mathbf{f}$ that represents the nodes which are waiting to (possibly) retransmit the packet in the next time step/round. The algorithm-specific part of S is denoted by S_f, which will be described in detail in the next subsection where we specify the broadcast scheme. In this subsection, we mainly discuss the common part S_c.

The state S is updated whenever any transmission occurs in the network. The sequence of states is indexed by times $t = 0, 1, 2, \cdots$, denoting the rounds. Thus, the state at time t (or at the beginning of the t-th round) is expressed given by

$$S_t = \{S_{c,t}, S_{f,t}\}, \quad S_{c,t} \triangleq [\mathbf{r}_t\ \mathbf{f}_t], \tag{5.1}$$

where $\mathbf{r}_t \in \{0,1\}^{|V|}, \mathbf{f}_t \in \{0,1\}^{|V|}$ and $|V|$ is the number of nodes in the network. $\mathbf{r}_t$ is a column vector of $|V|$ elements which take the value of 0 or 1 where 1 indicates that the corresponding node has received the message before the t-th round starts, and 0 otherwise. On the other hand, $\mathbf{f}_t$ is a column vector of $|V|$ elements which take the value of 0 or 1 where 1 indicates that the corresponding node is a possible forwarder at time t. Integrating these two vectors, we have a $|V|$-by-2 matrix $S_{c,t}$.

The next state S_{t+1} depends on the current state S_t only. Let π be the broadcast scheme. The receiving nodes $\mathbf{r}_{t+1}$ and the possible forwarding nodes $\mathbf{f}_{t+1}$ in $S_{c,t+1}$ are calculated by algorithm-specific functions f_r^π and f_f^π, respectively. Then, $\mathbf{r}_{t+1}$ and $\mathbf{f}_{t+1}$ are expressed as

$$\mathbf{r}_{t+1} \in f_r^\pi(S_t), \quad \mathbf{f}_{t+1} \in f_f^\pi(S_t, \mathbf{r}_{t+1}).$$

Note that there can be many such states $S_{t+1} = [\mathbf{r}_{t+1}\,\mathbf{f}_{t+1}]$ visited from the previous state S_t, each with an associated probability. In the above equation, $f_r^\pi(\cdot)$ and $f_f^\pi(\cdot)$ generate such possible states. The detail of these functions are specified according to the broadcast strategy. Similarly, $S_{f,t+1}$ is defined more precisely once the broadcast scheme is given.

We can calculate state transition probabilities $P^\pi(S_{t+1} = S'|S_t = S)$ for all possible S and S' and for all t. This calculation mostly depends on the matrix $\mathbf{Q}$. Let S^* be the set of final states where no further transition is possible, i.e., $f_t = \mathbf{0}$ for some t. For instance, S^* includes the state where all nodes in the network received the packet ($\mathbf{r}_t = \mathbf{1}$ for some t), the state where all nodes except the source $\mathcal{S}$ fails to receive the packet ($\mathbf{r}_t = \mathbf{0}$ for some t), etc. Given all states $S_t, \forall t$ and the transition probabilities $P^\pi(S_{t+1}|S_t)$, we can then calculate the average performance measures as follows. The expected number of nodes which received a broadcast packet is obtained by

$$R^\pi(n, K, \mathbf{Q}) = \sum_{S \in S^*} \left(\sum_t \parallel r_t \parallel \cdot Pr^\pi(S_t = S) \right), \tag{5.2}$$

where we have abused the notation $\parallel \cdot \parallel$ to denote the number of nonzero elements in a vector. Basically, it is the average of the size of r in all final states. To calculate $Pr^\pi(S_t = S)$, consider all sample paths to reach the state S at time t. We further denote by S_t^i the state S_t reached by path i. Then its associated probability to reach

S_n^i is $P^\pi(S_t^i = S) = \prod_{k=1}^t P^\pi(S_k|S_{k-1})$. Thus, $Pr^\pi(S_t = S) = \sum_i P^\pi(S_t^i = S)$. The expected number of total transmissions is calculated similarly by

$$T^\pi(n, K, \mathbf{Q}) = \sum_{S\in S^*}\sum_t \left(\sum_i \left(\sum_{k=1}^t \| \mathbf{f}_k^i \|\right) \cdot Pr^\pi(S_t^i = S)\right), \tag{5.3}$$

where $\mathbf{f}_k^i$ denotes the forwarding nodes at time k along the path i. Thus, $\| \mathbf{f}_k^i \|$ represents the number of such forwarding nodes. Finally, the expected delay is obtained as follows.

$$D^\pi(n, K, \mathbf{Q}) = \sum_{S\in S^*}\left(\sum_t t \cdot Pr^\pi(S_t = S)\right). \tag{5.4}$$

In the next subsection, we show how this state-space method is applied to the probabilistic broadcast scheme as an example. We will often use a column vector $\mathbf{x}_i$ to represent N_i such that each element is 1 if its corresponding index is a member of N_i, and 0 otherwise. The neighborhood matrix $X = [\mathbf{x}_1\, \mathbf{x}_2\, \cdots\, \mathbf{x}_{|V|}]$ is a $|V|$-by-$|V|$ matrix to represent the 1-hop neighbors of all nodes.

5.3.2 Modeling of the Probabilistic Scheme

Consider the probabilistic scheme π where the forwarding probability of each node which received the message is p. For this simple scheme, the algorithm-specific state S_f is not needed, i.e., $S_f = \emptyset$. Therefore, the state at time t is

$$S_t = S_{c,t} = [\mathbf{r}_t\, \mathbf{f}_t],$$

where $\mathbf{r}_t \in \{0,1\}^{|V|}, \mathbf{f}_t \in \{0,1\}^{|V|}$. The initial state S_0 is an all zero matrix except a row for the source $\mathcal{S}$ which contains ones. The transmission from $\mathcal{S}$ occurs with probability 1 and the rest transmissions occur with a forwarding probability p. Thus, we show $f_r^\pi(\cdot)$ and $f_f^\pi(\cdot)$ for the cases of $t = 0$ and $t \geq 1$ separately as follows.

For $t = 0$, the next state S_1 is determined by $N_\mathcal{S}$ the neighbors of $\mathcal{S}$, success probabilities $q_{\mathcal{S}j}, \forall j \in N_\mathcal{S}$, and a forwarding probability p. The functions $f_r^\pi(\cdot)$ and

$f_f^\pi(\cdot)$ are

$$f_r^\pi(S_0) = f_S(N_\mathcal{S}),$$

$$f_f^\pi(S_0, \mathbf{r}_1) = f_S(f_I(\mathbf{r}_1)),\ \forall \mathbf{r}_1 \in f_r^\pi(S_0),$$

where $f_S(\cdot)$ is a function which generates the power set of an input set where each element is represented in a vector form, and $f_I(\cdot)$ is a function which returns the set of indices where an input vector has nonzero values. Conceptually, the function $f_r^\pi(S_0)$ generates all possible cases of successfully receiving nodes while $f_f^\pi(S_0, \mathbf{r}_1)$ generates all possible choices of forwarding nodes among the present receiving nodes. The transition probability is calculated by

$$P^\pi(S_1|S_0) = P^\pi([\mathbf{r}_1\, \mathbf{f}_1]|[\mathbf{r}_0\, \mathbf{f}_0]) = \left(\prod_{j \in f_I(r_1)} q_{\mathcal{S}j}\right) \cdot p^{\|\mathbf{f}_1\|} \cdot (1-p)^{\|\mathbf{r}_1 - \mathbf{f}_1\|}, \tag{5.5}$$

where we can also add an active probability if we wish to model duty-cycling. For $t \geq 1$, $f_r^\pi(\cdot)$ and $f_f^\pi(\cdot)$ are calculated as follows.

$$f_r^\pi(S_t) = f_S(f_I(X \cdot \mathbf{f}_t)) \vee \mathbf{r}_t,$$

$$f_f^\pi(S_t, \mathbf{r}_{t+1}) = f_S(f_I(\mathbf{r}_{t+1} - \mathbf{r}_t)),\ \forall \mathbf{r}_{t+1} \in f_r^\pi(S_t).$$

The transition probability from the state S_t to the state S_{t+1} is obtained by

$$P^\pi(S_{t+1}|S_t) = \left(\prod_{i \in f_I(\mathbf{f}_t)} \prod_{j \in f_I(\mathbf{r}_{t+1} - \mathbf{r}_t)} q_{ij}\right) \cdot p^{\|\mathbf{f}_t\|} \cdot (1-p)^{\|\mathbf{r}_{t+1} - \mathbf{r}_t - \mathbf{f}_t\|}.$$

In the above equation, transition probabilities can be expressed with a collision probability as well as an active probability. For instance, if a neighbor is receiving from two or more transmitters at the same time, the neighbor receives the packet with a certain probability associated with the number of transmitters. We have now completed the state diagram with all states and state transition probabilities. We can next calculate the performance metrics using Eqn. (5.2), Eqn. (5.3) and Eqn. (5.4).

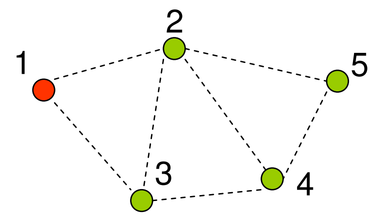


Figure 5.1: A small network with 5 nodes.

5.3.3 An Illustrative Example

As an example, we consider a small network topology depicted in Figure 5.1, where there are 5 nodes and the node 1 is a source. In this particular example, we intend to illustrate how these whole states are generated and how to calculate the metrics given the diagram. The initial state S_0 and the neighbor matrix X are

$$S_0 = [\mathbf{r}_0 \, \mathbf{f}_0] = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}. \tag{5.6}$$

At the time period $t = 1$, the nodes which received the message from node 1 are its 1-hop neighbors with probability 1, which are nodes 2 and 3. According to the forwarding decision of nodes 2 and 3, it leads to four states with different f_1. Each of those states is associated with a transition probability as provided in Eqn. (5.5). For simplicity, we only considered the forwarding probability p while assuming transmission success with probability 1. In addition, if a node receives packets transmitted by more than two neighbors at the same time, we assume that those packets are lost.

Figure 5.2 shows the state diagram of the probabilistic scheme on the example of the above network. Each state can be reached along the sample path from the initial state with an associated probability which is the product of probabilities along the

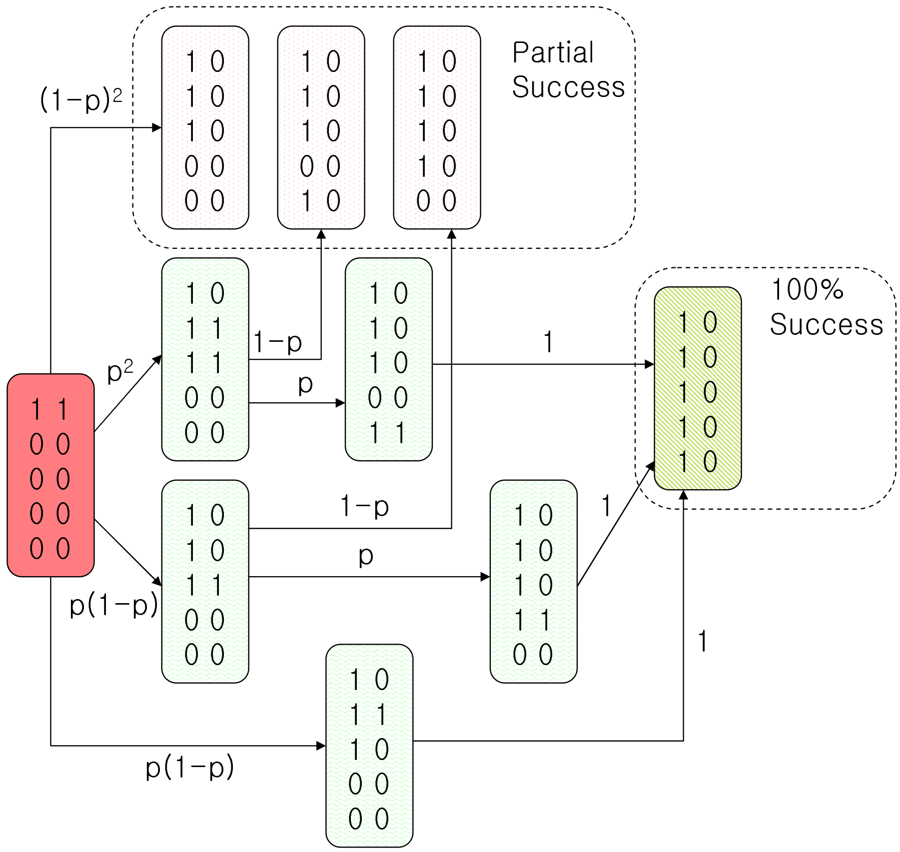


Figure 5.2: The state diagram of the probabilistic Scheme.

path. There are four ending states where three of them are partially successful states (meaning some of nodes are never reached), and one 100%-successful state.

The performance metrics are calculated as follows. The probability of failure to broadcast the entire network is the sum of probabilities to reach three top states as shown in Figure 5.2. It is calculated by $1 - Pr(\text{Partial success}) = p$. The average number of nodes reached at the end of the broadcast process, the reachability of this network, is the weighted sum of $\mathbf{r}_t$ of the last state of each path leading to the partial success states or the complete success state, which is $60(1-p)^2+80p(1-p)^2+100p(1-p)^3+80p^2(1-p)+300p^2(1-p)^2+300p^3(1-p)+100p^4$. The number of forwarders is calculated by averaging $\sum_{t=0}^{t^*} \mathbf{f}_t$ along each path leading to the complete success state where t^* is the time period when the success state is reached. Note that $\hat{\mathbf{f}}_0$ is equal to $\mathbf{f}_0$. Thus, the average number of forwarders is $N_f = (5p^4 + 11p^3(1-p) + 9p^2(1-p)^2 + 2p(1-p)^3)/p$ for this example. At last, we calculate average delay to reach the success states, the average of t^*, which is $(4p^4 + 10p^3(1-p) + 9p^2(1-p)^2 + 2p(1-p)^3)/p$.

5.3.4 Computational Complexity

In this subsection, we show the computational complexity of this model applied to the probabilistic broadcast algorithm. We will assume that the diameter of the network is given by $\mathcal{D}$, which is the maximum number of hops in the network. It has been shown that $\mathcal{D}$ is on the order of $\Theta(\log |V|)$ in a random power law graph [27]. Using this result, the number of states is approximated as follows. In the beginning, only the sender has the message. The number of starting state is 1. The size of the set of states that may be entered during the next step is the size of the powerset of the sender's neighbors. Let us assume that the number of neighbors are approximately d, the average degree of nodes in the network. Then the number of such states are 2^d. In turn, the set of next possible states depends on the number of nodes that received the packet successfully. The size of this next set is the size of powerset of the set of such nodes. This process is repeated until all nodes are reached. Denote the number of states at time period n by N_{S_i}. Then,

$$
\begin{aligned}
N_{S_0} &= 1, \\
N_{S_1} &= 2^d, \\
N_{S_2} &= N_{S_1} \cdot 2^d - N_{S_1} = 2^d(2^d - 1), \\
&\cdots \\
N_{S_{\mathcal{D}}} &= N_{S_{\mathcal{D}-1}} \cdot 2^d - N_{S_{\mathcal{D}-1}} = 2^d(2^d - 1)^{\mathcal{D}-1}.
\end{aligned}
$$

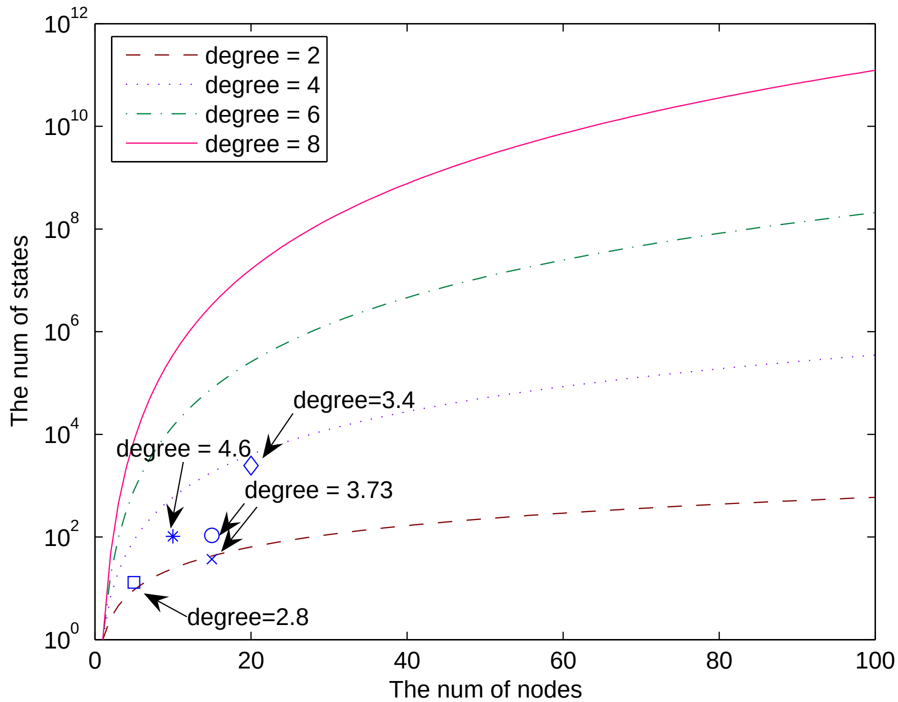


Figure 5.3: Complexity of the state-space based model.

The total number of states is thus

$$\sum_{i=0}^{\mathcal{D}} N_{S_i} = \frac{2^d(2^d-1)^{\mathcal{D}+1} - 2^d}{(2^d-1)(2^d-2)}. \tag{5.7}$$

The above quantity is approximated to be $\Theta(2^{d \log |V|})$, and is evaluated numerically in Figure 5.3. Five marks shown in the figure are obtained from simulation, which match well with analytically obtained graphs.

5.4 The Hybrid Model

In this section, we describe the proposed hybrid model. We first introduce the general framework, and then apply the approach to both the probabilistic and the counter-based broadcast schemes.

5.4.1 General Framework

Consider a source node $\mathcal{S}$ in the network. Denote by X_k the set of forwarding nodes at the k-th step. Transmissions by nodes in X_k are assumed to be synchronized without collision. Y_k represents the nodes that successfully received the broadcast packet transmitted from nodes in X_k. Initially, $X_1 = \{\mathcal{S}\}$ and $Y_0 = \{\mathcal{S}\}$. A neighbor of $\mathcal{S}$, $i \in N_{\mathcal{S}}$, may receive the packet successfully with a probability $q_{\mathcal{S}i}$. Thus,

$$Y_1 = \bigcup_{i \in N_{\mathcal{S}}} \{i\} \cdot I_{\{r_{\mathcal{S}i}=1\}},$$

where $I_{\{\cdot\}}$ is an indicator function and $r_{\mathcal{S}i}$ is a Bernoulli random variable that takes the value 1 with probability $q_{\mathcal{S}i}$ and 0 with probability $1 - q_{\mathcal{S}i}$. The realization of Y_1 is a subset of $N_{\mathcal{S}}$ generated based on $\mathbf{Q}$. Our goal is to select the most representative neighbors among $N_{\mathcal{S}}$ to be successful. While it's not clear how this may be done precisely – if we did know we wouldn't need the current model – there are a number of plausible heuristics to do so. Below we follow one such idea.

We will choose an average number of successful receivers. Specifically, the expected number of successful receivers is calculated as follows.

$$E[|Y_1|] = \sum_{i \in N_{\mathcal{S}}} q_{\mathcal{S}i}.$$

If random duty-cycling is considered, the above equation is simply multiplied by an active probability. With this expected number of transmissions we can choose $\lfloor E[|Y_1|] \rfloor$ number of neighbors among $N_{\mathcal{S}}$. The selection of this set again can be based on a number of heuristics. Below we list a few studied in this chapter:

- Best-q: choose neighbors with the largest values of $q_{\mathcal{S}i}$;
- Random-q: choose neighbors randomly;
- Worst-q: choose neighbors with the smallest values of $q_{\mathcal{S}i}$.

Denote by $\hat{Y}_1$ the set of $\lfloor E[|Y_1|] \rfloor$ number of neighbors chosen by one of above heuristics. Once the successful receivers are determined, the broadcast scheme π decides which nodes among $\hat{Y}_1$ shall forward next. X_2 represents the forwarding nodes at the second step. Thus,

$$X_2 = \bigcup_{i \in \hat{Y}_1} \{i\} \cdot I_{\{f_i^\pi = 1\}},$$

where f_i^π denotes a random variable which is specific to the broadcast scheme π to determine whether node i forwards next. This value can also be determined by some additional variables specific to the broadcast scheme π. (For instance, counter-based broadcast scheme requires counter values for each node.) The selection of forwarding nodes could be probabilistic or deterministic according to π. If it is probabilistic, we make a choice of $\hat{X}_2$ based on one of the heuristics listed above.

At the k-th step, given the current set of transmitting nodes $\hat{X}_k$ and previously received nodes $Y_0, \hat{Y}_1, \cdots, \hat{Y}_{k-1}$, we obtain Y_k as follows.

$$Y_k = \bigcup_{i \in \hat{X}_k} \bigcup_{j \in N_i / \{X_1 \cup \hat{X}_2 \cup \cdots \hat{X}_k\}} \{j\} \cdot I_{\{r_{ij} = 1\}},$$

where r_{ij} is a Bernoulli random variable that takes value 1 with probability q_{ij} and 0 with probability $1 - q_{ij}$. If we consider a collision between packets transmitted in the same round, there is an extra collision probability to be added in the above equation. It follows that the expected number of receivers at the k-th step is

$$E[|Y_k|] = \sum_{j \in \cup_{l \in \hat{X}_k} N_l / \{X_1 \cup \hat{X}_2 \cup \cdots \hat{X}_k\}} 1 - \prod_{i \in \hat{X}_k} (1 - q_{ij}).$$

Based on heuristics listed above, $\hat{Y}_k$ includes $\lfloor E[|Y_k|] \rfloor$ number of neighbors among $(\bigcup_{i \in \hat{X}_k} N_i) / \{X_1 \cup \hat{X}_2 \cup \cdots \hat{X}_k\}$. Then, the next forwarding nodes X_{k+1} is determined by

$$X_{k+1} = \bigcup_{i \in \{\hat{Y}_1 \cup \cdots \hat{Y}_k\} / \{X_1 \cup \cdots X_k\}} \{i\} \cdot I_{\{f_i^\pi = 1\}}, \tag{5.8}$$

where f_i^π denotes a random variable which is specific to the broadcast scheme π to determine whether node i should forward next, given the sequence $X_1, \hat{X}_2, \cdots, \hat{X}_k$, and $Y_0, \hat{Y}_1, \cdots, \hat{Y}_{k-1}$. In addition to these, f_i^π can also depend on variables defined specifically under π. This will be illustrated with example broadcast schemes in the following section. If π is not deterministic, $\hat{X}_{k+1}$ is chosen among $\{\hat{Y}_1 \cup \cdots \hat{Y}_k\}/\{X_1 \cup \cdots X_k\}$ by the heuristic rules listed above.

This procedure is repeated until Y_k is empty. Let $k^* = \inf\{k : Y_k = \varnothing\}$, we attain $D^\pi(n, K, \mathbf{Q}) = k^*$, $T^\pi(n, K, \mathbf{Q}) = X_1 + \sum_{i=2}^{k^*} |\hat{X}_i|$, and $R^\pi(n, K, \mathbf{Q}) = |Y_0 \cup \bigcup_{i=1}^{k^*} \hat{Y}_i|$.

5.4.2 Modeling of Example Broadcast Schemes

In this subsection, we model two broadcast schemes described in Section 5.2 using the analysis-emulation hybrid model.

The Probabilistic Scheme

Consider the probabilistic scheme π where each node randomly decides whether to forward with a probability p. Consider the k-th step of the hybrid model described above. Given the current transmitting nodes $\hat{X}_k$, the set of successful receiving nodes $\hat{Y}_k$ is determined in the way given in the previous section. Then, the set of next forwarding nodes $\hat{X}_{k+1}$ is determined based on π as follows. In Eqn. (5.8), we set $f_i^\pi = s_i$ for $i \in \hat{Y}_k/\{X_1 \cup \cdots X_k\}$ and $f_i^\pi = 0$ otherwise, where s_i is a Bernoulli random variable with probability p of taking the value 1. The expected number of forwarders is given by

$$E[|X_{k+1}|] = |\hat{Y}_k/\{X_1 \cup \cdots X_k\}| \cdot p.$$

We can then select $\lfloor E[|X_{k+1}|] \rfloor$ number of nodes among $\hat{Y}_k/\{X_1 \cup \cdots X_k\}$ using the same heuristics that we used when choosing successful receivers.

The Counter-based Scheme

We next consider the counter-based scheme π where each node rebroadcasts the message if it has received the message less than C_{th} times. The counter-based scheme π determines the next forwarders $\hat{X}_{k+1}$ among nodes which received the message $\hat{Y}_k$ at the k-th step. In Eqn. (5.8), whether the value of f_i^{π} for node i is 1 or 0 is determined subject to π as follows. We define two more additional variables to denote the counter value and the timer value for node i by C_i and T_i, respectively. For the nodes that newly received the message $\hat{Y}_k$, we set their counters by 1. Also, the timers are set using the same heuristic that used when choosing successful receivers. Since T_i is randomly distributed, we attempt to divide nodes in $\hat{Y}_k$ for each time value and assign their timer values accordingly. For instance, nodes with higher success probabilities are assigned by smaller timer values according to the Best-q method. For the nodes that already have the message but received again with the unexpired timer, their counter values increase by 1. When the timer T_i is expired, if C_i is smaller than C_{th}, node i transmits. Thus,

$$f_i^{\pi} = (C_i < C_{th})\,\&\,(T_i == 0),\ \forall i \in \{\hat{Y}_1 \cup \cdots \hat{Y}_k\}/\{X_1 \cup \cdots X_k\}.$$

5.5 Numerical Analysis

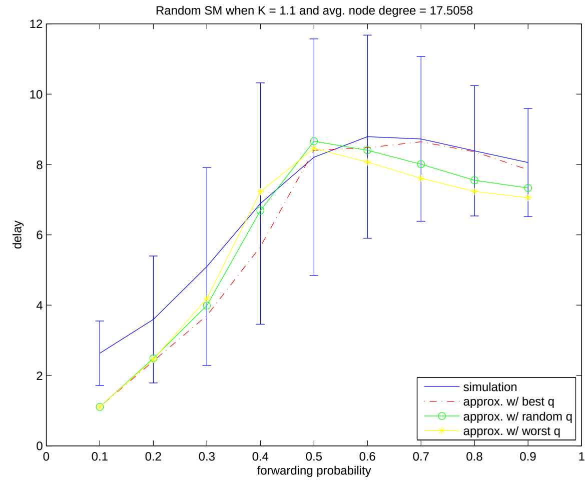
In this section, we validate the hybrid model by comparing its results with simulation results. We generate 200 topologies randomly where each topology contains 100 nodes in the unit square. In this network, any two nodes i and j within the distance less than $R(n) = K\sqrt{\frac{\log n}{n}}$ for some $K > 1$ has transmission success probability q_{ij} which is generated by two different ways. In this simulation, we use $K = 1.1$ so that we have average node degree equal to 17.5. One is that q_{ij} is uniformly and randomly chosen between 0 and 1 without depending on the distance between nodes i

and j. The other is that q_{ij} is linearly decreasing from 1 to 0 as the distance increases from 0 to $R(n)$. Note that these two models were studied in the previous chapter. Given each topology and transmission success probability, we randomly pick a source and broadcast a message based on the probabilistic scheme and the counter-based scheme.

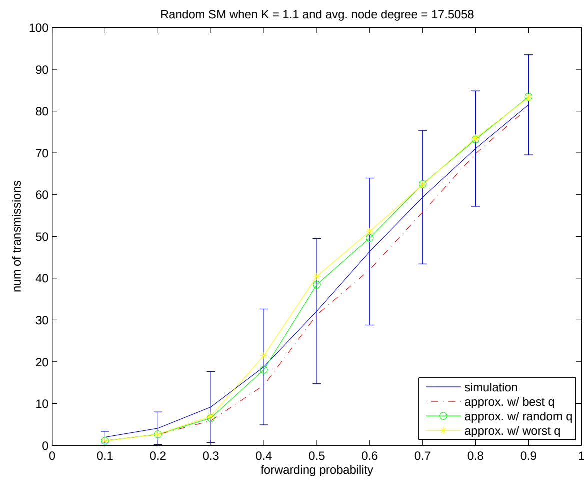
The probabilistic scheme

Consider that transmission success probabilities are drawn from a uniform distribution. We first consider the probabilistic scheme. As described in Section 5.4.2, a heuristic is needed to determine the nodes that successfully receive the packet and the nodes that rebroadcast it. The best heuristic is to capture the average behavior of broadcasting. We introduced three approaches: Best-q, Random-q, and Worst-q in the previous section. Figure 5.4 shows the performance of the probabilistic scheme using the analysis-emulation hybrid model compared to simulation results. The performance obtained by all three approaches are pretty accurate. Note that there is discrepancy when forwarding probability is low. It is because the expected number of forwarding nodes is below 1. In such case, the number is floored in our model and thus no transmission happens.

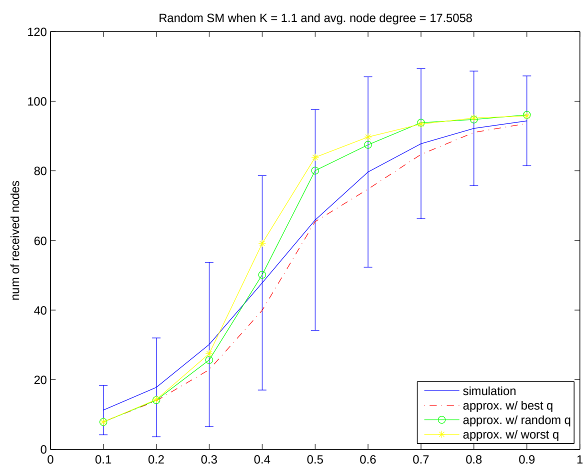
When we use the distance-based success probability model, the heuristic approaches performs differently. As you can see from Figure 5.5, Best-q method is lower and the other two are higher than the average. It is very intuitive in distance-based case. Consider Best-q method which picks nodes with good success probability. In the distance-based model, nodes with higher success probability tends to be closer to the transmitter. Thus, it is more likely that forwarding nodes' transmission range overlaps in large. This implies that many of such transmissions would be redundant. The Worst-q cases is the opposite. The chosen nodes are located far from the trans-



(a) Delay

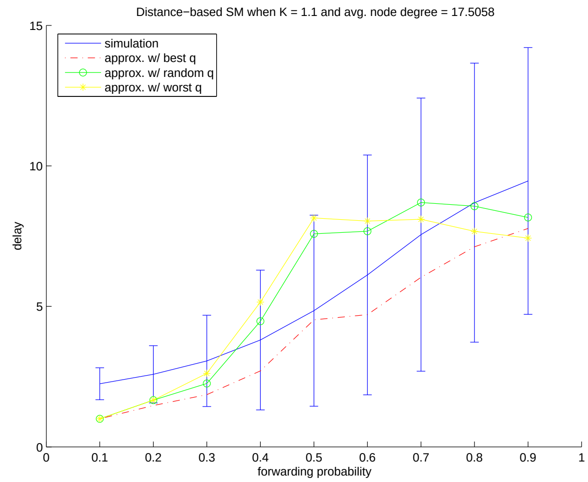


(b) Overhead

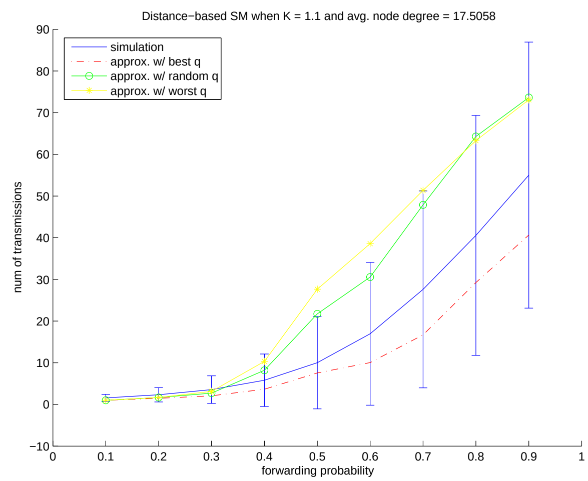


(c) Reachability

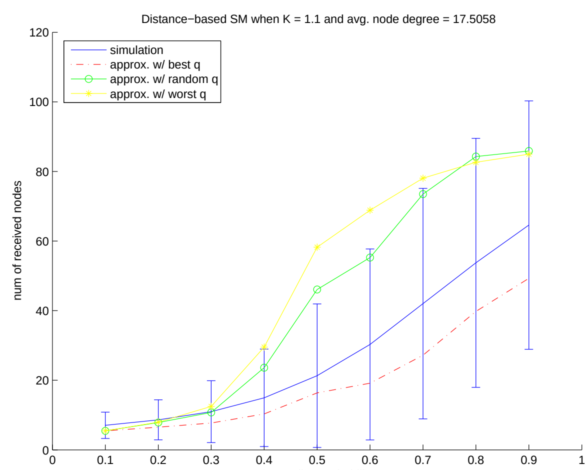
Figure 5.4: The performance of the probabilistic scheme estimated by the analysis-emulation hybrid model under the uniform success probability model when average node degree = 17.5.



(a) Delay



(b) Overhead



(c) Reachability

Figure 5.5: The performance of the probabilistic scheme estimated by the analysis-emulation hybrid model under the distance-based success probability model when average node degree = 17.5.

mitter with high probability. Thus, broadcasting proceeds faster. From this results, we can see that our model estimates the performance of the probabilistic scheme reasonably close to the average obtained by simulations for both success probability models.

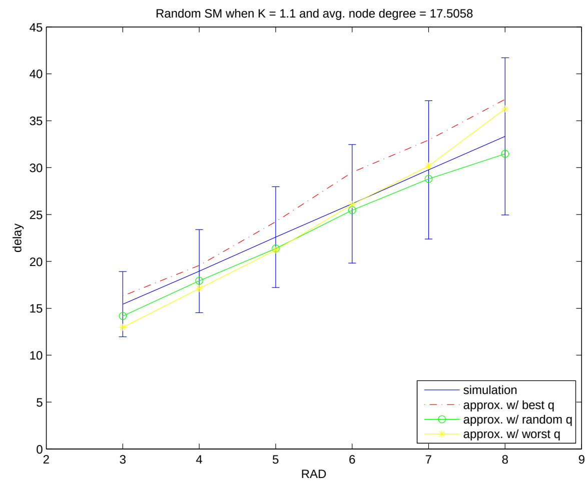
The counter-based scheme

We assume nodes' initial timer values are drawn from a uniform distribution between 1 and RAD. As described in Section 5.4.2, a heuristic is needed to determine the nodes that successfully receive the packet, and also to set their timer values. For this scheme, we use three heuristics: Best-q, Random-q, and Worst-q.

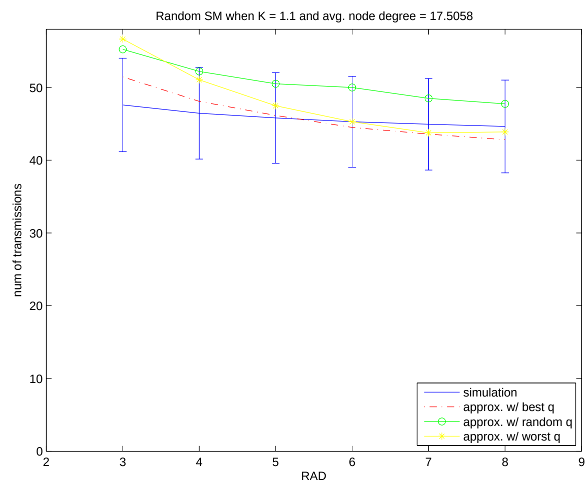
Figure 5.6 and 5.7 show the performance of the counter-based scheme using the analysis-emulation hybrid model compared to simulation results. Under the random success probability model, the proposed analysis-emulation hybrid model with each heuristic works pretty close to the simulation results. However, when success probabilities are generated strictly depending on distance, the selection process of successfully received nodes or their initial timer values assignment based on the suggested heuristics are more critical in the performance than the random success probability case. In the random success probability model, nodes' success probabilities are assigned arbitrarily and thus the selection of nodes or timer values based on the success probabilities are not related to their locations (i.e., the overlaps of the covered area by simultaneous transmissions). Especially, the performance of Best-q exhibits too loose lower-bounds. It is because allowing the nodes with high success probabilities implies that there is very little newly covered area because they are too close to the transmitter under the distance based success probability model. Thus, developing heuristics which works well with a certain success probability model is important to estimate the accurate performance of the broadcast scheme, which is out of scope of this chapter.

5.6 Chapter Summary

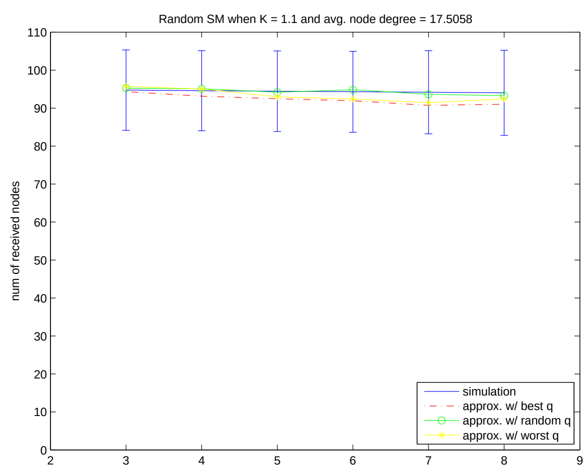
In this chapter we considered broadcasting in a wireless sensor network. We studied the performance of broadcasting schemes under the uncertainties caused by unreliable communications, duty-cycling, etc. Given transmission success probability as well as the network topology, the performance of a broadcast scheme is obtained by a state-space model and a analysis-emulation hybrid model. Numerical results are provided to evaluate the accuracy of the hybrid model.



(a) Delay

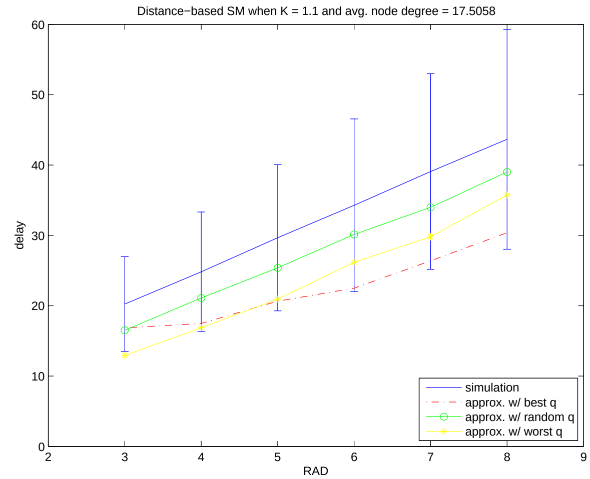


(b) Overhead

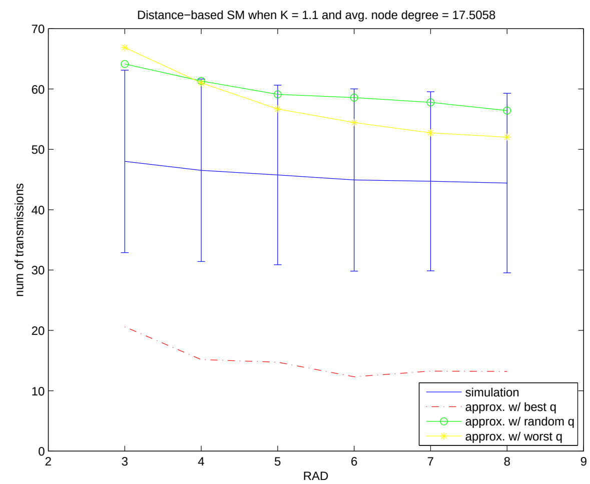


(c) Reachability

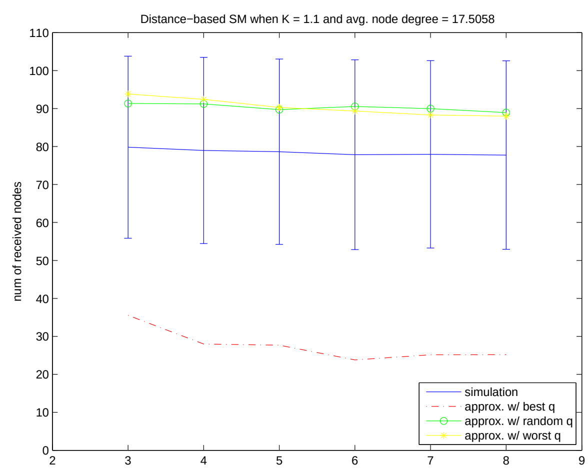
Figure 5.6: The performance of the counter-based scheme estimated by the analysis-emulation hybrid model under the uniform success probability model when average node degree = 17.5.



(a) Delay



(b) Overhead



(c) Reachability

Figure 5.7: The performance of the counter-based scheme estimated by the analysis-emulation hybrid model under the distance-based success probability model when average node degree = 17.5.

CHAPTER 6

Conclusion

This dissertation studied the energy-efficient design of low duty-cycled sensor networks. In Chapter 2, we first derived the fundamental relationship between the amount of redundancy required vs. the achievable reduction in duty cycle for a fixed performance criterion. When sensor nodes are randomly duty-cycled according to a fixed active probability, we derived the sufficient and necessary conditions for the network to be connected as the number of node grows to infinity. These conditions are in the form of the joint scaling behavior of the number of nodes in the network as well as the active probability. From these results, we showed how duty-cycling should be scaled as the network gets denser in order to maintain network connectivity.

In Chapter 3, we studied a routing problem in wireless sensor networks where sensors are duty-cycled. The problem is formulated as an optimal stochastic routing problem (also referred to as opportunistic) in the presence of duty-cycling as well as unreliable wireless channels. We first developed and analyzed an optimal centralized stochastic routing algorithm for a randomly duty-cycled wireless sensor network, and then simplified the algorithm when local sleep/wake states of neighbors are available. We further developed a distributed algorithm utilizing local sleep/wake states of neighbors which performs better than some existing distributed algorithms such as ExOR.

In Chapter 4, we investigated how the routing delay of this type of algorithms scales compared to conventional (non-opportunistic) routing algorithms in a limiting regime where the network becomes dense. In wireless sensor networks, there are uncertainties involved in packet transmission due to fading, node failures, power saving, etc. These characteristics were captured by transmission success probability randomly drawn from a given distribution. We have shown that when the transmission success probabilities are not bounded away from zero, non-opportunistic routing results in extremely large delay whereas opportunistic routing can easily get out of such situation because nodes are always chosen among successful receivers after observing outcome of transmission. In addition, we have shown that maintaining multi-paths of non-opportunistic routing overcomes infinite routing delay problem under the transmission success probability not bounded away from zero. However, since such multi-path routing achieves similar delay performance as opportunistic routing with much higher overhead, we concluded that it is in general advantageous to employ opportunistic routing in wireless networks with respect to delay.

In Chapter 5 we developed an analysis-emulation hybrid model that combines analytical models with elements of numerical simulation to obtain the desired modeling accuracy and computational efficiency. This model has been motivated by the state-space based model which calculates the metrics by averaging over all possible outcomes, which is thus accurate but highly complicated. The analysis-emulation hybrid model emulates the representative behavior of the broadcast scheme. Basically it follows the carefully chosen one of the sample path among all possible outcomes. By using this model, we achieved reasonably accurate performance estimation of broadcasting schemes with much less complexity.

APPENDICES

APPENDIX A

Detailed Calculation for the Optimal Policy in Example 3.1

In Example 3.1, we show the optimal policy $\pi^*(W, A)$ for the network illustrated in Figure 3.1, given $W = \{1, 2, 4\}$ and $A \in F(W)$. Note that $F(W) = \{\{3a, 5a\}, \{3a, 5s\}, \{3s, 5a\}, \{3s, 5s\}\}$. When all nodes are awake, i.e., $A = \{3a, 5a\}$,

$$
\begin{aligned}
&V^{\pi^*}(\{1,2,4\},\{3a,5a\}) \\
&\quad = \max_{i \in \{1,4,I\}} \{-c_i + \sum_{W' \supseteq \{1,2,4\}} \sum_{A' \in F(W')} P^i(W', A' | \{1,2,4\}, \{3a, 5a\}) V^{\pi^*}(W', A')\} \\
&\quad = -1 + \max_{i \in \{1,4,I\}} \{ \sum_{W' \supseteq \{1,2,4\}} \sum_{A' \in F(W')} P^i(W', A' | \{1,2,4\}, \{3a, 5a\}) V^{\pi^*}(W', A')\},
\end{aligned}
$$

where the second equality is based on the assumption of unit cost.

When node 1 is transmitting, possible W' is $\{1, 2, 4\}$ with probability 0.2 or $\{1, 2, 3, 4\}$ with probability 0.8. Then, the term in max function with $i = 1$ is calculated as follows.

$$
\begin{aligned}
&\sum_{A' \in F(\{1,2,4\})} 0.2 P(A') V^{\pi^*}(\{1,2,4\}, A') + \sum_{A' \in F(\{1,2,3,4\})} 0.8 P(A') V^{\pi^*}(\{1,2,3,4\}, A') \\
&\qquad = \sum_{A' \in F(\{1,2,4\})} 0.2 P(A') V^{\pi^*}(\{1,2,4\}, A') + 0.08 V^{\pi^*}(\{1,2,3,4\}, \{5a\})
\end{aligned}
$$

(A.1)

$$
+ 0.72 V^{\pi^*}(\{1,2,3,4\}, \{5s\}),
$$

where $V^{\pi^*}(\{1,2,3,4\},\{5a\})$ and $V^{\pi^*}(\{1,2,3,4\},\{5s\})$ are calculated similarly. Since $\pi^*(\{1,2,3,4\},\{5a\}) = 3$ and $\pi^*(\{1,2,3,4\},\{5s\}) = I$,

$$\begin{aligned} V^{\pi^*}(\{1,2,3,4\},\{5a\}) &= -1 + 0.028V^{\pi^*}(\{1,2,3,4\},\{5a\}) \\ &\qquad + 0.252V^{\pi^*}(\{1,2,3,4\},\{5s\}) + 0.72R_5 \\ V^{\pi^*}(\{1,2,3,4\},\{5s\}) &= -1 + 0.1V^{\pi^*}(\{1,2,3,4\},\{5a\}) + 0.9V^{\pi^*}(\{1,2,3,4\},\{5s\}). \end{aligned}$$

From the above simultaneous equations, we obtain $V^{\pi^*}(\{1,2,3,4\},\{5a\}) = -4.8889 + R_5$ and $V^{\pi^*}(\{1,2,3,4\},\{5s\}) = -14.8889 + R_5$. Thus, Eqn. (A.1) becomes

$$\sum_{A' \in F(\{1,2,4\})} 0.2P(A')V^{\pi^*}(\{1,2,4\},A') - 11.1111 + 0.8R_5.$$

If node 4 is transmitting, i.e., $i = 4$, possible W' is $\{1,2,4\}$ with probability 0.4 or $\{1,2,4,5\}$ with probability 0.6 and the term in max function is

$$\begin{aligned} &\sum_{A' \in F(\{1,2,4\})} 0.4P(A')V^{\pi^*}(\{1,2,4\},A') + \sum_{A' \in F(\{1,2,4,5\})} 0.6P(A')V^{\pi^*}(\{1,2,4,5\},A') \\ &\quad = \sum_{A' \in F(\{1,2,4\})} 0.4P(A')V^{\pi^*}(\{1,2,4\},A') + 0.06V^{\pi^*}(\{1,2,4,5\},\{3a\}) \\ &\qquad\quad + 0.54V^{\pi^*}(\{1,2,4,5\},\{3s\}) \\ &\quad = \sum_{A' \in F(\{1,2,4\})} 0.4P(A')V^{\pi^*}(\{1,2,4\},A') + 0.6R_5. \end{aligned}$$

If I is chosen, it is just $\sum_{A' \in F(\{1,2,4\})} P(A')V^{\pi^*}(\{1,2,4\},A')$.

Let $S \triangleq \sum_{A' \in F(\{1,2,4\})} P(A')V^{\pi^*}(\{1,2,4\},A')$. Then, combining these together, we have

$$\text{(A.2)} \quad V^{\pi^*}(\{1,2,4\},\{3a,5a\}) = \max\{0.2S + 0.8R_5 - 12.1111, 0.4S + 0.6R_5 - 1, S - 1\}$$

Similarly, $V^{\pi^*}(\{1,2,4\},A)$ is calculated for the remaining $A \in F(\{1,2,4\})$. Then, we have $V^{\pi^*}(\{1,2,4\},\{3a,5s\}) = \max\{0.2S + 0.8R_5 - 12.1111, S-1\}$, $V^{\pi^*}(\{1,2,4\},\{3s,5a\}) =$

$\max\{0.4S + 0.6R_5 - 1, S - 1\}$ and $V^{\pi^*}(\{1,2,4\},\{3s,5s\}) = S - 1$. Intuitively, the optimal choices are straight-forward for some A so that $\pi^*(\{1,2,4\},\{3a,5s\}) = 1$, $\pi^*(\{1,2,4\},\{3s,5a\}) = 4$ and $\pi^*(\{1,2,4\},\{3s,5s\}) = I$. Thus,

$$S = 0.01V^{\pi^*}(\{1,2,4\},\{3a,5a\}) + 0.09(0.2S + 0.8R_5 - 12.1111)$$
$$+ 0.09(0.4S + 0.6R_5 - 1) + 0.81(S - 1).$$

For $A = \{3a, 5a\}$, $S = R_5 - 15.7546$ when node 1 is chosen whereas $S = R_5 - 98.6447$ when node 4 is chosen. Thus, the maximum of $V^{\pi^*}(\{1,2,4\},\{3a,5a\})$ is achieved when node 4 is transmitting. Hence, $\pi^*(\{1,2,4\},\{3a,5a\}) = 4$.

APPENDIX B

The proof of Lemma 3.1

If $\pi(W_1, A_1) = \pi^*(W_1, A_1) = r_i$ for some $i \in W_1$ and some A_1, Eqn. (3.9) holds.

Next, we consider the case where both policies π and π^* do not retire but transmit or wait. Suppose $\pi(W_1, A_1) = \pi^*(W_1, A_1) = I$ for some A_1. Let (W_2, A_2) and (W_2^*, A_2^*) be the state after the idle action when in (W_1, A_1) for π and π^*. Obviously, W_2 and W_2^* are the same as W_1 while A_2 is the same as A_2^* for any given sample path, but not necessarily the same as A_1. Both π and π^* pay the idle costs until they reach the state for transmission.

Suppose $\pi(W_1, A_1) = \pi^*(W_1, A_1) = i \in W_1$ for some A_1. Let (W_2, A_2) and (W_2^*, A_2^*) be the state after i's transmission when in (W_1, A_1) for π and π^*. Since node i is transmitting for both policies, $W_2 = W_2^* \supseteq W_1$. Again A_2 is the same as A_2^* for any given sample path. By Eqn. (3.7), we have $V^{\pi}(W_2, A_2) = V^{\pi^*}(W_2^*, A_2^*)$ for $W_2 = W_2^* \supset W_1$ if at least one node receives the packet successfully. Otherwise, we have $W_2 = W_2^* = W_1$ and $A_2 = A_2^*$, which may or may not be different from A_1. Similar to the case of choosing the idle action, by Eqn. (3.8), π chooses the same action (the idle action or transmission but fail) as π^* until it reaches the state where $W_2 \supset W_1$ and any A_2. Hence, Eqn. (3.9) holds.

APPENDIX C

The proof of Lemma 3.2

We define a new policy π on state (W_1, A_1) as follows. Suppose (W_4, A_4) is the state after transmission or idle action by π^* when in (W_2, A_2), where $W_4 \supseteq W_2$. Let π make the same decision as π^* did, which is possible because the node in $W_2 \cup \{I\}$ chosen by π^* is also available in the set $W_1 \cup \{I\} \subseteq W_2 \cup \{I\}$. Let (W_3, A_3) be the state after transmission or idle action by π when in (W_1, A_1). The nodes which are not in W_1 and receive the packet are included in W_3 as well as W_4. However, the nodes which are not in W_2 but in W_1 and receive the packet are included in W_4 whereas W_3 contains all nodes in W_1. Hence, $W_3 \supseteq W_4$. Accordingly, the sleep/wake states of the nodes in $\Omega - W_3$ are the same as A_3 while the nodes in $W_3 - W_4$ may be in different sleep/wake states. Therefore, $A_4 \in F(W_4 | W_3, A_3)$.

At the next step, π acts on (W_3, A_3) by choosing the same node as π^* acts on (W_4, A_4). The process repeats in the same way until π retires when π^* does. Let (W_{f1}, A_{f1}) and (W_{f2}, A_{f2}) be the states at retirement for π and π^*, respectively. We have $W_{f2} \subseteq W_{f1}$ and $A_{f2} \in F(W_{f2} | W_{f1}, A_{f1})$. Total cost incurred by π is the same as π^* because both policies chose the same nodes at every step before retirement. At retirement, π and π^* receive rewards $R(W_{f1})$ and $R(W_{f2})$, respectively. $R(\cdot)$ is a G-index function because it satisfies Eqn. (3.5). Thus, $W_{f1} \supseteq W_{f2}$ results in $R(W_{f1}) \geq R(W_{f2})$ which proves $V^{\pi}(W_1, A_1) \geq V^{\pi^*}(W_2, A_2)$. Finally, because π^* is optimal, $V^{\pi^*}(W_1, A_1) \geq V^{\pi}(W_1, A_1)$ holds. This completes the proof.

APPENDIX D

The proof of Lemma 3.3

The proof is constructive. Let us define π recursively using the following rules:

(D.1) $\qquad \pi(\Omega, A) = \pi^*(\Omega, A), \quad A = \{1, 1, \cdots, 1\},$

$\pi(W - \{j\}, A) = \pi(W, A_i), \quad \forall W \subseteq \Omega, \forall A_i \in F(W), \forall A \in F(W - \{j\}|W, A_i),$

(D.2) $\qquad \forall j \in W : \pi(W, A_i) \neq j, r_j \text{ for } \forall A_i,$

$\pi(W - \{j\}, A) = \pi^*(W - \{j\}, A), \quad \forall W \subseteq \Omega, \forall A \in F(W - \{j\}|W, A_i),$

(D.3) $\qquad \forall j \in W : \pi(W, A_i) = j, r_j \text{ for some } A_i.$

If $N = 1$, the lemma is true directly by Eqn. (D.1). Hence, we assume that $N \geq 2$.

Eqn. (D.2) shows that π satisfies Eqn. (3.10) and Eqn. (3.11) in the first property of this lemma. We now focus on its second property. We prove Eqn. (3.12) by backward induction on the cardinality of W. As the induction basis, we show Eqn. (3.12) is true for $W = \Omega$ and $A = \{1, 1, \cdots, 1\}$. We know that $\pi^*(\Omega, A) = r_i$ for some i such that $i = \arg\max_{k \in \Omega} R_k$ because π^* is optimal. Thus, $V^{\pi^*}(\Omega, A) = R_i$. According to Eqn. (D.1), $\pi(\Omega, A) = r_i$ and $V^{\pi}(\Omega, A) = R_i$ which proves the second equality in Eqn. (3.12). In order to show the first and third equalities, let A_1 be all ones but zero for node $j \in \Omega - \{i\}$. By Eqn. (D.2), we have $\pi(\Omega - \{j\}, A) = \pi(\Omega - \{j\}, A_1) = \pi(\Omega, A) = r_i$ which means that π retires and receives R_i. Thus, $V^{\pi}(\Omega - \{j\}, A) = V^{\pi}(\Omega - \{j\}, A_1) = R_i$. This proves its first equality of Eqn. (3.12).

For $\forall j \in \Omega - \{i\}$, we have $\pi^*(\Omega - \{j\}, A) = r_i$ and $V^{\pi^*}(\Omega - \{j\}, A) = R_i$ because the optimal policy π^* chose node i in Ω which is still in the set $\Omega - \{j\}$ and has the highest reward among nodes in $\Omega - \{j\}$. Similarly, $\pi^*(\Omega - \{j\}, A_1) = r_i$ and $V^{\pi^*}(\Omega - \{j\}, A_1) = R_i$. This proves the last equality of Eqn. (3.12) for $W = \Omega$.

As the induction hypothesis, assume that Eqn. (3.12) holds for any state (W, A) where $|W| = L + 1$ and any possible $A \in F(W)$. If $N = 2$, the basis completes the proof of Eqn. (3.12). Thus, we assume $N > 2$ and $2 \leq L < N$. Consider a state (W_1, A_i) where $|W_1| = L$ and $A_i \in F(W_1)$. If there is $j \in \Omega - W_1$ such that $\pi(W_1 \cup \{j\}, F(W_1 \cup \{j\}|W_1, A_i)) \neq j, r_j$, then we have $\pi(W_1, A_i) = \pi(W_1 \cup \{j\}, F(W_1 \cup \{j\}|W_1, A_i))$ by Eqn. (D.2). By the induction hypothesis, Eqn. (3.12) is true for $W = W_1 \cup \{j\}$. Thus, we have $V^{\pi}(W_1 \cup \{j\} - \{j\}, A_i) = V^{\pi^*}(W_1 \cup \{j\} - \{j\}, A_i)$ which proves the second equality of Eqn. (3.12). That is,

$$V^{\pi}(W_1, A_i) = V^{\pi^*}(W_1, A_i). \tag{D.4}$$

On the other hand, if there is $j \in \Omega - W_1$ such that $\pi(W_1 \cup \{j\}, F(W_1 \cup \{j\}|W_1, A_i)) = j$ or r_j, then by Eqn. (D.3) $\pi(W_1, A_i) = \pi^*(W_1, A_i)$. By the induction hypothesis, we have $V^{\pi}(W, A_i) = V^{\pi^*}(W, A_i)$ for $\forall W \supset W_1$. By Lemma 3.1 we proved that Eqn. (D.4) holds for this case. We have shown that the second equality of Eqn. (3.12) holds for any W_1 where $|W_1| = L$ and any $A_i \in F(W_1)$.

In order to show the first and third equalities of Eqn. (3.12) below, we note that there are two cases: either $\pi(W_1, A_i) = n_i \in W_1 \cup \{I\}$ or $\pi(W_1, A_i) = r_{n_i}$ for all $A_i \in F(W_1)$. Let $j \in W_1$, $j \notin N_{W_1}$ where $N_{W_1} = \bigcup_{i=1}^{m_1} n_i - \{I\}$ and $m_1 = 2^{N-|W_1|}$. Consider the case where $\pi(W_1, A_i) = r_{n_i}$. By Eqn. (D.2) $\pi(W_1 - \{j\}, A') = r_{n_i}, \forall A' \in F(W_1 - \{j\}|W_1, A_i)$. This implies that $V^{\pi}(W_1, A_i) = V^{\pi}(W_1 - \{j\}, A') = R_{n_i}, \forall A' \in F(W_1 - \{j\}|W_1, A_i)$. For an optimal policy π^*, since $W_1 - \{j\} \subset W_1$ and $j \notin N_{W_1}$, by Lemma 3.2 we get $V^{\pi^*}(W_1 - \{j\}, A') \leq V^{\pi^*}(W_1, A_i), \forall A' \in F(W_1 - \{j\}|W_1, A_i)$. By

Eqn. (D.4) we have $V^{\pi^*}(W_1-\{j\}, A') \leq R_{n_i}, \forall A' \in F(W_1-\{j\}|W_1, A_i)$. On the other hand, because $i \in W_1 - \{j\}$ and π^* is an optimal policy, $V^{\pi^*}(W_1 - \{j\}, A') \geq R_{n_i}$, $\forall A' \in F(W_1 - \{j\}|W_1, A_i)$. Hence,

$$V^{\pi^*}(W_1 - \{j\}, A') = R_{n_i}, \forall A' \in F(W_1 - \{j\}|W_1, A_i).$$

This completes the proof of the Eqn. (3.12) for $\pi(W_1, A_i) = r_{n_i}$.

We now prove the first and the third equalities of Eqn. (3.12) in the case of $\pi(W_1, A_i) = n_i \in W_1 \cup \{I\}$. Let us prove the first equality as follow. Let $W \supseteq W_1 - \{j\}$, $j \notin N_W$. We first show the following.

$$\text{(D.5)} \qquad \pi(W, A) \neq j, r_j, \quad \forall A.$$

We prove this in two cases: $j \in W$ and $j \notin W$. If $j \notin W$, $\pi(W, A) \neq j, r_j$ for any A. If $j \in W$, $W_1 \subseteq W$ and $|W| \geq L$. If $|W| = L$, $W = W_1$ and $\pi(W, A) \neq j, r_j$ for any A because of $j \notin N_W$ as given. Assume $|W| > L$. If $\pi(W, A) = j$ for some A, removing all nodes from $W - W_1$ one by one results in $\pi(W_1, A_i) = j$ by Eqn. (D.2), for some A_i which has the same values for nodes in $\Omega - W$ and arbitrary values for nodes in $W - W_1$. This contradicts the hypothesis which is $\pi(W_1, A_i) = n_i \neq j$. Similarly if $\pi(W, A) = r_j$, we have $\pi(W_1, A_i) = r_j$ for some A_i. We have shown that Eqn. (D.5) is true in all cases when $\pi(W_1, A_i) = n_i$. Then, the following is true for any $W' \supseteq W_1$, any $A' \in F(W')$, and any $\tilde{A} \in F(W_1 - \{j\}|W_1, A_i)$:

(D.6)

$$P^{n_i}(W', A'|W_1, A_i) = P^{n_i}(W', A'|W_1-\{j\}, \tilde{A}) + \sum_{A'' \in F(W'-\{j\}|W', A')} P^{n_i}(W'-\{j\}, A''|W_1-\{j\}, \tilde{A}).$$

By Eqn. (D.5) and Eqn.(D.6), we have $V^{\pi}(W_1 - \{j\}, \tilde{A}) = V^{\pi}(W_1, A_i)$ for $\forall \tilde{A} \in F(W_1 - \{j\}|W_1, A_i)$. Next, we prove the third equality of Eqn. (3.12) in case of $\pi(W_1, A_i) = n_i$. By Lemma 3.2 we have $V^{\pi^*}(W_1, A_i) \geq V^{\pi^*}(W_1 - \{j\}, \tilde{A})$ for $\forall \tilde{A} \in$

$F(W_1 - \{j\}|W_1, A_i)$. In addition, π^* is optimal so that $V^{\pi^*}(W_1 - \{j\}, \tilde{A}) \geq V^{\pi}(W_1 - \{j\}, \tilde{A})$. Since $V^{\pi}(W_1, A_i) = V^{\pi^*}(W_1, A_i)$ by Eqn. (D.4),

(D.7)

$$V^{\pi}(W_1 - \{j\}, \tilde{A}) = V^{\pi}(W_1, A_i) = V^{\pi^*}(W_1, A_i) \geq V^{\pi^*}(W_1 - \{j\}, \tilde{A}) \geq V^{\pi}(W_1 - \{j\}, \tilde{A}).$$

This proves Eqn. (3.12) for $\pi(W_1, A_i) = n_i$. We have shown that Eqn. (3.12) is true for all $W \subseteq \Omega$ where $|W| \geq 2$ and all possible $A_i \in F(W)$.

We prove now that π is an optimal Markov policy. As we showed in the second property, $V^{\pi}(W, A) = V^{\pi^*}(W, A)$ for any W where $|W| \geq 2$ and any A. From this relationship Eqn. (3.12), we also have $V^{\pi}(\{i\}, A) = V^{\pi^*}(\{i\}, A)$ for $\forall i \in \Omega$ such that $\pi(\{i\} \cup \{j\}, A) = i$ or I for all A where $j \in \Omega$. If there is no such i left, we still have $V^{\pi}(\{j\}, A) = V^{\pi^*}(\{j\}, A)$ for $\forall j \in \Omega$ by Eqn. (D.3) when $\pi(\{i\} \cup \{j\}, A) = i \in \Omega$ for all A.

Appendix E
The proof of Lemma 3.4

By Eqn. (D.7), we know $V^{\pi^*}(\cdot)$ satisfies

(E-1) $$V^{\pi^*}(W \cup \{i\}, F(W \cup \{i\}|W, A)) \geq V^{\pi^*}(W, A), \forall W \subseteq \Omega, \forall A \in F(W), i \in W.$$

Consider a Markov policy π which satisfies Eqn. (3.10) and Eqn. (3.11). Suppose $V^{\pi}(W, A) = V^{\pi}(W_1, A_1)$ for some $W_1 \subset W$ and $A_1 \in F(W_1)$ s.t. $i \notin W_1$. This implies $\pi(W, A) \neq i$ for all $A \in F(W)$. Then, by Eqn. (3.12) we have $V^{\pi^*}(W, A) = V^{\pi^*}(W - \{i\}, \tilde{A}), \forall \tilde{A} \in F(W - \{i\}|W, A)$. From the above properties of $V^{\pi^*}(\cdot)$, we conclude that $V^{\pi^*}(W, A) \geq V^{\pi^*}(W_1, A_1), \forall W_1 \subseteq W$ and $\forall A_1 \in F(W_1)$. There exists $W_1 \subseteq W$ such that $V^{\pi^*}(W, A) = V^{\pi^*}(W_1, A_1)$ for each $A \in F(W)$. This satisfies Eqn. (3.5).

APPENDIX E

The proof of Lemma 4.5

We have

$$\lim_{n\to\infty} E[X_i^{(n)}]$$
$$= \lim_{n\to\infty} \sum_{m=1}^{\infty} E[X_i^{(n)}|M(n) = m]Pr(M(n) = m)$$
$$= \lim_{n\to\infty} \sum_{m=c_1 \log n}^{c_2 \log n} E[X_i^{(n)}|M(n) = m]Pr(M(n) = m), \tag{E-1}$$

where the second equality holds because by definition there are positive constants c_1 and c_2 such that $\lim_{n\to\infty} Pr(c_1 \log n \le M(n) \le c_2 \log n) = 1$. Given that $E[X_i^{(n)}|M(n) = m]$ is nondecreasing, Eqn. (E-1) is upper-bound by

$$\lim_{n\to\infty} E[X_i^{(n)}]$$
$$\le \lim_{n\to\infty} E[X_i^{(n)}|M(n) = c_2 \log n] \cdot \sum_{m=c_1 \log n}^{c_2 \log n} Pr(M(n) = m)$$
$$= \lim_{n\to\infty} E[X_i^{(n)}|M(n) = c_2 \log n] \cdot \lim_{n\to\infty} Pr(c_1 \log n \le M(n) \le c_2 \log n)$$
$$= \beta^*.$$

Similarly, it is lower-bound by

$$
\begin{aligned}
&\lim_{n\to\infty} E[X_i^{(n)}] \\
&\geq \lim_{n\to\infty} E[X_i^{(n)}|M(n) = c_1 \log n] \cdot \sum_{m=c_1 \log n}^{c_2 \log n} Pr(M(n) = m) \\
&= \beta^*.
\end{aligned}
$$

Hence, $E[X_i^{(n)}]$ is a convergent sequence and the limit is β^* as $n \to \infty$ by the Sandwich Theorem.

APPENDIX F

The proof of Lemma 4.6

For any $\epsilon > 0$, we have

$$Pr(|R(n)\sum_{i=1}^{1/R(n)} X_i^{(n)} - E[X_i^{(n)}]| > \epsilon)$$

$$= Pr(R(n)\sum_{i=1}^{1/R(n)} X_i^{(n)} - \beta^{(n)} > \epsilon) + Pr(R(n)\sum_{i=1}^{1/R(n)} X_i^{(n)} - \beta^{(n)} < -\epsilon). \tag{E-1}$$

By Hoeffding's Inequality, both terms of Eqn. (E-1) are upper-bounded as follows. Since $X_i^{(n)}, 1 \le i \le n$ are bounded and i.i.d., suppose $X_i^{(n)} \in [a, b]$ for all i. Then for any $\epsilon > 0$, we have

$$Pr(R(n)\sum_{i=1}^{1/R(n)} X_i^{(n)} - \beta^{(n)} > \epsilon) = Pr(\sum_{i=1}^{1/R(n)} X_i^{(n)} - \frac{\beta^{(n)}}{R(n)} > \frac{\epsilon}{R(n)})$$

$$\le \exp(-\frac{2\epsilon^2}{R(n)(b-a)^2}). \tag{E-2}$$

And,

$$Pr(R(n)\sum_{i=1}^{1/R(n)} X_i^{(n)} - \beta^{(n)} < -\epsilon) \le \exp(-\frac{2\epsilon^2}{R(n)(b-a)^2}). \tag{E-3}$$

Next, we show $\sum_n Pr(|R(n)\sum_{i=1}^{1/R(n)} X_i^{(n)} - \beta^{(n)}| > \epsilon) < \infty$ as follows. From Eqn. (E-2) and Eqn. (E-3), we have

$$\sum_n Pr(|R(n)\sum_{i=1}^{1/R(n)} X_i^{(n)} - \beta^{(n)}| > \epsilon) \le \sum_n 2\exp(-\frac{2\epsilon^2}{R(n)(b-a)^2})$$

$$= \sum_n 2\exp(-\frac{2\epsilon^2}{K(b-a)^2}\sqrt{\frac{n}{\log n}}). \tag{E-4}$$

We show the convergence of $\sum_n Pr(|R(n)\sum_{i=1}^{\frac{1}{R(n)}} X_i^{(n)} - \beta^{(n)}| > \epsilon)$ by showing the convergence of the infinite series in Eqn. (E-4). There is positive N such that for $n \geq N$ we have $\sqrt{\frac{n}{\log n}} > \alpha \log n$ for some $\alpha > 1$. Then,

$$
\begin{aligned}
& \sum_n 2\exp(-\frac{2\epsilon^2}{K(b-a)^2}\sqrt{\frac{n}{\log n}}) \\
& < \sum_{n=1}^{N} 2\exp(-\frac{2\epsilon^2}{K(b-a)^2}\sqrt{\frac{n}{\log n}}) + \sum_{n=N}^{\infty} 2e^{-\frac{2\epsilon^2}{K(b-a)^2}\alpha\log n} \\
& = \sum_{n=1}^{N} 2\exp(-\frac{2\epsilon^2}{K(b-a)^2}\sqrt{\frac{n}{\log n}}) + \sum_{n=N}^{\infty} 2e^{\log n^{-\frac{2\epsilon^2}{K(b-a)^2}\alpha}} \\
& = \sum_{n=1}^{N} 2\exp(-\frac{2\epsilon^2}{K(b-a)^2}\sqrt{\frac{n}{\log n}}) + \sum_{n=N}^{\infty} 2n^{-\frac{2\epsilon^2}{K(b-a)^2}\alpha}
\end{aligned}
\tag{E-5}
$$

In Eqn. (E-5), the first term is finite because it is a finite sum and the second term is also finite for some α such that $\alpha > \min\{\frac{K(b-a)^2}{2\epsilon^2}, 1\}$. Thus, we have $\sum_n Pr(|R(n)\sum_{i=1}^{\frac{1}{R(n)}} X_i^{(n)} - \beta^{(n)}| > \epsilon) < \infty$. By Borel-Cantelli lemmas, we have

$$\lim_{n\to\infty} R(n)\sum_{i=1}^{\frac{1}{R(n)}} X_i^{(n)} = \lim_{n\to\infty} \beta^{(n)} = \beta^* \text{ (almost surely).}$$

APPENDIX G

The Proof of Lemma 4.7

Proof. Let $Y_k^{(n)}$ be the hop progress toward the destination at the kth hop. It is well-approximated by $L_k^{(n)} \cos \alpha_k^{(n)}$ as shown in [44]. More precisely, for all $j < h(n)$ and $\gamma(n) > 0$,

$$R(n) \sum_{k=1}^{j} \tilde{L}_k^{(n)} \cos \alpha_k^{(n)} - (R(n))^2 \sum_{k=1}^{j} \frac{(\tilde{L}_k^{(n)})^2}{\gamma(n)} < \sum_{k=1}^{j} Y_k^{(n)} < R(n) \sum_{k=1}^{j} \tilde{L}_k^{(n)} \cos \alpha_k^{(n)}. \tag{E-1}$$

We will show that upper and lower bounds of $h(n)$. First, its upper bound is proven by contradiction. Suppose $h(n) \leq \frac{k_2}{R(n)}, \forall k_2 > \frac{1}{\beta^*}$ is not true. Then, there exists a subsequence $n_i, i = 1, 2, \cdots$ such that $h(n_i) > \frac{k_2}{R(n_i)}$. Eqn. (4.1) implies that for all i

$$\sum_{k=1}^{\frac{k_2}{R(n_i)}} Y_k^{(n_i)} < R(n_i) < 1.$$

However, Eqn. (E-1), which holds for $j < h(n)$, holds when $j = \frac{k_2}{R(n_i)}$. Then, by Lemma 4.6, we have

$$R(n) \sum_{k=1}^{j} \tilde{L}_k^{(n_i)} \cos \alpha_k^{(n_i)} \to k_2 \beta^* > 1. \tag{E-2}$$

Since $\tilde{L}_k^{(n_i)} \leq 1$ and $\frac{M(n)}{\gamma(n)} \to 0$, we obtain

$$(R(n_i))^2 \sum_{k=1}^{j} \frac{(\tilde{L}_k^{(n_i)})^2}{\gamma(n_i)} \to 0. \tag{E-3}$$

Therefore, by Eqn. (E-2) and Eqn. (E-3), we have the lower bound of Eqn. (E-1) converges to $k_2\beta^*$ which is greater than 1. So, we have $\lim_{n\to\infty}\sum_{i=1}^{\frac{k_2}{R(n_i)}} Y_k^{(n_i)} > 1$, which contradicts Eqn. (E-2). Therefore, we prove the upper-bound of $h(n)$.

The lower bound of $h(n)$ is proved by contradiction as well. Suppose $h(n) < \frac{k_1}{R(n)}$ for some $k_1 < \frac{1}{\beta^*}$. Since $h(n) > \frac{1}{R(n)}$, there is a subsequence n_i, $i = 1, 2, \cdots$ such that for some $h_1 \in (1, k_1)$,

$$R(n_i)h(n_i) \to h_1. \tag{E-4}$$

Then,

$$|\sum_{k=1}^{h(n_i)} \tilde{L}_k^{(n_i)} \cos\alpha_k^{(n_i)} - \sum_{k=1}^{h_1/R(n_i)} \tilde{L}_k^{(n_i)} \cos\alpha_k^{(n_i)}| \leq |h_1/R(n_i) - h(n_i)|R(n_i). \tag{E-5}$$

Thus, the upper bound in Eqn. (E-1) is again upper-bounded by

$$1 = \sum_{k=1}^{h(n_i)} Y_k^{(n_i)} \leq \sum_{k=1}^{h(n_i)} \tilde{L}_k^{(n_i)} \cos\alpha_k^{(n_i)} \leq \sum_{k=1}^{h_1/R(n_i)} \tilde{L}_k^{(n_i)} \cos\alpha_k^{(n_i)} + |h_1/R(n_i) - h(n_i)|R(n_i),$$

where the first term approaches to h_1 by Eqn. (4.5) and the second term approaches to 0. Thus, it contradicts $\sum_{k=1}^{h(n_i)} Y_k^{(n_i)} = 1$. Therefore, we have shown the upper bound of $h(n)$. $\square$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Christian Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, June 2002.

[2] Vartika Bhandari and Nitin H. Vaidya. Reliable broadcast in wireless networks with proababilistic failures. In *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, May 2007.

[3] M. Bhardwaj, T. Garnett, and A. P. Chandrakasan. Upper bounds on the lifetime of sensor networks. In *Proc. of the International Conference on Communications(ICC)*, volume 3, June 2001.

[4] Sanjit Biswas and Robert Morris. Opportunistic routing in multi-hop wireless networks. In *Workshop on Hot Topics in Networks (HotNets-II)*, November 2003.

[5] Sanjit Biswas and Robert Morris. ExOR: Opportunistic multi-hop routing for wireless networks. In *Conference of the Special Interest Group on Data Communication (SIGCOMM)*, August 2005.

[6] J. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 1, March 2000.

[7] J-H. Chang and L. Tassiulas. Routing for maximum system lifetime in wireless ad-hoc networks. In *Allerton Conference on Communication, Control, and Computing*, September 1999.

[8] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM)*, August 2003.

[9] Fei Dai and Jie Wu. Performance analysis of broadcast protocols in ad hoc networks based on self-pruning. *IEEE Transactions on Parallel and Distributed Systems,*, 15(11), November 2004.

[10] T. V. Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2003.

[11] William Feller. *An Introduction to Probability Theory and Its Applications.* John Wiley & Sons, 1968.

[12] Dario Ferrara, Laura Galluccio, Alessandro Leonardi, Giacomo Morabito, and Sergio Palazzo. Macro: An integrated mac/routing protocol for geographic forwarding in wireless sensor networks. In *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 2005.

[13] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks. Technical report ucla/csd-tr 02-0013, February 2002.

[14] P. Gupta and P. R. Kumar. Critical power for asymptotic connectivity in wireless networks. In *Stochastic Analysis, Control, Optimization and Application: A Volume in Honor of W.H. Fleming (W.M.McEneany, G.Yin, and Q.Zhang, eds.)*, Boston, MA:Birkhauser, March 1998.

[15] Piyush Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2), March 2000.

[16] Zygmunt J. Haas and Marc R. Pearlman. The performance of query control schemes for the zone routing protocol. *IEEE/ACM Transactions on Networking*, 9(4), August 2001.

[17] Marc Heissenbuttel, Torsten Braun, Markus Walchli, and Thomas Bernoulli. Optimized stateless broadcasting in wireless multi-hop networks. In *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, April 2006.

[18] Chihfan Hsin and Mingyan Liu. Network coverage using low-duty cycled censors: Rangom & coordinated sleep algorithms. In *IEEE Information Processing in Sensor Networks (IPSN)*, 2004.

[19] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *IEEE Transactions on Networking*, 11(1), 2003.

[20] David B. Johnson and David A. Maltz. *Dynamic source routing in ad hoc wireless networks.* Mobile computing, Kluwer Academic Publishers, 1996.

[21] Alireza Keshavarz-Haddad, Vinay Ribeiro, and Rudolf Riedi. DRB and DCCB: Efficient and robust dynamic broadcast for ad hoc and sensor networks. In *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, June 2007.

[22] Ben Liang and Zygmunt J. Haas. Hybrid routing in ad hoc networks with a dynamic virtual backbone. *IEEE Transactions on Wireless Communicatios*, 5(6), June 2006.

[23] Hyojun Lim and Chongkwon Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *the ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, August 2000.

[24] Christopher Lott and Demosthenis Teneketzis. Stochastic routing in ad-hoc networks. *IEEE Transactions on Automatic Control*, 51(1), 2006.

[25] Gang Lu and Bhaskar Krishnamachari. Minimum latency joint scheduling and routing in wireless sensor networks. *Ad Hoc Networks*, 5(6), August 2007.

[26] Gang Lu, Narayanan Sadagopan, Bhaskar Krishnamachari, and Ashish Goel. Delay efficient sleep scheduling in wireless sensor networks. In *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 2005.

[27] Linyuan Lu. The diameter of random massive graphs. In *Proceedings of the Twelfth ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2001.

[28] R. Meester and R. Roy. *Continuum Percolation.* Cambridge University Press, 1996.

[29] Tommaso Melodia, Dario Pompili, and Ian F. Akyildiz. Optimal local topology knowledge for energy efficient geographical routing in sensor networks. In *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 2004.

[30] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *ACM Mobile Networks and Applications Journal*, October 1996.

[31] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM)*, August 1999.

[32] Andrzej Pelc and David Peleg. Feasibility and complexity of broadcasting with random transmission failures. In *ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, July 2005.

[33] Wei Peng and Xi-Cheng Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM)*, August 2000.

[34] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination sequenced distance vector routing (dsdv) for mobile cmputers. In *Conference of the Special Interest Group on Data Communication (SIGCOMM)*, October 1994.

[35] Thomas K. Philips, Shivendra S. Panwar, and Asser N. Tantawi. Connectivity properties of a packet radio network model. *IEEE Transactions on Information Theory*, 35(5), 1989.

[36] Amir Qayyum, Laurent Viennot, and Anis Laouiti. Multipoint relaying for flooding broadcast messages in mobilewireless networks. In *The Annual Hawaii International Conference on System Sciences (HICSS)*, January 2002.

[37] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Optimizaing sensor networks in the energy-latency-density design space. *IEEE Transactions on Mobile Computing*, 1(1), 2002.

[38] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Topology management for sensor networks: Exploiting latency and density. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, June 2002.

[39] Karim Seada, Marco Zuniga, Ahmed Helmy, and Bhaskar Krishnamachari. Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2004.

[40] Rahul C. Shah and Jan M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, March 2002.

[41] S. Shakkottai, R. Srikant, and N. Shroff. Unreliable sensor grids: Coverage, connectivity and diameter. In *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 2, April 2003.

[42] Mihail L. Sichitiu. Cross-layer scheduling for power efficiency in wireless sensor networks. In *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 2004.

[43] Fred Stann, John Heidemann, Rajesh Shroff, and Muhammad Zaki Murtaza. Rbp: Robust broadcast propagation in wireless networks. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, November 2006.

[44] S. Subramanian and S. Shakkottai. Geographic routing with limited information in sensor networks. In *submitted to IEEE Information Processing in Sensor Networks (IPSN)*, 2005.

[45] Sundar Subramanian, Sanjay Shakkottai, and Ari Arapostathis. Broadcasting in sensor networks: The role of local information. In *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, April 2006.

[46] Di Tian and Nicolas D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002.

[47] Yu-Chee Tseng, Sze-Yao Ni, and En-Yu Shih. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. *IEEE Transactions on Computers*, 52(5), May 2003.

[48] Peng-Jun Wan, K.M Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. In *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, June 2002.

[49] Brad Williams and Tracy Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, June 2002.

[50] Jie Wu and Fei Dai. Broadcasting in ad hoc networks based on self-pruning. In *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, April 2003.

[51] F. Xue and P. R. Kumar. The number of neighbors needed for connectivity of wireless networks. *Wireless Networks*, 10, 2004.

[52] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, June 2002.

[53] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. Usc/isi technical report isi-tr-567, January 2003.

[54] M. Younis, M. Youssef, and K. Arisha. Energy-aware routing in cluster-based sensor networks. In *IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS)*, October 2002.

[55] Dan Yu and Hui Li. On the definition of ad hoc network connectivity. In *International Conference on Communication Technology (ICCT)*, April 2003.

[56] Hao Zhang and Zhong-Ping Jiang. Analysis of two ad hoc broadcasting protocols. In *IEEE Wireless Communications and Networking Conference (WCNC)*, March 2004.

[57] Zifei Zhong and Srihari Nelakuditi. On the efficacy of opportunistic routing. In *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, June 2007.

[58] N. Zhou, H. Wu, and A. A. Abouzeid. Reactive routing overhead in networks with unreliable nodes. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM)*, August 2003.