PERFORMANCE ANALYSIS AND OPTIMIZATION OF QUERY-BASED WIRELESS SENSOR NETWORKS

by

Guvenc Degirmenci

B.S., Middle East Technical University, 2006

M.S., Middle East Technical University, 2008

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2013

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Guvenc Degirmenci

It was defended on

March 18, 2013

and approved by

Jeffrey P. Kharoufeh, PhD, Associate Professor

Oleg A. Prokopyev, PhD, Associate Professor

Bryan A. Norman, PhD, Associate Professor

Rusty O. Baldwin, PhD, Professor

Dissertation Director: Jeffrey P. Kharoufeh, PhD, Associate Professor

PERFORMANCE ANALYSIS
AND OPTIMIZATION OF QUERY-BASED
WIRELESS SENSOR NETWORKS

Guvenc Degirmenci, PhD

University of Pittsburgh, 2013

This dissertation is concerned with the modeling, analysis, and optimization of large-scale, query-based wireless sensor networks (WSNs). It addresses issues related to the time sensitivity of information retrieval and dissemination, network lifetime maximization, and optimal clustering of sensor nodes in mobile WSNs. First, a queueing-theoretic framework is proposed to evaluate the performance of such networks whose nodes detect and advertise significant events that are useful for only a limited time; queries generated by sensor nodes are also time-limited. The main performance parameter is the steady state proportion of generated queries that fail to be answered on time. A scalable approximation for this parameter is first derived assuming the transmission range of sensors is unlimited. Subsequently, the proportion of failed queries is approximated using a finite transmission range. The latter approximation is remarkably accurate, even when key model assumptions related to event and query lifetime distributions and network topology are violated.

Second, optimization models are proposed to maximize the lifetime of a query-based WSN by selecting the transmission range for all of the sensor nodes, the resource replication level (or time-to-live counter) and the active/sleep schedule of nodes, subject to connectivity and quality-of-service constraints. An improved lower bound is provided for the minimum transmission range needed to ensure no network nodes are isolated with high probability. The optimization models select the optimal operating parameters in each period of a finite planning horizon, and computational results indicate that the maximum lifetime can be significantly extended by adjusting the key operating parameters as sensors fail over time due to energy depletion.

Finally, optimization models are proposed to maximize the demand coverage and minimize the costs of locating, and relocating, cluster heads in mobile WSNs. In these models, the locations of mobile sensor nodes evolve randomly so that each sensor must be optimally assigned to a cluster head during each period of a finite planning horizon. Additionally, these models prescribe the optimal times at which to update the sensor locations to improve coverage. Computational experiments illustrate the usefulness of dynamically updating cluster head locations and sensor location information over time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I would like start by expressing my gratitude to my dissertation advisor, Dr. Jeffrey P. Kharoufeh, for his guidance and feedback throughout my graduate studies at the University of Pittsburgh. This research was supported by a grant from the National Science Foundation (CNS-0831707), and complimentary simulation software was provided by OPNET. Thanks are also due to my committee members, Drs. Oleg A. Prokopyev, Rusty O. Baldwin and Bryan A. Norman, for their insightful questions and constructive comments on the dissertation.

My colleagues Murat Kurt, Yasin Ulukus, Erhun Ozkan, Anahita Khojandi and John Flory also deserve my appreciation for helping to shape my research with their fruitful thoughts. Many thanks are due to Patrick Glen O'Donnell, John Migliozzi and Christopher Bistline, undergraduate research assistants, for gathering reference materials and providing computer code when needed. I am also grateful to my former advisor, Dr. Meral Azizoglu of the Middle East Technical University (METU), for her moral support throughout the course of the Ph.D. program.

Last, but certainly not least, I owe so much to my wife Sara Baskentli for her patience and love. She was always there and stood by me through the good times and bad.

1.0 INTRODUCTION

1.1 BASIC DEFINITIONS AND CONCEPTS

A wireless sensor network (WSN) is a collection of autonomous sensing devices (called sensor nodes) linked by a wireless communication medium to gather and convey information about the environment in which they are deployed. The sensor nodes are usually inexpensive and are capable of sensing, communication and some level of computation. Large-scale WSNs are emerging in such diverse applications as military, environmental, health monitoring, industrial processes monitoring, infrastructure security and residential use. The ever-increasing interest in WSNs stems from their ability to sense and convey critical information about objects, their surroundings, and interactions between them autonomously.

The hardware in a typical wireless sensor node includes a radio transceiver, a processor, memory, a power supply in the form of a small battery, and one or more sensors. The processor is responsible for scheduling tasks, processing data and controlling the other components. Sensors produce a measurable response signal to a physical condition such as temperature or humidity. Unlike traditional wireless devices like cell phones and personal digital assistants, wireless sensor nodes do not rely on pre-determined communication infrastructures. Each of the nodes communicates with its local peers and forwards any data cooperatively in a multi-hop fashion. The wireless communication is performed by the integrated transceiver. While energy is consumed by sensing and processing, the most energy-consuming activity is communication (i.e., receiving and transmitting packets).

Due to their small physical dimensions, the sensing nodes have very limited energy reserves, local memory, and computational capabilities. Moreover, to conserve power and alleviate contention for access to the transmission medium, it is desirable to limit each node's transmission range to that required to ensure a connected network. Although each WSN application has its own design, certain design issues are shared by all applications. Network lifetime, for instance, is directly correlated with

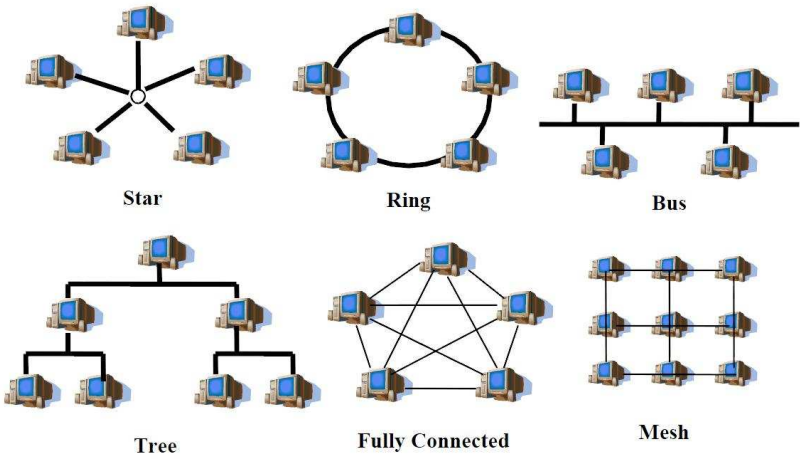


Figure 1: Examples of basic network topologies (see [68]).

the lifetime of individual sensor nodes. Therefore, it is crucial to prolong the lifetime of those nodes. In many applications, information gathered by the WSNs is time-sensitive. Processing and/or communication may cause unacceptably long delays. On the other hand there is a tradeoff between meeting the quality-of-service (QoS) requirements and energy expenditure at a node, and algorithms have been devised to address these issues. The optimal operation of WSNs has a profound impact on their performance. WSNs are more flexible than wired networks because with no wires or cables to route, installation time and cost is significantly reduced. They can accommodate the addition of sensor nodes, upgrades and expansions without infrastructure changes. Moreover, for some applications, wires and cables make monitoring impractical. For instance, moving or rotating equipment, such as a wind turbine blade, add complexity to the system and require additional maintenance if a traditional wired monitoring system is used. A WSN eliminates the complexity of monitoring these components. Moreover, when a WSN is used for diagnostic purposes, each sensor can reduce component downtime by providing a critical component's current health status. With the data obtained, preventive maintenance actions can be taken.

Advances in hardware technology and engineering design have accelerated the development of WSNs to monitor a wide variety of environmental conditions. Depending on application requirements, one of several basic network topologies, including random mesh, star, ring, bus and tree-shaped, might be used. Some basic network topologies are shown in Figure 1. Sensor deployment differs for every network topology. For example, groups of sensors are sometimes dropped

out of a moving aircraft to maintain random sensor deployment in the field. In other applications, sensors must be placed one by one by a robot or a human. After the initial deployment, sensors might even be relocated either intentionally or due to environmental effects.

1.2 APPLICATIONS OF WSNS

WSNs can monitor a wide variety of conditions such as temperature, humidity, pressure, noise levels, the presence or absence of certain substances, vehicular movement, lightning conditions, etc. (see [5]). Although WSNs were initially designed for military applications, they have been used in numerous civilian and industrial applications. In this section, we review common WSN applications and give some examples.

Military applications: The rapid setup, self-organization and fault/failure tolerance characteristics of WSNs make them very promising for military applications (see [91, 114]). For instance, remote terrains and paths can be monitored for the movements of enemy forces using WSNs. The destruction of WSN sensor nodes by opposing forces does not affect the military actions significantly, because sensor nodes are disposable, inexpensive and the node deployment is typically dense. Moreover, new sensor nodes can be added or new WSNs can be deployed at any time.

Infrastructure security: In addition to military applications, WSNs are also used for the security of critical facilities such as fossil or nuclear power plants and airports. For instance, Flammini et al. [40] proposed an early warning system based on WSNs to monitor structural failures and security threats including both natural hazards and terrorist attacks in railway infrastructures.

Environmental monitoring: Environmental applications of WSNs include, but are not limited to, tracking of wildlife including birds, insects or small animals, forest fire surveillance, volcano monitoring, bio-complexity mapping of the environment, irrigation, flood detection, pollution monitoring and weather forecasting. For instance, WSNs can be used to collect data in glacial environments. Traditional sensor nodes cannot be installed inside the ice and the sub-glacial sediment without disturbing the environment; therefore, WSNs facilitate the collection of data in environmental studies (see [73, 82]). Another example is volcano monitoring where WSNs are used to monitor volcanic activity, as shown in the Figure 2, with greater spatial resolution than traditional wired monitoring systems.

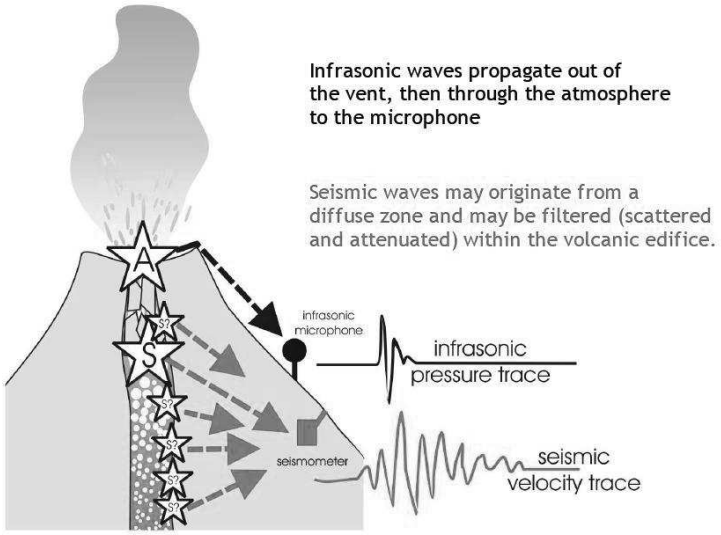


Figure 2: WSN application of volcano monitoring (see [105]).

Health monitoring: In healthcare, WSNs can monitor health indicators such as blood pressure and body temperature. Physiological data can be gathered and stored for a long period of time. WSNs can recognize predefined symptoms and physical signs of an illness in a subject patient without affecting the patient's quality-of-life. The development of wearable and implantable sensors introduces a special type of WSN called a body sensor network (see [109, 77]) which is primarily used to monitor patients with acute diabetes, epilepsy, other debilitating neurological disorders and chronic cardiac diseases. As WSNs can collect health-related data and report it automatically, they have the potential to reduce the cost and inconvenience of regular physician visits.

Monitoring industrial processes: Monitoring corrosion, wear and aging in machines or machine components using manual processes, such as exposure of corrosion (weight loss) coupons and offline probes, to ensure safe operation is extremely costly and time consuming. Moreover, many times problems cannot be identified before a substantial damage or a failure occurred (see [58]). Catastrophic failures can be prevented by monitoring certain health measures of machines using WSNs (see [71, 45]). For instance, WSNs are used for corrosion monitoring in a wide variety of industries including oil, gas and petrochemical.

Embedded structural health monitoring: WSNs are used to assess the health status of a structure (e.g. a bridge or building), or to detect the changes that affect its performance without interfering with the operation of the structure. For example, structural health monitoring of

automobile and aircraft tires using WSNs is gaining prominence. WSNs can be installed to monitor tire pressure, strain, and temperature to enhance overall safety of a vehicle [79]. Other examples of structural monitoring regarding to disaster response (earthquake, explosion, etc.) and continuous structure-health monitoring (ambient vibrations, wind, etc.) can be found in [22, 61, 60, 72].

Residential applications: Advances in hardware technologies and networking solutions are required for extensive WSN home applications (see [68]). An envisioned smart environment is proposed by Herring and Kaplan [52] wherein wireless sensor nodes can be integrated into the furniture and appliances, allowing them to communicate with one another so that they become self-organizing, self-regulated, and adaptive systems. WSN residential applications can also help elderly residents by providing memory enhancement, control of home appliances, medical data lookup, and emergency communication (see [101]).

1.3 CHALLENGES OF WSNS

WSNs have been used in several environmental, engineering and commercial applications, however they present a number of challenges that must be addressed before their full potential can be realized. In particular, network deployment, energy efficiency, routing, localization, data gathering and querying, time sensitivity, robustness, scalability and privacy and security are some of the primary challenges being addressed by researchers. Some of these challenges are next described in greater detail.

Network deployment: WSN deployment can be described as the problem of deciding how and where sensors should be placed to maintain connectivity and coverage. For example, in forest fire monitoring sensors should be deployed in a way that every point in the environment is within the range of at least one sensor and every sensor can communicate with every other sensor. Depending on the application-specific requirements, sensors might be deployed either randomly or in a pre-determined way, such as by hand or via autonomous robots [64]. In some applications, nodes are over-deployed for robustness against node failures and energy depletion, while in other applications, sensor nodes can be added or replaced incrementally when it is necessary. Network topology (star, grid, mesh or random topology) may vary depending on the application and it affects the network deployment.

Energy efficiency and extending lifetime: Sensing, processing, computation and communication all consume considerable energy, but communication consumes the lion's share of energy. The power source of a sensor node is typically a small battery with limited energy stores. In many cases, nodes are deployed in a hostile or remote environment; therefore, it is impractical to recharge or replace the battery. For this reason, energy consumption of the nodes is a critical limiting factor in the design of WSNs. WSN lifetime has been studied extensively in the open literature. Consequently, there are many diverse network lifetime definitions. The most common definition of WSN lifetime is the time until a certain fraction of nodes fail due to energy depletion. However, this definition is insufficient in many applications because it does not account for network connectivity and any quality-of-service (QoS) requirements. A WSN can be viewed as a communication graph, where sensor nodes acts as the vertices and an edge represents a communication path between any two nodes. To maintain connectivity, vertices of the graph should not be partitioned into more than one connected component. In some networks, one or more base stations serve as a gateway for collecting data from the sensor nodes. In such scenarios, connectivity is ensured when a base station can be reached from any node. A number of researchers integrate the connectivity and quality-of-service in the lifetime definition (see [34, 44]). To prolong the network lifetime, several schemes exploit node mobility, clustering, heterogeneity, such as using a *network backbone* powered by more capable nodes in particular areas of the network, which most importantly support node activity in terms of sensing, data processing and communication (see [23, 99]). Novel routing and channel access protocols can also extend network lifetime. For example, nodes commonly enter a low-energy-consumption mode (sleep mode) and schedule *active/sleep status* to conserve energy. WSNs must be designed with energy efficiency to ensure that they can remain operational without sensor or battery replacements.

Routing: Routing in WSNs is challenging for several reasons. First, building a global addressing scheme for large scale WSNs is not possible. Second, unlike the wired networks, in WSNs required data is routed to the query node (sink) from multiple sensors (sources). In harsh environments or when sensors are mobile the link between sensors may fail or the quality of data transmission may fluctuate [64]. Therefore, periodic data collection or gathering the information from multiple sensors may be required. Finally, to maintain the accuracy of the information and robustness, the data traffic might have redundancy in it, which must be addressed for energy efficiency (see [4]). Moreover, routing protocols may consider link quality, link distance, residual energy of sensors, location and mobility information for energy efficiency. Data transmission might

be performed via a multi-hop or single-hop path, depending on the application-specific requirements, existing link distance and available energy. Routing protocols might also consider clustering the nodes so that pre-determined sensors or devices (cluster heads) can do data gathering thereby maintain reduction of data and energy expenditure. A variety of routing techniques are reviewed in a cogent survey by Al-Karaki and Kamal [6]. Most protocols aim to minimize the energy expended by the network while satisfying quality-of-service guarantees.

Localization: Localization is a mechanism to form a map of the network through measurements. The main motivation behind the localization is that the spatial location of the sensors might be the data that is required, especially in applications such as warehousing. Besides, it might be used in routing algorithms that uses the positions of sensors to make packet forwarding decisions. In most cases, the location information is initially known by all nodes. However, when nodes are mobile, not all sensors are required to know their locations all the time. A fraction of nodes could be identified and assigned *a priori* locations or they might improve their location knowledge over time. The time scale depends on the application and it affects the energy efficiency and performance of the network. One of the common localization systems is the Global Positioning System (GPS) (see [20]). However, GPS may not always meet the operational requirements of WSNs, especially the limited energy reserves of sensors. Therefore, localization problems are challenging and attractive for researchers.

Data gathering and querying: Many WSN applications require data be sent to a base station, which is resource-rich in terms of its computational ability, storage capacity, and energy store. This causes non-uniform energy consumption patterns. For example, in habitat monitoring, nodes which are close to the path of a mobile base station (such as a roving van), forward information coming from all other nodes; therefore, they tend to fail earlier and/or become a bottleneck for network throughput. In other applications, queries retrieve information from sensing nodes. For instance, suppose a WSN is collecting and storing information from a particular region on individual nodes. Queries are sent to sensing nodes to retrieve this information. This type of information dissemination increases information availability in WSNs. Sensor nodes advertise the information to a certain number of nodes. Additionally, clustering and the use of mobile agents that randomly traverse the network to gather and deliver data have also been used for efficient data collection (see [57]).

Time sensitivity: In many applications, the sensed data must be delivered, or accessed, within certain time constraints. For example, in a forest fire monitoring system, fires must be detected immediately, or even predicted, and this information is used to prescribe immediate remedial or preventive action. Most of the research to date either ignores the time limitations or focuses on minimizing the time to deliver or reach information. Routing protocols may prioritize forwarded information according to its deadline or distance left to travel. Other protocols use feedback control to maintain an average delay for transmissions (see [101]). However, few researchers have explicitly addressed ensuring the on-time delivery of time-sensitive data.

Scalability: Because it is envisioned that WSNs will cover large areas, it is anticipated that they will be comprised of hundreds of thousands, or even millions, of sensor nodes. Large node density is also a design challenge. For industrial processes monitoring, the node density is around 300 sensor nodes in a $5 \times 5\text{m}^2$ region, while this number will be even higher when hundreds of sensors are embedded in eye glasses, clothing, shoes and human body (see [5, 64]). The new schemes and protocols must be able to provide such scalability by using localized communication and hierarchical architectures.

Robustness: Since it is usually impractical to physically reach sensor nodes, the only practical type of maintenance in WSNs is the updating of software over the wireless channel. Service should not be interrupted significantly. Moreover, human interaction in some of the areas, such as habitat monitoring, is undesirable; therefore, nodes should be able to reconfigure and adjust to changing environmental conditions. Self-configuring and fault tolerance are important as network performance should not be affected significantly due to individual node failure, which might result from unpredictable external events (see [14]).

Privacy and security: In many applications such as battlefield monitoring, data confidentiality is essential. Moreover, data integrity should be guaranteed to prevent unauthorized data in the network. In such cases, encryption, authentication and intrusion detection schemes maintain security and reliability (see [86]). However, these consume much of the already constrained energy sources.

1.4 RESEARCH OBJECTIVES

While there are a myriad of problems associated with WSN design, modeling and analysis, this dissertation will address three specific problems associated with time sensitivity of information gathering, extending network lifetime and maximizing coverage in WSNs. In particular, the focus of this dissertation is on the performance analysis and optimization of *query-based* WSNs followed by a study related to clustering of nodes for networks with mobile sensor nodes.

In many applications, information must be delivered within a certain time so that the data is up-to-date and, if required, appropriate actions can be taken. As a QoS requirement, the probability that a query fails to locate information prior to a certain deadline cannot exceed a predetermined threshold. Many of the studies to date have failed to explicitly consider QoS requirements. However, usually there is a tradeoff between energy conservation and QoS measures. In the literature there are very few analytical models that account for energy efficiency, QoS and scalability. A mathematical framework to evaluate the performance of large-scale WSNs considering QoS is critically needed.

The low cost and flexible monitoring capabilities of WSNs make them very attractive for a number of applications. However, due to the nature of WSNs and hazardous sensing environments, prolonging the network lifetime is essential for the performance of WSNs. Even when renewable energy sources are used to obtain additional power, the energy available is still limited [100]. Therefore, energy conservation is targeted at the network design level by efficiently designing communication schemes, since transmitting is the most energy-intense activity, and using sleep/active schedules for nodes or particular components of nodes such as transmission radio. Communication schemes include decisions regarding the transmission range and time-to-live counter. Energy expenditure is proportional to the transmission range, such that a longer transmission range increases energy consumption. However, the transmission range must be large enough to maintain application requirements. During operation, nodes will fail due to battery drain; therefore, node density will change and the current transmission range must be adjusted to ensure functionality of the WSN. Similarly, active/sleep schedules should also be updated as node failures and number of alive nodes in the network changes. Existing mechanisms rarely incorporate simultaneously setting operating parameters such as sleep schedules, transmission range and time-to-live counter to maximize network lifetime.

In mobile WSNs, sensor locations typically change after initial deployment due to an incidental side effect such as environmental influences (wind or water) or due to a desired property of the system. Putting GPS receivers in every node or manually configuring locations is not cost effective for most sensor network applications [54, 89]. Some localization techniques have been proposed to allow nodes to estimate their locations using information transmitted by a set of seed nodes that know their own locations as they have GPS receivers. These techniques suffer from many problems from the requirements of special hardware to requirement to a particular network topology. Therefore, mathematical models that focus on relocation of cluster heads under imperfect knowledge of sensor locations is required to improve the data coverage and reduce the cost by eliminating unnecessary traffic and cost caused by broadcasting sensor locations in every time interval.

The primary objectives of this doctoral dissertation are as follows:

1. To develop an analytical queueing-based framework for the performance evaluation of large-scale, query-based wireless sensor networks whose nodes detect and advertise significant events that are useful for only a limited time;
2. To use the analytical framework developed in Research Objective 1, to create and solve mathematical programming models to maximize WSN lifetime by dynamically choosing operating parameters (namely resource replication level, transmission range and active/sleep schedules) subject to QoS and connectivity constraints;
3. To create mathematical programming models to maximize the total sensor demand coverage by locating and relocating cluster heads and determining the optimal times at which to update sensor locations in wireless sensor networks with mobile sensor nodes.

The next subsection provides an outline of the dissertation and highlights the main contributions related to the performance analysis and optimization of wireless sensor networks.

1.5 DISSERTATION OUTLINE AND CONTRIBUTIONS

Chapter 2 presents two models for evaluating the performance of large-scale WSNs with time-sensitive events and queries using a queueing-theoretic approach. The first model leads to an approximation for the steady state proportion of query failures as a measure of QoS that is insensitive to the network's size, while the second model captures the realistic effects of a limited

transmission range and is asymptotically valid. Both models can accommodate generally distributed (non-exponential) event agent and query lifetimes. The numerical results indicate that the approximations perform very well (as compared to results obtained via a commercial network simulator), even when several of the key model assumptions are violated; the maximum absolute deviation in probability between the benchmark and approximated values is about 0.0319. The main results can be used for optimally designing and/or operating large-scale, query-based WSNs. Specifically, our models provide a proxy for energy expenditure (in the form of traffic rates) and proportion of query failures as a measure of QoS, and the approximations can be used to optimize other operating parameters including (but not limited to) the transmission range and/or the time-to-live counter so that a QoS constraint is satisfied. For instance, one might be interested in optimally selecting operating parameters to minimize energy expenditure while ensuring that the proportion of failed queries does not exceed a specified threshold. For this purpose, our procedures can be used to quickly evaluate and rank alternative operating policies without the need for costly and time-consuming simulation runs.

Chapter 3 explores the problem of network lifetime maximization in query-based WSNs while focusing on connectivity and QoS requirements. We first provide an approximation of the probability that the network is connected taking into account the border effects of a square region. Then, we develop nonlinear and linear mixed-integer programming models to maximize the network lifetime by choosing the optimal time-to-live counter, transmission range and active/sleep schedules. The limiting proportion of queries that fail to be answered on time is a critical QoS measure for query-based WSNs. The approximation of this measure presented in Chapter 2 is used in the mathematical models. In addition, a solution algorithm for solving a special case of the model in which all of the alive nodes remain in active mode is proposed. Computational results reveal that the network lifetime can be prolonged by selecting optimal parameters, and that the average network lifetime may actually decrease as the variance of energy expenditure increases. The need to minimize the energy consumption has motivated most of the research in WSNs so far. However, the simultaneous optimization of multiple operating parameters, while satisfying QoS and connectivity constraints, has not been explicitly addressed in WSN optimization research. The models of Chapter 3 are designed to maximize the network lifetime by reducing the energy consumption through determining the operating parameters. This model takes a major step forward towards improving WSN performance; however, it is clear that further research is needed to solve the proposed models for large-scale applications.

Chapter 4 proposes an optimization model to maximize the demand coverage and minimize the costs of locating, and relocating, cluster heads in a WSN with mobile nodes and unreliable links. There are similar models proposed for cluster head relocation in the literature; however, these models either fail to consider the randomness in the movements of mobile sensors or do not focus on the optimization. Instead, they typically propose relocation protocols, which are supported by simulation models. Our contribution here is to present a descriptive representation of the system, which leads to optimal relocation decisions. Additionally, we describe a method to transform the stochastic model into a deterministic version which can be solved using a commercial solver. Chapter 4 also uses numerical examples to illustrate the potential gains of making relocation decisions over time and updating information related to the sensors' locations when their movements are described by a stochastic process.

Finally, Chapter 5 summarizes the major findings of this dissertation and explores directions for future research.

2.0 PERFORMANCE ANALYSIS OF QUERY-BASED WSNS

This chapter presents a framework to evaluate the performance of large-scale, query-based WSNs whose sensors detect and advertise significant events that are useful for only a limited time (e.g., detecting hazardous biological agents, military surveillance, environmental monitoring, etc.). Event lifetimes are established to ensure that sensor nodes have the most up-to-date information to share with other nodes in the network. Query-based WSNs derive their name from the fact that communication between nodes is either event- or query-driven. That is, either the witnessing of an event (e.g., a sudden increase in temperature), or the generation of a query (e.g., a request for the temperature at a distant region of the network) triggers communication between nodes which act as routers for other nodes' packets due to a limited sensor transmission range. A query, which itself has a limited lifetime, traverses the network according to a two-dimensional random walk until it either locates the desired information or expires. For this type of network, the total proportion of generated queries that are not answered within their useful lifetime is a critical performance parameter. This chapter develops simple analytical approximations for this proportion, along with other network quality-of-service measures, within a queueing framework. The analytical approach is unique in that it explicitly accounts for the realism of limited event and query lifetimes which are generally distributed.

Wireless sensor networks have been analyzed from a variety of perspectives including design considerations, routing protocols, random wake-up schedules and resource management strategies, to name only a few. Some useful survey papers related to WSN sensing tasks, applications, design issues, and communications architectures include Akyildiz et al. [5], Yick et al. [110] and Dietrich and Dressler [34]. Owing to the fact that sensor nodes are energy-constrained, defining WSN lifetime and operating policies has emerged as a critical issue. Dietrich and Dressler [34] surveyed many definitions of WSN lifetime including the number of "alive" nodes, network coverage, network connectivity, and quality-of-service considerations (e.g., event detection rates). Other authors

(cf. Anastasi et al. [9]) have classified the various energy conservation approaches (e.g., sensor sleep/wake protocols, data acquisition schemes, mobile sink-based approaches, etc.). WSN lifetime and energy conservation strategies have further been discussed in [8, 25, 81, 83, 118, 94]. Routing in WSNs is challenging for at least the following reasons: (1) building a global addressing scheme for the deployment of nodes is not possible; (2) redundancies in the data since multiple nodes may generate the same data; (3) sensor nodes have limited transmission capabilities, limited on-board energy, and limited processing and storage capabilities.

Routing protocols constitute the largest area of research related to the performance of wireless sensor networks. Routing protocols are broadly labeled as flat, hierarchical (see [84]), and location-based. Flat (or data-centric) protocols assume all sensor nodes have equal capabilities and similar roles, whereas hierarchical protocols assign different roles to the nodes. Location-based protocols use sensor node position information to make routing decisions. The models analyzed in this dissertation fall into the category of flat routing and, more specifically, query-based flat routing. Classical data-centric approaches to locate and advertise data include flooding and gossiping (see [49]) which are known to be energy- and bandwidth-inefficient. Alternatively, rumor-routing protocols (see [4, 19, 13, 39, 85, 95, 108]) can be used. Rumor routing uses packets with relatively long lifetimes called *agents*. When a node detects an event, it adds information pertaining to the event in a local *event table* and immediately creates a time-limited agent that "advertises" the local information to distant nodes via subsequent packet transmissions. Consequently, a node of the network generates a query, any node with the information stored in its local event table can respond, when the query is received. This approach obviates the need for flooding, thereby reducing energy expenditure. Rumor routing is effective (relative to flooding) when the arrival rate of events is relatively low but generally requires significant overhead. Specifically, witnessed events are assigned a time-to-live (TTL) counter, or *resource replication level*, that is tracked while query lifetimes (which are limited) are also be tracked.

The *resource replication level* or *time-to-live (TTL) counter* (a hop counter) is the number of times a witnessed event is replicated in the network, and studies related to this parameter are relatively sparse. Bellavista et al. [15] developed a simulation model (REDMAN) to explore resource replication levels and related network settings. Krishnamachari and Ahn [65] derived cost expressions as a function of the resource replication level for unstructured networks in which the source node is unknown. They used expanding ring queries to search for the information and formulated a nonlinear programming (NLP) model to determine the optimal number of resource

replicates, subject to a network storage capacity constraint. Ahn and Krishnamachari [2] extended the results of [65] to structured networks in $d-$dimensional space, and studied structured and unstructured two-dimensional grid and random topology networks. The authors also presented a model to obtain the optimal resource replication level that minimizes the total expected cost of replication and searching, subject to a storage capacity constraint. An algorithm for dissemination and retrieval of information that ensures an even geographical distribution of the informed nodes is proposed for unstructured wireless ad-hoc networks by Miranda et al. [78]. Antoniou et al. [11] presented a nature-inspired data flow model for WSNs that considers congestion regions and dead zones (regions with failing nodes) in a sensor field. Most relevant to our work here, Mann et al. [76] used a queueing framework to obtain the optimal replication level that minimizes a proxy for energy expenditure, subject to a performance guarantee on the steady state proportion of failed queries. Their approach is unique in that it considers time-limited event agents and queries but is limited to memoryless (exponentially-distributed) lifetimes. Bisnik and Abouzeid [18] used a queueing network model to analyze random access, multi-hop wireless networks and derived the average end-to-end delay. Niyato and Hossain [80] developed a queueing model to investigate the performance of different sleep and wake-up strategies. Chiasserini et al. [27] proposed a fluid queueing model that accounts for energy consumption, active/sleep dynamics, and traffic routing. Jiang et al. [56] proposed a queueing-theoretic, power-saving scheme to address non-uniform node power consumption patterns. Ata [12] considered the problem of dynamically choosing the transmission rate in a general wireless communications network such that the average energy consumption per time unit is minimized, subject to a quality-of-service constraint. In that work, the transmission queue was modeled as a finite-buffer, $M/M/1$ system. With the exception of Mann et al. [76], none of the analytical models described herein account explicitly for limited event agent and query lifetimes.

Most specifically, we present a queueing-theoretic framework for evaluating the steady state proportion of query failures (i.e., the limiting proportion of generated queries that fail to be answered on time) in a large-scale WSN with time-critical data. While the network model itself is similar to the one described in [76], it has several important distinguishing attributes. Specifically, Mann et al. [76] consider only exponentially-distributed event agent and query lifetimes, whereas our model allows both types of lifetimes to be generally distributed. Second, the model of [76] is only an infinite-range (single-hop) model that ignores network topology and the limitations of a finite transmission range. Our approach explicitly models the dynamics of query movement over

time using a temporally-nonhomogeneous stochastic model that depends explicitly on the transmission range. We derive analytical approximations that explicitly account for (1) time-limited event agents and queries, (2) the limited transmission range of sensor nodes, and (3) generally-distributed resource and query lifetimes. The first approximation, derived using a single-hop model, is shown to be insensitive to the network's size. The second approximation, derived from a finite-range (or multi-hop) model, is shown to be asymptotically valid, and extensive numerical comparisons with simulated networks verify the exceptional accuracy of the approximations, even when key model assumptions are violated. It is well known that energy efficiency is a critical issue for WSNs; however, there exists a delicate tradeoff between satisfying quality-of-service guarantees and minimizing energy consumption (or maximizing the network's lifetime). Our proposed framework provides easy-to-implement approximations that can be used to devise optimal design or operating strategies for WSNs (e.g., optimizing the transmission range and/or TTL counter) to minimize energy expenditure or maximize network lifetime while limiting the proportion of failed queries to a fixed threshold.

The remainder of this chapter is organized as follows. Section 2.1 provides a description of the network model, queueing models of sensor node elements, and the most relevant attributes. In Section 2.2, we derive the steady state proportion of query failures assuming an unlimited sensor transmission range, while Section 2.3 presents an approximation that explicitly accounts for the limited transmission range of sensors. Section 2.4 presents extensive numerical results that validate the analytical approximations.

2.1 MODEL DESCRIPTION

Consider a multi-hop wireless sensor network (WSN) represented by an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N} = \{1, 2, \ldots, N\}$ is the node set (or set of vertices), N is the number of sensor nodes in the network, and $\mathcal{A}$ is the arc set of the sensor network. An arc (i, j) is an element of $\mathcal{A}$ if and only if nodes i and j are within transmission range of each other. Once deployed, the sensor nodes are spatially stationary (i.e., they are not mobile). In this research, we consider only networks with sensor nodes deployed in a rectangular sensor field R, a subset of Euclidean 2-space. The nodes are assumed to be spatially randomly distributed in R, i.e., the node locations are uniformly distributed. The node density of the network, ψ, is the average number of nodes

per unit area (in nodes/m^2) given by $\psi = N/L$ where L is the area of sensor field R. For each $i \in \mathcal{N}$, denote by $\boldsymbol{x}_i$ the position of sensor node i in Euclidean 2-space. Then for $j \in \mathcal{N}$, $j \neq i$, the Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is $\rho(i,j) \equiv \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ where $\|\cdot\|$ denotes the Euclidean norm. Assuming each sensor node has a transmission range r (in meters), the *degree* of node $i \in \mathcal{N}$ is the number of nodes within transmission range of i given by

$$d_i(r) \equiv \sum_{j \in \mathcal{N} \setminus \{i\}} \mathbf{1}(\rho(i,j) \leq r),$$

where $\mathbf{1}(x)$ is an indicator function that assumes the value 1 if condition x holds and 0 otherwise. Obviously, $d_i(r)$ depends on the deployment of nodes in R, the network topology, and the transmission range of individual sensor nodes. Finally, the average degree of the network is

$$\bar{d}(r) \equiv \frac{1}{N} \sum_{i=1}^{N} d_i(r).$$

A node $i \in \mathcal{N}$ for which $d_i(r) = 0$ is said to be *isolated*. Isolated nodes are essentially useless to the WSN since they cannot exchange information with other nodes. The WSN is said to be *disconnected* if there is a non-empty subset of isolated nodes in $\mathcal{N}$ but is *completely connected* if there is at least one path between nodes i and j for every $i, j \in \mathcal{N}$. Obviously, it is undesirable for the network to be disconnected, particularly when the information relayed by nodes is time sensitive. When the nodes are uniformly distributed in R with homogeneous node density ψ, the minimum transmission range needed to ensure the network is completely connected with probability p is (see Theorem 1 of [16])

$$\widehat{r} \geq \sqrt{\frac{-\ln\left(1 - p^{1/N}\right)}{\pi\, \psi}}. \tag{2.1}$$

The lower bound in (2.1) can be used, for example, to create discrete-event simulation models of wireless sensor networks that ensure connectivity with high probability.

Next, we describe individual sensor nodes in greater detail. (This discussion is similar to that of Mann [76].) It is assumed that sensor nodes are identical, i.e., they have identical resource requirements, physical limitations, and performance limitations. They are also similar with respect to their information requirements and the rates at which they observe and report relevant phenomena. Each sensor node is equipped with processing, transmitting, and sensing capabilities, as well as a limited power supply in the form of an on-board battery that cannot be recharged and is generally difficult, if not impossible, to replace.

In query-based WSNs, sensor nodes serve as both producers and consumers of network resources, and the transmission of data is triggered when an event occurs or a query is generated. A node

produces a *resource* when (1) it monitors the environment and gathers data on the occurrence of pertinent events; or (2) it offers a particular service to the network. In addition to data gathering, nodes are also required to execute specific applications in support of the network's goals. When a node requires access to a resource that is not available locally, the node is forced to traverse the network to locate the necessary information and/or services. Next, we describe node activities triggered by the occurrence of an event or a request for information.

When a node witnesses a relevant phenomenon or offers a particular service to the network, it broadcasts this information to a subset of the network by means of an *event agent* – a packet that describes the resource available, the location of the resource (or, alternatively, the data itself), and the duration of time the resource is available or valid. In this chapter, we assume that agents are transmitted from node-to-node via a random walk until either the witnessed event expires (i.e., it reaches its deadline), or it exhausts its *time-to-live* (TTL) counter – an integer hop counter representing the maximum number of times the resource may be replicated in the network. It is worth mentioning that a variety of routing protocols can be assumed (cf. [75]), but the results herein assume transmission to a randomly-selected neighbor. This type of random-walk-based routing protocol is useful for maintaining load balancing in a statistical sense (see [6]). Additionally, it is simple to implement, requires nodes to store very little state information, and is a pragmatic choice for large-scale networks with limited node mobility. Each sensor node is equipped with an on-board *event table*. Whenever an event agent is received, or an event is witnessed by the node, the contents of the event agent are added to the event table, and the node is labeled as *informed*, as long as the event agent's lifetime has not expired. On the other hand, if a node's event table does not contain the information witnessed or delivered by an event agent, then the node is said to be *uninformed*.

In addition to witnessing and forwarding events, nodes generate *queries* to request data or resources from the network. A query contains at least three pieces of information: the identifier and/or location of the node originating the request, the resource sought, and the maximum amount of time the query is permitted to traverse the network in search of an informed node. Only informed nodes are capable of answering the queries of uninformed nodes. Similar to event agents, queries are forwarded from node-to-node via a random walk. If a query is received by an informed node, the query is terminated and the informed node generates a *response* that is returned to the query origin node via the shortest path (least number of hops). We assume responses follow the shortest path because, whatever protocol is used to determine the response route, the best currently available

route will be discovered first since those routing packets will reach the route-requesting node first. The query response packet contains the information stored in the informed node's event table and, if available, the desired data. If a query cannot locate an informed node within its lifetime, the query fails. It is worth noting that we assume there are known data elements, and each query requests a particular data element; so there is a one-to-one correspondence between a query and a satisfying data element. Moreover, while it is conceivable that nodes receive redundant queries and/or event agents from multiple sources, we assume the receiving node neither aggregates nor generalizes the data in any way. Finally, it is assumed that all transmitted data are accurate, and there are no packet collisions.

Our main objective is to assess a critical quality-of-service measure for query-based WSNs, namely the long-run proportion of queries that fail to be answered on time. To this end, we create a queueing network model that leads to simple analytical expressions and accommodates easy computational implementation.

2.1.1 Queueing Models of Node Elements

For each $i \in \mathcal{N}$, events are assumed to arrive according to a Poisson process with rate λ. Each witnessed event is time sensitive, i.e., it is useful for only a limited time before it expires. Therefore, once an event is witnessed by a node, it is added to the node's event table and assigned a *lifetime*, Z, a non-negative, non-defective random variable. Event lifetimes (across all nodes) are independent and identically distributed (i.i.d.) random variables with common cumulative distribution function (c.d.f.) $G(w) \equiv \mathbb{P}(Z \leq w)$, $w \geq 0$, and mean $\mathbb{E}(Z) = 1/\delta < \infty$. As long as the event agent has not expired in the event table, the node is informed and can answer queries arriving from other nodes.

Because event agents are mutually independent, and do not necessarily expire in their order of arrival, the event table can be modeled as an $M/G/\infty$ queueing system whose input is a Poisson process with aggregate arrival rate Λ and whose service time is generally distributed with c.d.f. G (see Figure 3).

The event arrival rate Λ depends on many factors, not the least of which is the transmission range r. We pause here to remark that, in general, the superposition of locally-witnessed events and externally-generated event advertisements does not necessarily form a Poisson process since the latter do not (in general) originate from a Poisson stream. Furthermore, the event table may not realistically have infinite capacity. Therefore, the evolution of the number of busy servers in

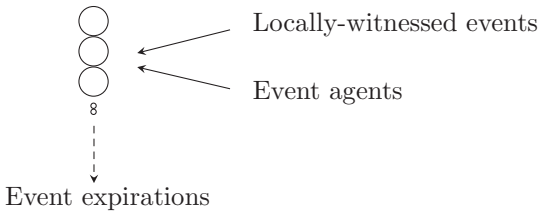


Figure 3: Graphical depiction of a sensor node's event table as an $M/G/\infty$ queue.

the $M/G/\infty$ model must be viewed as an approximation of the evolution of event table content. However, it will be shown in Section 2.4 that this assumption is not overly restrictive and that the proportion of query failures is surprisingly insensitive to the Poisson assumption. We choose the $M/G/\infty$ model for its tractability and generality with respect to event lifetimes. Specifically, it provides a simple expression for the steady state proportion of time an arbitrary node in the WSN is uninformed given by

$$\pi_0 \equiv \mathbb{P}(E = 0) = \exp\left(-\Lambda/\delta\right),$$

where E is the steady state number of events in the event table. Once a node witnesses an event, the information is forwarded until its TTL counter is exhausted. Henceforth, we denote the TTL counter by $\ell \in \mathbb{N}$.

Each sensor node contains a transmitter along with an (assumed) infinite buffer for storing data packets (queries, event agents, or responses). When a non-expired event agent arrives to a node, either because an event was witnessed, or because the agent is received from another node, the agent joins the transmission queue after a copy has been added to the node's event table. Moreover, when a node receives a query, either the query or the response is sent to the transmission queue, depending on whether the node is informed or uninformed. In either case, the query fails if the time elapsed from the moment of its inception until it locates an informed node exceeds its lifetime. While responses also join the transmission queue, this traffic stream is assumed to be negligible (relative to event agent and query traffic) since responses follow the shortest path and make far fewer hops than event agent agents or queries. Hence, we do not include response traffic in the total arrival rate calculation. The node's transmission queue is modeled as a single-server queueing system as depicted in Figure 4.

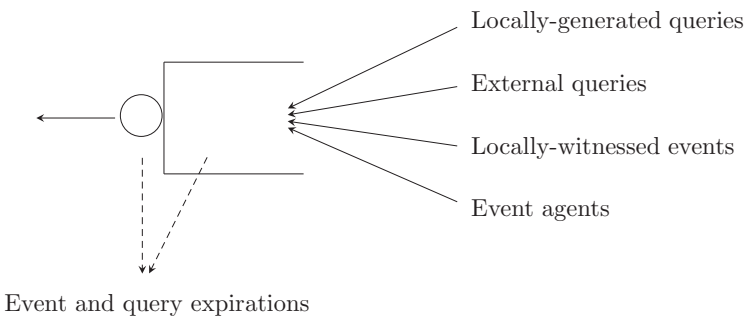


Figure 4: Graphical depiction of the sensor node's transmission queue.

Specifically, we assume that each node's transmission queue operates as a non-prioritized, multi-class $M/M/1$ queueing system with a first-come-first-served (FCFS) queueing discipline. Arrivals are assumed to originate from a Poisson process with rate λ_q. As depicted in Figure 4, the aggregate arrival process is comprised of locally-witnessed events, agents from other nodes, locally-generated queries, and queries arriving from other nodes. When a query is generated or received by a node, it joins the transmission queue only if the node is uninformed. The service time is the time needed to transmit a query or an event agent (either a locally-witnessed event or an advertisement from another node). Irrespective of the packet type, we assume the transmission time τ is an exponential random variable with parameter μ, c.d.f. $F(x) \equiv \mathbb{P}(\tau \leq x) = 1 - \exp(-\mu x)$, and finite mean $\mathbb{E}(\tau) = 1/\mu$. This assumption not only facilitates analytical tractability, but allows for larger variance in the transmission time. The transmission queue is stable if and only if $\mu > \lambda_q$. This condition is usually met in practice since transmission rates are generally very high. It is important to note that the total arrival rate of traffic to the transmission queue serves as a proxy for energy expenditure at a node since transmitting is the primary energy consuming activity (cf. [87]).

2.1.2 Network Performance Parameters

The primary concern of this chapter is the assessment of the steady state probability that a generated query fails to be answered on time. We refer to this performance parameter as the proportion of query failures. A query is said to fail if it expires awaiting transmission or while being transmitted. Our main aim is to provide easy-to-use analytical expressions for this parameter that allow us to circumvent costly, time-consuming simulation experiments for large-scale networks. To this end,

let T be a non-negative random variable denoting the total time needed for a query to locate an informed node as measured from the time the query is generated at a node $n \in \mathcal{N}$. This random time depends on the status of node n at the time of creation. Define the indicator variable

$$I_n = \begin{cases} 1, & \text{if node } n \text{ is informed,} \\ 0, & \text{if node } n \text{ is uninformed.} \end{cases}$$

The c.d.f. of $[T|I_n = 0]$ is denoted by $B(t) \equiv \mathbb{P}(T \leq t|I_n = 0)$, $t \geq 0$ for any $n \in \mathcal{N}$. At the time of generation, the query is assigned a lifetime, X, so that if T exceeds X, the query does not locate the desired information before expiring, and it fails. The c.d.f. of X is $H(x) \equiv \mathbb{P}(X \leq x)$, $x \geq 0$, and its mean is $\mathbb{E}(X) = 1/\beta < \infty$. Recall that π_0 is the proportion of time an arbitrary node is uninformed. Proposition 4.1 characterizes the primary performance parameter.

Proposition 2.1. *The unconditional proportion of query failures is*

$$\Delta \equiv \mathbb{P}(T > X) = \pi_0 \int_0^\infty [1 - B(x)] \, \mathrm{d}H(x). \tag{2.2}$$

Proof. Since (in steady state) a node is uninformed with probability π_0, we can use a conditioning argument to obtain

$$\begin{aligned} \Delta \equiv \mathbb{P}(T > X) &= \sum_{i=0}^{1} \int_0^\infty \mathbb{P}(T > X|X = x, I_n = i)\mathbb{P}(I_n = i)\mathrm{d}H(x) \\ &= \int_0^\infty \mathbb{P}(T > x|I_n = 0)\mathbb{P}(I_n = 0)\mathrm{d}H(x) \\ &= \pi_0 \int_0^\infty \mathbb{P}(T > x|I_n = 0)\mathrm{d}H(x) \\ &= \pi_0 \int_0^\infty [1 - B(x)] \, \mathrm{d}H(x). \end{aligned}$$

□

The expression for the proportion of query failures is straightforward except that the distribution function B is difficult to characterize in all but a few cases. That is, the time to locate an informed node is influenced by many factors including (but not limited to) the transmission range, availability of the requested data, the query's lifetime, and network traffic, all of which are interrelated. The next section considers the case when the sensors all use an infinite transmission range.

2.2 UNLIMITED SENSOR TRANSMISSION RANGE

In this section, we provide an approximation for Δ when $r = \infty$, i.e., when each node's transmission range is large enough to ensure that any other node in the network can be reached with a single hop. Although several assumptions are employed, the primary purpose of this model is to provide a framework for a more realistic limited transmission range model.

2.2.1 Approximating Network Traffic

Here we establish approximations for event agent and query arrival rates at an arbitrary node of the network. The first result bounds the aggregate event agent arrival rate to the sensor node's event table. This bound sets the stage for an approximation of the steady state proportion of time that any node is uninformed.

Proposition 2.2. *Assume events arrive locally to each $n \in \mathcal{N}$ according to a Poisson process with rate λ. Then the aggregate event arrival rate Λ to an arbitrary $n \in \mathcal{N}$ is bounded above by $\lambda(1+\ell)$ where ℓ is the time-to-live counter.*

Proof. The aggregate event arrival rate Λ consists of the Poisson rate of locally-witnessed events, and the aggregate rate of witnessed events arriving from the other $N-1$ nodes in the network. Therefore, $\Lambda = \lambda + \Lambda_x$ where Λ_x denotes the rate of external event arrivals. An event agent can be forwarded to, at most, ℓ nodes. Since $r = \infty$, each node can be reached with a single hop; hence, each event advertisement is equally likely to be received by one of the other $N-1$ nodes. That is, a particular node receives one of the (potential) ℓ advertisements with probability $\ell/(N-1)$, and since $N-1$ other nodes transmit event agents,

$$\Lambda_x \leq \lambda(N-1)\left(\frac{\ell}{N-1}\right) = \lambda\ell.$$

Therefore, $\Lambda \leq \lambda + \lambda\ell = \lambda(1+\ell)$. □

Next, we provide a lower bound for the steady state proportion of time a node is uninformed. An event agent is assigned a lifetime Z once it enters the event table. The random variable Z has c.d.f. G and finite mean $\mathbb{E}(Z) = 1/\delta$. As noted in Section 2.1, the event table is approximated by an $M/G/\infty$ queue with (Poisson) arrival rate Λ and service time distribution G, since the

limiting probability of an empty system depends on G only through its mean. Using the well-known steady state distribution of the $M/G/\infty$ system (see [66]), the limiting proportion of time a node is uninformed is

$$\pi_0 = \exp(-\Lambda/\delta), \quad 0 < \delta < \infty. \tag{2.3}$$

By Proposition 2.2, the event agent arrival rate to a node is bounded above by $\lambda\,(1+\ell)$. Therefore,

$$\pi_0 \geq \exp\left[\frac{-\lambda(1+\ell)}{\delta}\right]. \tag{2.4}$$

Similarly, the event agent arrival rate to the node's *transmission queue*, λ_e, is also bounded above. Although an event agent is transmitted at most ℓ times, the node which receives it at the ℓth transmission does not add the agent to its transmission queue since the agent's time-to-live counter will have expired. Therefore,

$$\lambda_e \leq \lambda + \lambda(N-1)\sum_{i=1}^{\ell-1}\frac{1}{N-1} = \lambda\ell. \tag{2.5}$$

While the bounds of (2.4) and (2.5) are valid, they may not be tight since they do not explicitly account for the expiration of event agents waiting in the transmission queue, or those that expire during transmission. Proposition 2.3 provides an improved bound for λ_e (by considering the effect of event expirations) that leads to an improved approximation for π_0. In what follows, let α_j denote the probability that an event agent expires at the jth visited node, $j = 0, 1, 2, \ldots$, where the 0th visited node is the event witnessing node. For simplicity, define the expiration probability at the witnessing node by $\alpha \equiv \alpha_0$. Assuming the event lifetime c.d.f. G has an increasing failure rate (IFR), then $0 < \alpha \leq \alpha_1 \leq \alpha_2 \leq \cdots$. This assumption asserts that event agents age over time, i.e., given that an event agent is alive at time t, the likelihood that it expires in $(t, t+a)$ for some $a > 0$ is increasing in t.

Proposition 2.3. *Suppose G is an IFR distribution function so that $0 < \alpha \leq \alpha_1 \leq \alpha_2 \leq \cdots$. Then for a fixed time-to-live counter ℓ,*

$$\lambda_e \leq \lambda\left[\frac{1-(1-\alpha)^\ell}{\alpha}\right] \leq \lambda\ell.$$

Proof. Since each of the $N-1$ nodes is equally likely to receive an advertised event agent, an individual node receives the kth transmission with probability

$$\frac{1}{N-1}\prod_{j=0}^{k-1}(1-\alpha_j), \quad k = 1, 2, \ldots, \ell-1.$$

That is, the event agent is forwarded at most ℓ times; however, the last node that receives the agent does not add it to its transmission queue since the agent's time-to-live counter will have expired. Therefore, the approximate total rate of event arrivals to a node's transmission queue is given by

$$\begin{aligned}
\lambda_e &= \lambda + \lambda(N-1) \cdot \sum_{k=1}^{\ell-1} \frac{1}{N-1} \prod_{j=0}^{k-1} (1-\alpha_j) \\
&\leq \lambda + \lambda \cdot \sum_{k=1}^{\ell-1} (1-\alpha)^k \\
&= \lambda \sum_{k=0}^{\ell-1} (1-\alpha)^k \\
&\leq \lambda \left[\frac{1-(1-\alpha)^\ell}{\alpha} \right] \leq \lambda \ell,
\end{aligned}$$

where the last inequality follows from $\alpha \in (0,1)$. □

For the results that follow, we use the approximation,

$$\lambda_e \approx \lambda \left[\frac{1-(1-\alpha)^\ell}{\alpha} \right],$$

to improve the approximation of π_0. Similarly, because the event agent is forwarded at most ℓ times, it can be shown that the event agent arrival rate to the event table is approximated by

$$\Lambda \approx \widehat{\Lambda} = \lambda \left[\frac{1-(1-\alpha)^{\ell+1}}{\alpha} \right]. \tag{2.6}$$

Therefore, by equation (2.3), when $r = \infty$, the approximate steady state proportion of time a node is uninformed is given by

$$\pi_0 \approx \exp\left[-\frac{\lambda}{\delta} \left(\frac{1-(1-\alpha)^{\ell+1}}{\alpha} \right) \right]. \tag{2.7}$$

Approximations (2.6) and (2.7) essentially ignore the probability that an event agent revisits a node because, in the single-hop model with N large and ℓ moderately small, the likelihood of revisiting any node is negligible.

Next, we examine the total traffic experienced at the transmission queue. Let λ_q be the total arrival rate of event agents and queries to a sensor node's transmission queue. Each node generates local queries according to a Poisson process with rate γ. When a query is generated locally, or received from another node, it is added to the transmission queue only if the subject node is uninformed. The arrival rate of locally-generated queries to the transmission queue, q_l, is $q_l = \pi_0 \gamma$. Let q_x denote the rate at which external queries arrive at a node. In steady state, the query visits

an informed node with probability $1-\pi_0$ and an uninformed node with probability π_0, independent of the number of hops prior to the current visit. Consequently, the number of hops needed to first locate an informed node follows a geometric distribution with success probability $1-\pi_0$ and mean $1/(1-\pi_0)$. Since any node is equally likely to receive a query, the probability of receiving an external query is $1/[(1-\pi_0)(N-1)]$. Because all other $N-1$ nodes generate queries identically, q_x is approximately

$$q_x = \gamma\,\pi_0\,(N-1)\,\frac{1}{(1-\pi_0)(N-1)} = \frac{\gamma\pi_0}{1-\pi_0}. \tag{2.8}$$

Finally, we approximate the total arrival rate of traffic to a node's transmission queue by

$$\lambda_q \approx \widehat{\lambda}_e + q_l + q_x = \lambda\left[\frac{1-(1-\alpha)^\ell}{\alpha}\right] + \pi_0\gamma\left(\frac{2-\pi_0}{1-\pi_0}\right). \tag{2.9}$$

By (2.7) and (2.9), we see that π_0 and, consequently, λ_q are explicit functions of α. Therefore, the approximation of λ_q is written as

$$\lambda_q \approx c(\alpha) \equiv \lambda\left[\frac{1-(1-\alpha)^\ell}{\alpha}\right] + \gamma e^{-g(\alpha)}\left[\frac{2-e^{-g(\alpha)}}{1-e^{-g(\alpha)}}\right], \tag{2.10}$$

where

$$g(\alpha) = \frac{\lambda}{\delta}\left[\frac{1-(1-\alpha)^{\ell+1}}{\alpha}\right].$$

We are now prepared to provide an expression for α, the probability that an event agent expires in the first transmission queue. The result is approximate since the input to the transmission queue is assumed to be the superposition of independent Poisson arrival streams. The equilibrium random variable Z_e associated to the lifetime Z with c.d.f. G and mean $\mathbb{E}(Z)$ has c.d.f.

$$G_e(z) \equiv \mathbb{P}(Z_e \le z) = \frac{1}{\mathbb{E}(Z)}\int_0^z [1-G(u)]\mathrm{d}u.$$

We make use of the equilibrium distribution of the event agent lifetime in the following proposition that characterizes α.

Proposition 2.4. *Assume $\mu > c(\alpha)$ for each λ and δ such that $0 < \lambda < \infty$ and $0 < \delta < \infty$. Let W be the total time spent at a node's transmission queue (delay plus transmission time) by an arbitrary arrival in steady state. Then α satisfies the fixed point equation*

$$\alpha = \mathbb{P}(W > Z_e) = \widetilde{G}_e(\mu - c(\alpha)) = \mathbb{E}\left[e^{-[\mu-c(\alpha)]Z_e}\right], \tag{2.11}$$

where $\widetilde{G}_e(\mu - c(\alpha))$ denotes the Laplace-Stieltjes transform (LST) of G_e evaluated at $\mu - c(\alpha)$.

Proof. The transmission queue is modeled as an $M/M/1$ queueing system with mean transmission time $1/\mu$ and aggregate arrival rate $c(\alpha)$. Let W_n be the total time spent in the transmission queue (i.e., the delay time plus the transmission time) by the nth arrival to the queue, either an event agent or a query. It is well known (see [43]) that if $\mu > c(\alpha)$, then $W_n \Rightarrow W$ as $n \to \infty$ where W is exponentially distributed with mean $1/(\mu - c(\alpha))$ and $(\Rightarrow)$ is convergence in distribution (or weak convergence). Suppose an event agent arrives at time t so that $Z - t$ is the residual lifetime of the event. Using basic results from renewal theory (cf. [66]), $Z - t \Rightarrow Z_e$ as $t \to \infty$. Therefore, by conditioning on Z_e, we obtain

$$\alpha = \mathbb{P}(W > Z_e) = \int_0^\infty e^{-(\mu - c(\alpha))z} \mathrm{d}G_e(z) = \widetilde{G}_e(\mu - c(\alpha)) = \mathbb{E}\left[e^{-(\mu - c(\alpha))Z_e}\right].$$

□

To illustrate (2.11), suppose the event lifetime is exponentially distributed with mean $1/\delta$. Then, $G_e(z) = G(z) = 1 - \exp(-\delta z)$ for all $z \geq 0$, and the unique probability α solves the fixed point problem

$$\alpha = \frac{\delta}{\mu - c(\alpha) + \delta},$$

where $c(\alpha)$ is given by (2.10). For an arbitrary equilibrium distribution G_e, we need to solve (2.11) numerically to obtain α. As seen by (2.7) and (2.9), the approximations of π_0 and λ_q depend explicitly on α. Therefore, we use the following fixed point iteration algorithm, which is standard in most numerical analysis textbooks (cf. [21]), to solve for α. Let $\pi_0^{(k)}$, $\lambda_q^{(k)}$ and $\alpha^{(k)}$ be the approximated values of π_0, λ_q and α at the kth iteration of the algorithm, respectively. The algorithm first obtains an initial guess of α using bounds (2.4) and (2.5). Each subsequent iteration uses approximations (2.7) and (2.9) to update these values until a convergence criterion is satisfied.

Algorithm to Compute α:

Step 0: *Initialization via the bounds of* (2.4) *and* (2.5).

$k := 0$;

$\pi_0^{(k)} := \exp\left[-\lambda(1+\ell)/\delta\right]$;

$\lambda_q^{(k)} := \lambda\,\ell + \gamma\pi_0^{(k)}\left(\dfrac{2 - \pi_0^{(k)}}{1 - \pi_0^{(k)}}\right)$;

$\alpha^{(k)} := \widetilde{G}_e\left(\mu - \lambda_q^{(k)}\right)$.

Step 1: *Update the approximations.*

$k := k + 1;$

$$\pi_0^{(k)} := \exp\left[-\frac{\lambda}{\delta}\left(\frac{1-\left(1-\alpha^{(k-1)}\right)^{\ell+1}}{\alpha^{(k-1)}}\right)\right];$$

$$\lambda_q^{(k)} := \lambda\left[\frac{1-\left(1-\alpha^{(k-1)}\right)^{\ell}}{\alpha^{(k-1)}}\right] + \gamma\pi_0^{(k)}\left[\frac{2-\pi_0^{(k)}}{1-\pi_0^{(k)}}\right];$$

$$\alpha^{(k)} := \widetilde{G}_e\left(\mu - \lambda_q^{(k)}\right).$$

Step 2:*Check convergence criterion.*

If $\left|\alpha^{(k)} - \alpha^{(k-1)}\right| > \epsilon$, return to Step 1;

Else $\alpha := \alpha^{(k)}$;

Stop.

Recall that our aim is to approximate Δ of (2.2) by assuming $r = \infty$. To this end, let $\widetilde{T}$ denote the time to locate an informed node when $r = \infty$ and let

$$\Delta_\infty \equiv \mathbb{P}(\widetilde{T} > X) = \pi_0 \int_0^\infty [1 - B(x)]\mathrm{d}H(x).$$

The c.d.f. of $\widetilde{T}$ is a function of both λ_q ($c(\alpha)$) and π_0, both of which are determined by α. The next section shows how to obtain Δ_∞.

2.2.2 Approximate Query Failure Rate

Queries, which can be generated at any node $n \in \mathcal{N}$, are forwarded via a random walk to one-hop neighbors until either an informed node is located, or the query expires while awaiting transmission in some node's transmission queue (or while being transmitted). Once generated, a query is assigned a lifetime X having c.d.f. $H(x) \equiv \mathbb{P}(X \leq x)$, $x \geq 0$. Let us assume for the moment that a query generated at an uninformed node can be forwarded indefinitely (i.e., $X = \infty$ w.p. 1), and let M be the (integer) number of hops needed to first locate an informed node.

Let T_k denote the time spent by a query at its kth location. That is, T_0 denotes the time spent at the query origin node (which is uninformed), T_1 is the time spent at the first visited node, which might be informed or uninformed, and so forth. To simplify notation, let $\widetilde{T}_u \equiv [\widetilde{T}|I_n = 0]$ be the

elapsed time between creation of the query at an *uninformed* node and the time it first locates an informed node. It is easy to see that

$$\widetilde{T}_u = \sum_{k=0}^{M-1} T_k.$$

Because we assume $r = \infty$ and identical nodes, in steady state, a query visits an informed node with probability $1-\pi_0$ and an uninformed node with probability π_0, independent of any prior visits. Thus, M is a geometric random variable with success probability $1-\pi_0$ and mean $1/(1-\pi_0)$, i.e., $\widetilde{T}_u$ is a geometric sum of i.i.d. exponential random variables.

Lemma 2.1. *Given that a query is generated at an uninformed node, the time to locate an informed node is exponentially distributed with parameter* $(1-\pi_0)(\mu-\lambda_q)$*, i.e.,*

$$B(x) \equiv \mathbb{P}(\widetilde{T}_u \leq x) = 1 - \exp\left[-(1-\pi_0)(\mu-\lambda_q)x\right], \quad x \geq 0, \tag{2.12}$$

where $\lambda_q \equiv c(\alpha)$ *is obtained using the value of* α *that solves the fixed point equation* (2.11)*.*

Using Lemma 2.1, we next provide our approximate expression for the steady state proportion of query failures when $r = \infty$.

Proposition 2.5. *Assuming Poisson event arrivals and query generation, the proportion of query failures in an infinite-range WSN is*

$$\Delta_\infty = \mathbb{P}(\widetilde{T} > X) = \pi_0 \, \widetilde{H}[(1-\pi_0)(\mu-\lambda_q)], \tag{2.13}$$

where $\widetilde{H}(s) = \mathbb{E}\left(e^{-sX}\right)$ *is the LST of the query lifetime distribution function* H*.*

Proof. The proof follows directly by conditioning on the lifetime X and utilizing Lemma 2.1. Specifically,

$$\begin{aligned}
\Delta_\infty = \mathbb{P}(\widetilde{T} > X) &= \int_0^\infty \mathbb{P}(\widetilde{T} > X | I_n = 0, X = x)\mathbb{P}(I_n = 0)\mathrm{d}H(x) \\
&= \pi_0 \int_0^\infty e^{-(1-\pi_0)(\mu-\lambda_q)x}\mathrm{d}H(x) \\
&= \pi_0 \, \widetilde{H}\left[(1-\pi_0)(\mu-\lambda_q)\right].
\end{aligned}$$

□

Proposition 2.5 provides the steady state proportion of generated queries that fail to be answered on time, and it holds for all query lifetime distributions that possess a LST. However, if the distribution function H is heavy-tailed and does not possess an LST, the transform approximation

method (TAM) developed by Harris and Marchal [48], or its modification by Shortle et al. [96], can be used to approximate $\widetilde{H}$. It is noteworthy that (2.13) is insensitive to the size of the network N.

Scalability of the WSN is an important issue as realistic networks are envisioned to have thousands or even hundreds of thousands of sensor nodes. The infinite range approximations of this section are appealing due to their insensitivity to N. In this single-hop model, for large N, the likelihood that a given node is visited more than once by an event agent or query is negligible since a witnessing node forwards an event agent to, at most, ℓ *distinct* nodes. Similarly, queries are assumed to visit a distinct node at each hop, independently of all prior hops. However, to conserve energy, realistic sensor nodes use a limited transmission range, so the likelihood of revisiting neighbors when using a random-walk protocol can be significant, as highlighted by Rodero-Merino et al. [88]. Obviously, forwarding event agents to informed nodes, and/or repeatedly transmitting queries to uninformed nodes wastes precious energy stores, prolongs the time needed to locate informed nodes and, ultimately, increases the query failure rate. This revisiting effect is even more pronounced for nodes located near the borders of the deployment region, as these nodes generally have a smaller node degree. In the next section, we present an approximation scheme that assumes a limited transmission range and explicitly accounts for the revisiting and border effects.

2.3 LIMITED SENSOR TRANSMISSION RANGE

In this section, we present an approximation for the steady state proportion of query failures that explicitly accounts for the limited transmission range of wireless sensors (i.e., a multi-hop model). Specifically, we approximate Δ_r, the steady state proportion of query failures when the sensor nodes have transmission range of r ($r < \infty$). Additionally, we show that for large N, the approximation converges appropriately to Δ_∞ as $r \to \infty$.

2.3.1 Modeling Query Dynamics

Here we consider the status (and movement) of an individual query from its inception until it either locates an informed node or fails due to expiration. If a query is generated at an informed node, it is answered immediately and never forwarded; therefore, we focus on the case when a query is generated at an uninformed node. At its inception the query is instantaneously assigned a lifetime

X with c.d.f. $H(x)$ and mean $\mathbb{E}(X) = 1/\beta$ $(0 < \beta < \infty)$. It is forwarded to a randomly selected node within the r-radius of the current node until either an informed node is located or the query lifetime ends, in which case it is destroyed. In what follows, all random quantities are conditioned on the event $\{X = x\}$; therefore, we make the dependence on x explicit. Before proceeding with the formal model description, let us recall our node-labeling convention.

The query origin node is labeled as the 0th visited node, and if the query is successfully transmitted to an uninformed node next, it joins that node's transmission queue. This subsequent node is labeled as the *first* visited node at which the query awaits its second transmission, and so forth. More generally, a query awaits its kth transmission at the $(k-1)$st visited node. Now, for each integer k $(k \geq 0)$, let Q_k be the status of the query just before *potentially* joining the transmission queue of the kth visited node. That is, following the $(k-1)$st visit, the query might not join the next transmission queue because its lifetime may have ended, or it may have been answered at the $(k-1)$st visited node. The query only joins the kth node's transmission queue if the query is alive and unanswered after the $(k-1)$st visit. Therefore, the query can be in one of three mutually exclusive and exhaustive states: *active* (state 0), *answered* (state 1), or *expired* (state 2). For each $k \geq 0$, $Q_k \in S \equiv \{0, 1, 2\}$ where $Q_k = 0$ means the query, having been successfully transmitted k times, has not expired but has not been answered; $Q_k = 1$ means the query, having been successfully transmitted k times is answered at the kth visited node (i.e., the kth visited node is informed); and $Q_k = 2$ means the query was successfully transmitted $k-1$ times but expired awaiting its kth transmission (or during its kth transmission) at the $(k-1)$st visited node. (Note that $\mathbb{P}(Q_0 = 2) = 0$.) The process $Q \equiv \{Q_k : k \geq 0\}$ is an S-valued discrete-time Markov chain (DTMC) with *temporally-nonhomogeneous* one-step transition probability matrix, $\mathbf{P}(k, x)$, given by

$$\mathbf{P}(k,x) = \begin{bmatrix} p_{00}(k,x) & p_{01}(k,x) & p_{02}(k,x) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad k \geq 0, \;\; x \geq 0 \tag{2.14}$$

where for each $i, j \in S$,

$$p_{ij}(k,x) \equiv \mathbb{P}_x(Q_{k+1} = j | Q_k = i), \quad k \geq 0,$$

denotes the probability that the status of the query transitions from i to j at the $(k+1)$st step and $\mathbb{P}_x(A) \equiv \mathbb{P}(A|X = x)$ for any measurable event A. Once a query locates an informed node, it is no longer forwarded to a neighbor node, and if the query lifetime ends awaiting transmission (or during transmission), it is destroyed; therefore, states 1 and 2 are absorbing states of the DTMC. Row 0

of $\mathbf{P}(k,x)$ contains the critical transition probabilities. In particular, $p_{02}(k,x)$ is the probability that the query fails at the kth visited node, given it was active just before being added to the kth visited node's transmission queue. Likewise, $p_{00}(k,x)$ is the probability the query remains active just before being added to the $(k+1)$st visited node's transmission queue, given it was active just before being added to the kth node's transmission queue. Finally, $p_{01}(k,x)$ is the probability that a query is answered at the $(k+1)$st visited node, given it was active just before being added to the kth visited node's transmission queue.

Obviously, the DTMC Q is reducible with one transient state (state 0) and two closed communicating classes, namely $C_1 = \{1\}$ and $C_2 = \{2\}$; therefore, its limiting behavior is fairly easy to characterize. Before examining the limiting behavior, we characterize the distribution of $\{Q_k : k \geq 0\}$ at a particular step k. Let $v_j^k(x) = \mathbb{P}_x(Q_k = j)$ be the (unconditional) probability that the query is in state $j \in S$ just before joining the transmission queue of the kth node, and let $\boldsymbol{v}^k(x) = [v_j^k(x)]_{j\in S}$ be a (1×3) row vector comprised of these values. Because Q possesses a time-nonhomogeneous transition probability matrix, the vector $\boldsymbol{v}^k(x)$ can be obtained recursively (cf. [62]) by

$$\boldsymbol{v}^{k+1}(x) = \boldsymbol{v}^k(x)\,\mathbf{P}(k,x), \quad k \geq 0,$$

whose solution is given by

$$\boldsymbol{v}^{k+1}(x) = \boldsymbol{v}^0(x) \prod_{n=0}^{k} \mathbf{P}(n,x), \quad k \geq 0. \tag{2.15}$$

The square matrix on the right-hand side of (2.15) is the $(k+1)$-step transition probability matrix of Q. The transient analysis of Q facilitates an analysis of its limiting behavior which, in turn, is used to derive an expression for the steady state probability that a query fails to locate an informed node before its lifetime ends.

To this end, let us define the limiting probability vector

$$\boldsymbol{v}(x) \equiv \lim_{k\to\infty} \boldsymbol{v}^{k+1}(x) = \lim_{k\to\infty} \boldsymbol{v}^0(x) \prod_{n=0}^{k} \mathbf{P}(n,x) = \boldsymbol{v}^0(x) \lim_{k\to\infty} \prod_{n=0}^{k} \mathbf{P}(n,x). \tag{2.16}$$

Before approximating this vector, we first establish the existence and structure of the limit in the right-most term of (2.16) via Theorem 2.1.

Theorem 2.1. *For a fixed lifetime x $(x > 0)$, there are real numbers $\alpha_1(x)$ and $\alpha_2(x)$ such that*

$$\mathbf{A}(x) \equiv \lim_{k\to\infty} \prod_{n=0}^{k} \mathbf{P}(n,x) = \begin{bmatrix} 0 & \alpha_1(x) & \alpha_2(x) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $\alpha_1(x), \alpha_2(x) \in (0,1)$ *and* $\alpha_1(x) + \alpha_2(x) = 1$.

Proof. Using induction, it can be shown that the $(k+1)$-step transition probability matrix is given by

$$\prod_{n=0}^{k} \mathbf{P}(n,x) = \begin{bmatrix} \prod_{n=0}^{k} a_n & \sum_{n=0}^{k} b_n \left(\prod_{j=0}^{n-1} a_j \right) & \sum_{n=0}^{k} c_n \left(\prod_{j=0}^{n-1} a_j \right) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.17}$$

where $a_n \equiv p_{00}(n,x)$, $b_n \equiv p_{01}(n,x)$, $c_n \equiv p_{02}(n,x)$, $n \geq 0$, and $a_{-1} \equiv 1$. First, note that rows 1 and 2 of $\prod_{n=0}^{k} \mathbf{P}(n,x)$ are as given in (2.17) for any $k \in \mathbb{N}$; hence, we need only concern ourselves with row 0. Allowing $k \to \infty$ on both sides of (2.17), and noting that $0 < a_n < 1$, we see immediately that

$$\lim_{k\to\infty} \prod_{n=0}^{k} a_n = 0,$$

$$\alpha_1(x) \equiv \lim_{k\to\infty} \sum_{n=0}^{k} b_n \prod_{j=0}^{n-1} a_j = b_0 + \sum_{n=1}^{\infty} b_n \prod_{j=0}^{n-1} a_j \geq b_0 > 0, \tag{2.18}$$

and

$$\alpha_2(x) \equiv \lim_{k\to\infty} \sum_{n=0}^{k} c_n \prod_{j=0}^{n-1} a_j = c_0 + \sum_{n=1}^{\infty} c_n \prod_{j=0}^{n-1} a_j \geq c_0 > 0. \tag{2.19}$$

Since each row of $\mathbf{A}(x)$ is comprised of nonnegative real numbers, and the row sums must be unity (cf. [31]), we conclude that $\alpha_1(x) + \alpha_2(x) = 1$ which, in light of (2.18) and (2.19), implies that $0 < \alpha_1(x) < 1$ and $0 < \alpha_2(x) < 1$. □

For computational purposes, we approximate $\boldsymbol{v}(x)$ by truncating the infinite product of (2.16) at an appropriate integer q. Specifically, for a sufficiently large $q \in \mathbb{N}$, the approximation for $\boldsymbol{v}(x)$ is given by

$$\boldsymbol{v}(x) \approx \boldsymbol{v}^{q+1}(x) = \boldsymbol{v}^0(x) \prod_{n=0}^{q} \mathbf{P}(n,x), \tag{2.20}$$

where q is chosen such that $\|\boldsymbol{v}^{q+1}(x) - \boldsymbol{v}^q(x)\|_\infty < \epsilon$ with $\|\cdot\|_\infty$ the usual ∞-norm and ϵ a convergence threshold.

2.3.2 Approximate Query Failure Rate

Let Δ_r be the limiting probability of query failure provided each sensor's transmission range is r $(r < \infty)$ and let

$$v_2(x) \equiv \lim_{k\to\infty} v_2^k(x) = \lim_{k\to\infty} \mathbb{P}_x(Q_k = 2).$$

The unconditional proportion of query failures is approximately

$$\Delta_r = \int_0^\infty v_2(x)\mathrm{d}H(x). \tag{2.21}$$

Let $\pi_0(r)$ be the steady state proportion of time an arbitrary node is uninformed when the transmission range is r. Note that $v_2(x)$ depends implicitly on r through $\pi_0(r)$ since $\boldsymbol{v}^0(x) = (\pi_0(r),\, 1-\pi_0(r),\, 0)$, and $\mathbf{A}(x)$ depends on $\pi_0(r)$. However, we suppress this dependence on r for ease of notation. To compute $\boldsymbol{v}(x)$ (or its approximation $\boldsymbol{v}^q(x)$ via (2.20)), we now provide an expression for $p_{02}(k,x)$ and, subsequently, expressions for $p_{00}(k,x)$ and $p_{01}(k,x)$.

Lemma 2.2. *For a fixed lifetime x $(x>0)$, the transition probability $p_{02}(k,x)$ is*

$$p_{02}(k,x) = \frac{e^{-(\mu-\lambda_q)x}}{G(k,x)} \cdot \frac{[(\mu-\lambda_q)x]^k}{k!}, \quad k \geq 0, \tag{2.22}$$

where for each $k \geq 1$, $G(k,x)$ is the c.d.f. of a k-phase Erlang random variable with parameter $\mu-\lambda_q$ and $G(0,x) \equiv 1$.

Proof. If the query is transmitted to the kth node, then it had k successful prior transmissions without expiring. As before, let T_i denote the sojourn time at the ith visited node, $i \geq 0$. Because each node's transmission queue is modeled as a stable $M/M/1$ queue, $\{T_i : i \geq 0\}$ is an i.i.d. sequence of random variables with parameter $\mu-\lambda_q$. Denote by Y_k the total time elapsed from the moment a query is generated at an uninformed node up to and including its kth transmission, i.e.,

$$Y_k = \sum_{i=0}^{k-1} T_i,$$

where Y_k is a k-phase Erlang random variable with parameter $\mu-\lambda_q$. It is well-known (cf. [66]) that, for $k \geq 1$, the c.d.f. of Y_k is

$$G(k,x) \equiv \mathbb{P}(Y_k \leq x) = 1 - \sum_{n=0}^{k-1} e^{-(\mu-\lambda_q)x} \frac{[(\mu-\lambda_q)x]^n}{n!}.$$

We can express the conditional probability $p_{02}(k,x)$ in terms of the random variables Y_k and Y_{k+1} by noting that

$$p_{02}(k,x) = \mathbb{P}(Y_{k+1} > x | Y_k \leq x)$$

is the probability the query lifetime ends at the kth visited node while awaiting its $(k+1)$st transmission, given it had successfully made k prior transmissions and was active just before joining

the kth node's transmission queue. When $k = 0$, $p_{02}(0, x)$ is the probability the query lifetime ends in the transmission queue of the query origin node given by

$$p_{02}(0, x) = \mathbb{P}(Y_1 > X | X = x) = \mathbb{P}(T_0 > x) = e^{-(\mu - \lambda_q)x}.$$

For $k \geq 1$, using basic conditional probability,

$$p_{02}(k, x) = \mathbb{P}(Y_{k+1} > x | Y_k \leq x) = \frac{G(k, x) - G(k+1, x)}{G(k, x)} = \frac{e^{-(\mu - \lambda_q)x}}{G(k, x)} \frac{[(\mu - \lambda_q)x]^k}{k!}.$$

□

The remaining probabilities in row 0 of $\mathbf{P}(k, x)$, $p_{00}(k, x)$ and $p_{01}(k, x)$, depend on whether or not the query revisits uninformed nodes during its lifetime when $r < \infty$. For this reason, it is necessary to first compute the probability that the query visits a particular node $n \in \mathcal{N}$ for the first time at its kth visit.

To this end, let U_k be the location of the query just after its kth hop and note that $\{U_k : k \geq 0\}$ is a time-homogeneous DTMC with state space $\mathcal{N} = \{1, \ldots, N\}$. Define its one-step transition probability matrix by $\boldsymbol{\theta}(r) = [\theta_{ij}(r)]_{i,j \in \mathcal{N}}$. As in Section 2.1, for $j \neq i$, let $\rho(i, j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ and let $d_i(r)$ be the degree of node $i \in \mathcal{N}$. Assuming any neighbor of the current node is equally likely to receive a query transmission, for $i, j \in \mathcal{N}$ such that $j \neq i$, the transition probability $\theta_{ij}(r)$ is

$$\theta_{ij}(r) = \begin{cases} 1/d_i(r), & \text{if } \rho(i, j) \leq r, \\ 0, & \text{if } \rho(i, j) > r. \end{cases}$$

(Note that $\theta_{ii}(r) = 0$ for all $i \in \mathcal{N}$ as a query cannot be transmitted to the current node.)

Now, to account for revisiting effects, let $q(k, r)$ be the probability that a query (or event agent) visits a distinct (previously unvisited) node at the kth visit, and let $u_r(i, j, k)$ be the probability of visiting node j *at least once* before the $(k+1)$st visit, given that the query (or agent) originates at node i. Let $w_r(i, j, k)$ denote the probability the query visits state j for the first time on the kth visit, given it originated at node i. We have the following lemma.

Lemma 2.3. *For each $k \in \mathbb{N}$ and $r \in (0, \infty)$,*

$$q(k, r) \approx \widehat{q}(k, r) = \frac{1}{N} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N} \setminus \{i\}} [u_r(i, j, k) - u_r(i, j, k-1)] \tag{2.23}$$

where

$$u_r(i, j, k) = \begin{cases} \theta_{ij}(r) + \displaystyle\sum_{m \in \mathcal{N} \setminus \{j\}} \theta_{im}(r)\, u_r(m, j, k-1), & k \geq 1, \\ 0, & k = 0. \end{cases}$$

Proof. The lemma is proved using standard results for DTMCs. Specifically, define

$$T_{ij}^r = \inf\{k \geq 1 : U_k = j | U_0 = i\}$$

as the first passage time to node $j \in \mathcal{N}$, given that the query (or event agent) was generated at node $i \in \mathcal{N}$. Then,

$$u_r(i,j,k) = \mathbb{P}(T_{ij}^r \leq k),$$

and these probabilities can be obtained recursively by conditioning on the location of the query after its first transmission. The derivation is similar to that outlined in Theorem 4.1 of [66] and shows that for $k \geq 1$,

$$u_r(i,j,k) = \theta_{ij}(r) + \sum_{m \in \mathcal{N} \setminus \{j\}} \theta_{im}(r)\, u_r(m,j,k-1), \quad i,j \in \mathcal{N},$$

where $u_r(i,j,0) \equiv 0$ for each $i,j \in \mathcal{N}$. Using $u_r(i,j,k)$, the probability the query's first visit to node j is the kth visit, given the query originated at node i, is

$$w_r(i,j,k) \equiv \mathbb{P}(T_{ij}^r = k) = u_r(i,j,k) - u_r(i,j,k-1), \quad k \geq 1.$$

Assuming a query is generated at any $i \in \mathcal{N}$ with equal probability (i.e., $\mathbb{P}(U_0 = i) = 1/N$ for all $i \in \mathcal{N}$), via unconditioning, the approximate probability a query visits a distinct node at the kth visit is

$$q(k,r) \approx \widehat{q}(k,r) = \frac{1}{N} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N} \setminus \{i\}} w_r(i,j,k), \quad k \geq 1.$$

□

Lemma 2.3 facilitates simple approximations for the transition probabilities $p_{00}(k,x)$ and $p_{01}(k,x)$, $k \geq 0$, which are provided in the next proposition.

Proposition 2.6. *The transition probabilities $p_{00}(k,x)$ and $p_{01}(k,x)$, $k \geq 0$, are respectively approximated by*

$$p_{00}(k,x) \approx [1 - \widehat{q}(k+1,r)(1-\pi_0(r))]\,[1 - p_{02}(k,x)], \tag{2.24}$$

$$p_{01}(k,x) \approx \widehat{q}(k+1,r)[1-\pi_0(r)]\,[1 - p_{02}(k,x)]. \tag{2.25}$$

Proof. This approximation assumes that if node i is uninformed when a query first visits the node, it remains uninformed during any subsequent visits to node i by the same query. We justify this assumption by noting that the mean recurrence time to node i is proportional to r. To approximate

$p_{00}(k,x)$, condition on whether or not the $(k+1)$st visited node is distinct. First, given the query does not expire at the kth visited node, the $(k+1)$st visited node is not distinct with probability $1-\widehat{q}(k+1,r)$. In the second case, given the query does not expire at the kth visited node, the $(k+1)$st node is distinct with probability $\widehat{q}(k+1,r)$, and it is uninformed with probability $\pi_0(r)$. Therefore, the probability of locating an uninformed node at the $(k+1)$st visit, given the query was active just before joining the transmission queue of kth node is, for $k \geq 0$,

$$\begin{aligned} p_{00}(k,x) &\approx [1-\widehat{q}(k+1,r)][1-p_{02}(k,x)] + \widehat{q}(k+1,r)\pi_0(r)[1-p_{02}(k,x)] \\ &= [1-\widehat{q}(k+1,r)(1-\pi_0(r))]\,[1-p_{02}(k,x)]. \end{aligned}$$

To approximate $p_{01}(k,x)$, note that the query moves from state 0 (active) to state 1 (answered) if it was successfully transmitted from the kth visited node to a distinct node that is informed. Therefore, for $k \geq 0$,

$$p_{01}(k,x) \approx \widehat{q}(k+1,r)[1-\pi_0(r)][1-p_{02}(k,x)].$$

□

Using the approximation of $\mathbf{P}(k,x)$, we now provide improved approximations for the WSN traffic rates, the steady state proportion of time nodes are uninformed, and the steady state proportion of failed queries. It was shown in Section 2.2 that, if each sensor's range is such that all $N-1$ other nodes belong to its neighborhood, the total arrival rate of witnessed events (both local and external) to the node's event table is

$$\Lambda \approx \widehat{\Lambda} = \lambda \left[\frac{1-(1-\alpha)^{\ell+1}}{\alpha}\right].$$

The approximation $\widehat{\Lambda}$ does not account for the revisiting effects noted in this section. The following result uses $\widehat{q}(k,r)$ to correct for revisits and improve the approximate total arrival rate to the event table. To distinguish these values, let $\Lambda(r)$ be the total arrival rate of local and external events as a function of r. Then we can write

$$\begin{aligned} \Lambda(r) \approx \widehat{\Lambda}(r) &= \lambda + \lambda\,\bar{d}(r)\left(\frac{\widehat{q}(1,r)(1-\alpha)}{\bar{d}(r)} + \frac{\widehat{q}(2,r)(1-\alpha)^2}{\bar{d}(r)} + \cdots + \frac{\widehat{q}(\ell,r)(1-\alpha)^{\ell}}{\bar{d}(r)}\right) \\ &= \lambda\left[1+\sum_{i=1}^{\ell}\widehat{q}(i,r)(1-\alpha)^i\right], \end{aligned}$$

where $\bar{d}(r)$ is the network's average node degree. Using $\widehat{\Lambda}(r)$, the steady state proportion of time nodes are uninformed, $\pi_0(r)$, is

$$\pi_0(r) \approx \exp\left[-\frac{\lambda}{\delta}\left(1+\sum_{i=1}^{\ell}\widehat{q}(i,r)(1-\alpha)^i\right)\right]. \tag{2.26}$$

Equation (2.26) is used to compute the elements of $\mathbf{P}(k,x)$, namely $p_{00}(k,x)$ and $p_{01}(k,x)$ via (2.24) and (2.25), respectively. These lead to the limiting matrix $\mathbf{A}(x)$, from which we obtain the limiting probability $v_2(x)$ via (2.20). Finally, we obtain Δ_r via (2.21). The asymptotic validity of this approximation is discussed in the next subsection.

2.3.3 Asymptotic Validity of Approximation

In this subsection, we show that the finite transmission range approximation is asymptotically valid by proving that, for large N, the proportion of query failures converges to Δ_∞ as $r \to \infty$. To this end, we have the following important lemma.

Lemma 2.4. *For large N, as $r \to \infty$, $\widehat{q}(k,r) \to 1$ for each $k \in \mathbb{N}$.*

Proof. First note that

$$\lim_{r\to\infty} d_i(r) = \lim_{r\to\infty} \sum_{j\in\mathcal{N}\setminus\{i\}} \mathbf{1}(\rho(i,j) \le r) = N-1.$$

Therefore, for $i, j \in \mathcal{N}$ with $j \neq i$,

$$\theta_{ij}(r) = \frac{1}{d_i(r)} \to \frac{1}{N-1}$$

as $r \to \infty$. By induction on $k \in \mathbb{N}$, we now characterize the limiting behavior of $u_r(i,j,k)$ as $r \to \infty$. For $k = 1$, note that $u_r(i,j,1) = \theta_{ij}(r) \to 1/(N-1)$. For $k = 2$, it is easy to show that

$$\begin{aligned}
\lim_{r\to\infty} u_r(i,j,2) &= \lim_{r\to\infty} \left(\theta_{ij}(r) + \sum_{m\in\mathcal{N}\setminus\{j\}} \theta_{im}(r)\, u_r(m,j,1) \right) \\
&= \frac{1}{N-1} + \sum_{m\in\mathcal{N}\setminus\{i,j\}} \left(\frac{1}{N-1}\right)^2 \\
&= \frac{2}{N-1} + O(N^{-2}),
\end{aligned}$$

where $O(N^{-2}) \to 0$ as $N \to \infty$. For the inductive step, assume $u_r(i,j,n) \to n/(N-1) + O(N^{-2})$ for any $n \in \mathbb{N}$. With some simplification we obtain

$$\begin{aligned}
\lim_{r\to\infty} u_r(i,j,n+1) &= \lim_{r\to\infty} \left(\theta_{ij}(r) + \sum_{m\in\mathcal{N}\setminus\{j\}} \theta_{im}(r)\, u_r(m,j,n) \right) \\
&= \frac{1}{N-1} + \sum_{m\in\mathcal{N}\setminus\{i,j\}} \frac{1}{N-1} \left[\frac{n}{N-1} + O(N^{-2}) \right] = \frac{n+1}{N-1} + O(N^{-2}),
\end{aligned}$$

which completes the induction proof. Therefore, for each $k \in \mathbb{N}$ and $i, j \in \mathcal{N}$ with $j \neq i$,

$$\lim_{r\to\infty} w_r(i,j,k) \equiv \lim_{r\to\infty} [u_r(i,j,k) - u_r(i,j,k-1)] = \frac{1}{N-1},$$

and consequently,

$$\lim_{r\to\infty} \widehat{q}(k,r) = \lim_{r\to\infty} \frac{1}{N} \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}\setminus\{i\}} w_r(i,j,k) = \frac{1}{N} \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}\setminus\{i\}} \frac{1}{N-1} = 1.$$

□

Lemma 2.4 is used to prove Theorem 2.2 which asserts that, as $r \to \infty$, the approximate event arrival rate, proportion of time uninformed, and the proportion of query failures all converge appropriately to their respective infinite-range counterparts for large networks.

Theorem 2.2. *For large N, as $r \to \infty$, $\widehat{\Lambda}(r) \to \Lambda$, $\pi_0(r) \to \pi_0$, and $\Delta_r \to \Delta_\infty$.*

Proof. By Lemma 2.4, $\widehat{q}(k,r) \to 1$ for each $k \in \mathbb{N}$ as $r \to \infty$. Therefore,

$$\begin{aligned}\lim_{r\to\infty} \widehat{\Lambda}(r) = \lim_{r\to\infty} \left[\lambda + \lambda \sum_{i=1}^{\ell} \widehat{q}(i,r)(1-\alpha)^i\right] &= \lambda \lim_{r\to\infty} \sum_{i=0}^{\ell} \widehat{q}(i,r)(1-\alpha)^i \\ &= \lambda \left[\frac{1-(1-\alpha)^{\ell+1}}{\alpha}\right] = \Lambda.\end{aligned}$$

Consequently, by (2.26) we see that $\pi_0(r) \to \pi_0$ as $r \to \infty$. Next, recall that for $r < \infty$,

$$\Delta \approx \Delta_r = \int_0^\infty v_2(x) \mathrm{d}H(x)$$

where $v_2(x) = \lim_{k\to\infty} v_2^k(x)$. So as $r \to \infty$, we substitute $\boldsymbol{v}^0(x) = (\pi_0,\, 1-\pi_0,\, 0)$ in the expression

$$\boldsymbol{v}^k(x) = \boldsymbol{v}^0(x) \prod_{n=0}^{k-1} \mathbf{P}(n,x).$$

Using (2.14), (2.22), (2.24), and (2.25), we now show by induction on k that the elements of $\boldsymbol{v}^k(x)$ are

$$v_0^k(x) = \pi_0^{k+1} G(k,x), \tag{2.27}$$

$$v_1^k(x) = \sum_{n=1}^{k} [\pi_0^n (1-\pi_0) G(n,x)] + 1 - \pi_0, \tag{2.28}$$

$$v_2^k(x) = \pi_0 \left[1 - \pi_0^{k-1} G(k,x) - (1-\pi_0) \sum_{n=1}^{k-1} \pi_0^{n-1} G(n,x)\right]. \tag{2.29}$$

For $k = 1$, applying (2.15) with $\boldsymbol{v}^0(x) = (\pi_0,\, 1-\pi_0,\, 0)$, it is easy to see that

$$\begin{aligned} v_0^1(x) &= \pi_0^2 G(1,x), \\ v_1^1(x) &= \pi_0(1-\pi_0)G(1,x) + 1 - \pi_0, \\ v_2^1(x) &= \pi_0[1 - G(1,x)], \end{aligned}$$

where the summation in (2.29) is 0 when $k = 1$. Similarly, for $k = 2$,

$$\begin{aligned} v_0^2(x) &= \pi_0^3 G(2,x), \\ v_1^2(x) &= \pi_0^2(1-\pi_0)G(2,x) + \pi_0(1-\pi_0)G(1,x) + 1 - \pi_0, \\ v_2^2(x) &= \pi_0\left[1 - (1-\pi_0)G(1,x) - \pi_0 G(2,x)\right]; \end{aligned}$$

therefore, the result holds for $k = 1, 2$. For the inductive step, assume that (2.27)–(2.29) hold for an arbitrary $m \in \mathbb{N}$. Then, after some matrix algebra, we obtain

$$\begin{aligned} v_0^{m+1}(x) &= \pi_0^{m+2} G(m+1,x), \\ v_1^{m+1}(x) &= \sum_{n=1}^{m+1} \left[\pi_0^n(1-\pi_0)G(n,x)\right] + 1 - \pi_0, \\ v_2^{m+1}(x) &= \pi_0 \left[1 - \pi_0^m G(m+1,x) - (1-\pi_0)\sum_{n=1}^{m} \pi_0^{n-1} G(n,x)\right], \end{aligned}$$

and the induction proof is complete. Now, as $r \to \infty$,

$$\begin{aligned} v_2(x) = \lim_{k\to\infty} v_2^k(x) &= \lim_{k\to\infty} \left[\pi_0 \left(1 - \pi_0^{k-1} G(k,x) - (1-\pi_0)\sum_{n=1}^{k-1} \pi_0^{n-1} G(n,x)\right)\right] \\ &= \pi_0 \left[1 - (1-\pi_0)\sum_{n=1}^{\infty} \pi_0^{n-1} G(n,x)\right]. \end{aligned}$$

We obtain a closed-form expression for $v_2(x)$ via its Laplace-Stieltjes transform, $\tilde{v}_2(s)$, given by

$$\begin{aligned} \tilde{v}_2(s) \equiv \int_0^\infty e^{-sx} \mathrm{d}v_2(x) &= \pi_0 \left[1 - \frac{1-\pi_0}{\pi_0} \sum_{n=1}^{\infty} \left(\frac{\pi_0(\mu - \lambda_q)}{\mu - \lambda_q + s}\right)^n\right] \\ &= \pi_0 \left[1 - \frac{1-\pi_0}{\pi_0} \left(\sum_{n=0}^{\infty} \left(\frac{\pi_0(\mu - \lambda_q)}{\mu - \lambda_q + s}\right)^n - 1\right)\right] \\ &= \pi_0 \left[1 - \frac{(1-\pi_0)(\mu - \lambda_q)}{(1-\pi_0)(\mu - \lambda_q) + s}\right]. \end{aligned}$$

Now, $\tilde{v}_2(s)$ can be inverted analytically to obtain

$$v_2(x) = \mathcal{L}^{-1}\left\{\frac{\tilde{v}_2(s)}{s}\right\} = \pi_0\, e^{-(1-\pi_0)(\mu-\lambda_q)x},$$

where $\mathcal{L}^{-1}$ is the inverse Laplace transform operator. Finally, we obtain

$$\begin{aligned}\lim_{r\to\infty}\Delta_r &= \int_0^\infty \pi_0\, e^{-(1-\pi_0)(\mu-\lambda_q)x}\mathrm{d}H(x)\\ &= \int_0^\infty \mathbb{P}(\widetilde{T}>X|I_n=0,X=x)\pi_0\mathrm{d}H(x)\\ &= \mathbb{P}(\widetilde{T}>X)\\ &= \Delta_\infty.\end{aligned}$$

□

In this section, we have modeled query dynamics using a temporally-nonhomogeneous DTMC. The elements of the transition probability matrix (2.14) are provided by Lemma 2.2 and Proposition 2.6. We derived a new approximation for the proportion of query failures via (2.21) by examining the limiting behavior of the DTMC. This analysis explicitly accounts for the dependence of the network's performance on a limited transmission range and query revisiting by approximating the probability, $q(k,r)$, that a query visits a distinct node on its kth visit. This probability also captures the boundary effect – namely that nodes near the borders of the deployment region are likely to have fewer neighbors, and hence, an increased likelihood of transmitting to previously visited nodes. In Section 2.4, we illustrate and assess the quality of the finite- and infinite-range approximations by comparing the steady state proportion of time uninformed and proportion of query failures with results obtained by a commercial network simulator.

2.4 NUMERICAL EXAMPLES AND VALIDATION

The analytical approximations of Sections 2.2 and 2.3 provide a relatively easy way to evaluate the behavior of query-based wireless sensor networks. In this section, we assess the quality of these approximations by comparing them with simulated values obtained using the OPNET commercial network simulator. Presented herein are summary tables and figures for uniform-topology networks with a variety of distributional assumptions and sensor transmission ranges. For each experiment, the minimum transmission range was chosen to ensure a connected network with probability $p = 0.9999$ using (2.1). Results for 1000- and 5000-node networks first are provided before presenting an extensive validation study that examines impact of our model assumptions.

For each scenario, we compute the maximum absolute deviation (MAD) between the approximated value and its simulated counterpart over a finite set of TTL values, $L \equiv \{1, 2, \ldots, 30\}$. We choose this set because, for many typical wireless applications, a TTL counter between 3 and 25 is suitable. For each $\ell \in L$, let π_0^ℓ be the approximate steady state proportion of time nodes are uninformed, assuming $r = \infty$, which is obtained via (2.7), i.e.,

$$\pi_0^\ell = \exp\left[-\frac{\lambda}{\delta}\left(\frac{1-(1-\alpha)^{\ell+1}}{\alpha}\right)\right].$$

Similarly, let $\pi_0^\ell(r)$ be the same value, assuming $r < \infty$, obtained by (2.26). That is,

$$\pi_0^\ell(r) = \exp\left[-\frac{\lambda}{\delta}\left(1 + \sum_{i=1}^{\ell} \widehat{q}(i,r)(1-\alpha)^i\right)\right].$$

For both cases, the probability α is approximated using the fixed point algorithm described in Section 2.2. To express the dependence of Δ on the TTL value ℓ, let Δ_∞^ℓ and Δ_r^ℓ denote the steady proportion of query failures when $r = \infty$ and $r < \infty$, respectively. Using (2.13) and (2.21), respectively, we compute

$$\Delta_\infty^\ell \equiv \pi_0^\ell \int_0^\infty \exp\left[-(1-\pi_0^\ell)(\mu - \lambda_q)x\right] \mathrm{d}H(x)$$

and

$$\Delta_r^\ell \equiv \int_0^\infty v_2(x)\mathrm{d}H(x),$$

where $v_2(x)$ is obtained via (2.20). In cases where the integrals cannot be evaluated in closed form, we perform numerical integration via the trapezoidal rule. Finally, we define $\pi_0^s(\ell)$ as the simulated steady state proportion of time nodes are uninformed, and $\Delta^s(\ell)$ as the simulated steady state proportion of query failures when the TTL counter is $\ell \in L$.

The MAD between the true (simulated) values and their corresponding analytical approximations are therefore

$$D_\pi \equiv \max_{\ell \in L} \left|\pi_0^s(\ell) - \widehat{\pi}_0(\ell)\right|, \tag{2.30}$$

where $\widehat{\pi}_0(\ell) = \pi_0^\ell$ if $r = \infty$, and $\widehat{\pi}_0(\ell) = \pi_0^\ell(r)$ if $r < \infty$. Similarly, let

$$D_\Delta \equiv \max_{\ell \in L} \left|\Delta^s(\ell) - \widehat{\Delta}_0(\ell)\right|, \tag{2.31}$$

where $\widehat{\Delta}(\ell) = \Delta_\infty^\ell$ if $r = \infty$, and $\widehat{\Delta}(\ell) = \Delta_r^\ell$ if $r < \infty$. For Examples 1 and 2 that follow, a few parameter values were held constant; these values are summarized in Table 1. Moreover, we assumed event lifetimes are exponentially distributed with mean $1/\delta$ in these two cases, but this assumption is relaxed in Example 3.

Table 1: Summary of parameter values for OPNET simulation: Examples 1, 2 and 4.

Parameter	Parameter description	Value
μ	Transmitter's exponential transmission rate	5.000
λ	Poisson rate of locally-witnessed events (for all $n \in \mathcal{N}$)	0.005
γ	Poisson rate of locally-generated queries (for all $n \in \mathcal{N}$)	0.050
$1/\delta$	Mean event lifetime	10.000
$1/\beta$	Mean query lifetime (for all distributions)	5.000

The analytical approximations were coded in the C programming language and executed in Microsoft® Visual Studio® 2008 on a personal computer equipped with an Intel® Core™ 2 Duo CPU operating at 3.00GHz with 2.00 GB of RAM. The simulated values were obtained via a discrete-event simulation model created in the OPNET Modeler® Wireless Suite v. 15. Ten (10) independent replications were performed for each $\ell \in L$ to ensure a standard error less than 5×10^{-4}. The plotted simulated values represent the average of the 10 replications. The run length was 3720s, including a 120s warm-up period for each replication. The simulation experiments were conducted on a personal computer equipped with an Intel® Core™ i7 CPU operating at 2.67GHz with 2.00 GB of RAM.

Table 2: MAD in the proportion of time uninformed (D_π) when $N = 1000$.

Query lifetime	350m		500m		1000m		5000m	
	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$
Exponential(0.2)	0.0523	0.0045	0.0289	0.0065	0.0202	0.0116	0.0024	0.0060
Triangular(0.1, 5.0, 9.9)	0.0529	0.0059	0.0296	0.0055	0.0173	0.0088	0.0173	0.0131
Uniform(0.1, 9.9)	0.0507	0.0068	0.0285	0.0041	0.0171	0.0093	0.0039	0.0070
Rayleigh(5.645)	0.0511	0.0050	0.0298	0.0055	0.0176	0.0088	0.0043	0.0069
Weibull(3.0, 5.6)	0.0511	0.0061	0.0302	0.0061	0.0173	0.0088	0.0035	0.0072

Example 1: 1000-Node Network: Here, we present results for a 1000-node wireless sensor network with nodes distributed randomly in a 3335m $\times$ 3335m sensor field. The node density is $\psi \approx 9.00 \times 10^{-5}$ nodes per square meter. To ensure a connected network with probability 0.9999, the minimum required sensor transmission range is $r = 239$m. Therefore, we considered the following transmission ranges: 350m, 500m, 1000m, 5000m. Table 2 summarizes the MAD in the proportion of time uninformed using each transmission range. The column labeled "$r = \infty$" corresponds to the infinite transmission range approximation, and the column labeled "$r < \infty$" is the finite range approximation. Table 2 indicates an order of magnitude improvement in the MAD by using the finite-range approximation, especially when the actual transmission range in the simulation model is small (350m). Because queries are more likely to revisit neighbors when the transmission range is small, the difference between the two approximations is quite pronounced. Table 2 also illustrates consistency in the performance of the approximations when the query

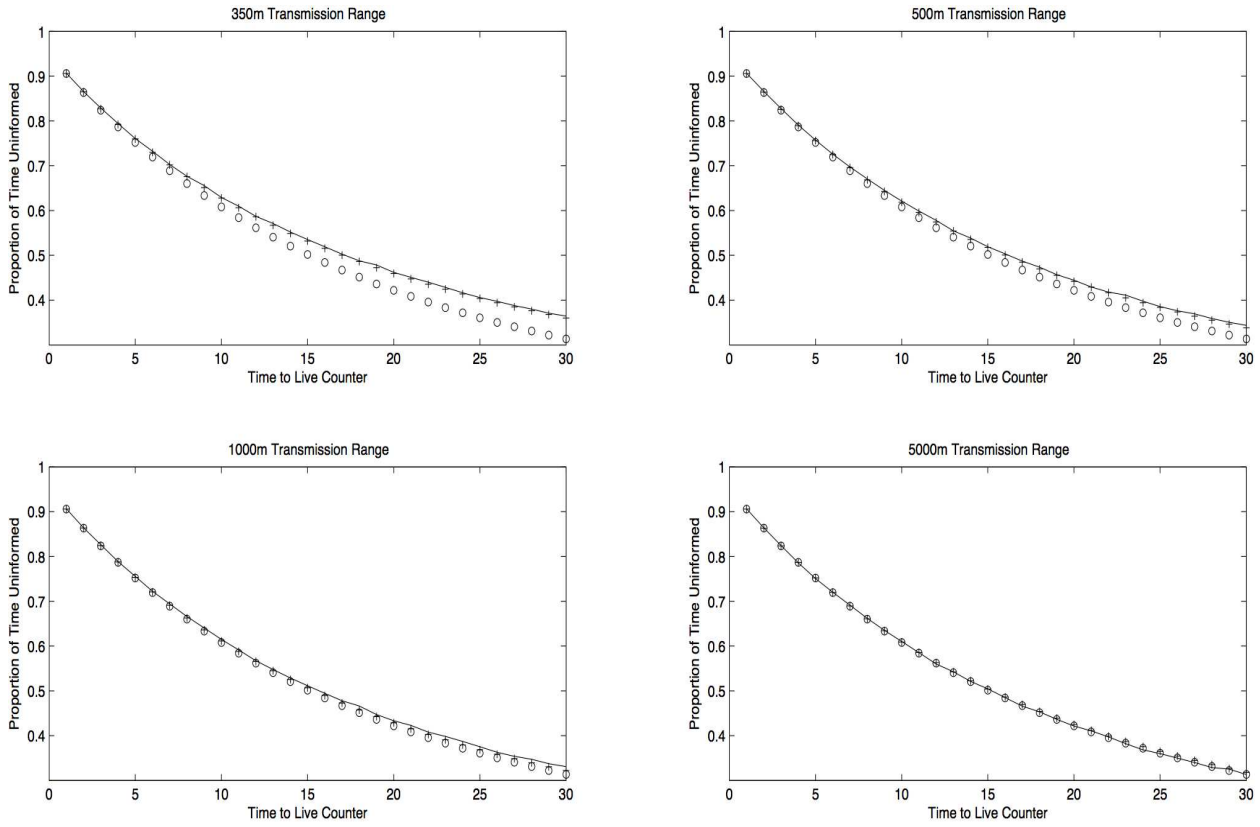


Figure 5: Comparison of π_0 values with Weibull query lifetimes ($N = 1000$): (-) OPNET; (o) $r = \infty$; (+) $r < \infty$.

Table 3: MAD in the proportion of failed queries (D_Δ) when $N = 1000$.

Query lifetime	350m		500m		1000m		5000m	
	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$
Exponential(0.2)	0.0371	0.0246	0.0168	0.0103	0.0047	0.0055	0.0060	0.0052
Triangular(0.1, 5.0, 9.9)	0.0459	0.0301	0.0170	0.0128	0.0030	0.0008	0.0082	0.0024
Uniform(0.1, 9.9)	0.0383	0.0258	0.0178	0.0121	0.0035	0.0015	0.0051	0.0016
Rayleigh(5.645)	0.0237	0.0127	0.0065	0.0158	0.0256	0.0270	0.0255	0.0247
Weibull(3.0, 5.6)	0.0485	0.0306	0.0230	0.0149	0.0031	0.0014	0.0022	0.0014

lifetime distribution is not exponential. Specifically, the magnitudes of the MAD values for the non-exponential cases are generally consistent with those of the exponential case. In the worst case, the MAD of the triangular lifetime distribution exceeds the MAD of the exponential by 0.01494 (5000m range assuming $r = \infty$); however, on average, the increase in the MAD over all the non-exponential cases is 0.0022, or roughly 0.2%. Figure 5 depicts the performance of the approximations and reveals that the finite range approximation is superior to the infinite range approximation for all TTL values when r is small. Indeed, the gap between the latter approximation and OPNET simulation values increases with ℓ since the revisiting effect is more pronounced when the TTL value is large. For larger ranges, the approximations nearly coincide and both closely track the simulated values.

Results for the steady state proportion of query failures are summarized in Table 3. Both approximation schemes perform extremely well (the maximum absolute deviation over all cases is less than 0.049). It is also worth noting that the finite range approximation outperforms the infinite range approximation, particularly when r is relatively small. The results here are also consistent for non-exponential query lifetimes. In the worst case, the MAD of the Rayleigh lifetime distribution exceeds the MAD of the exponential by 0.0215 (1000m range assuming $r < \infty$); on average, the increase in the MAD over all the non-exponential cases is 0.0065, or roughly 0.65%. Figure 6 graphically depicts the four cases.

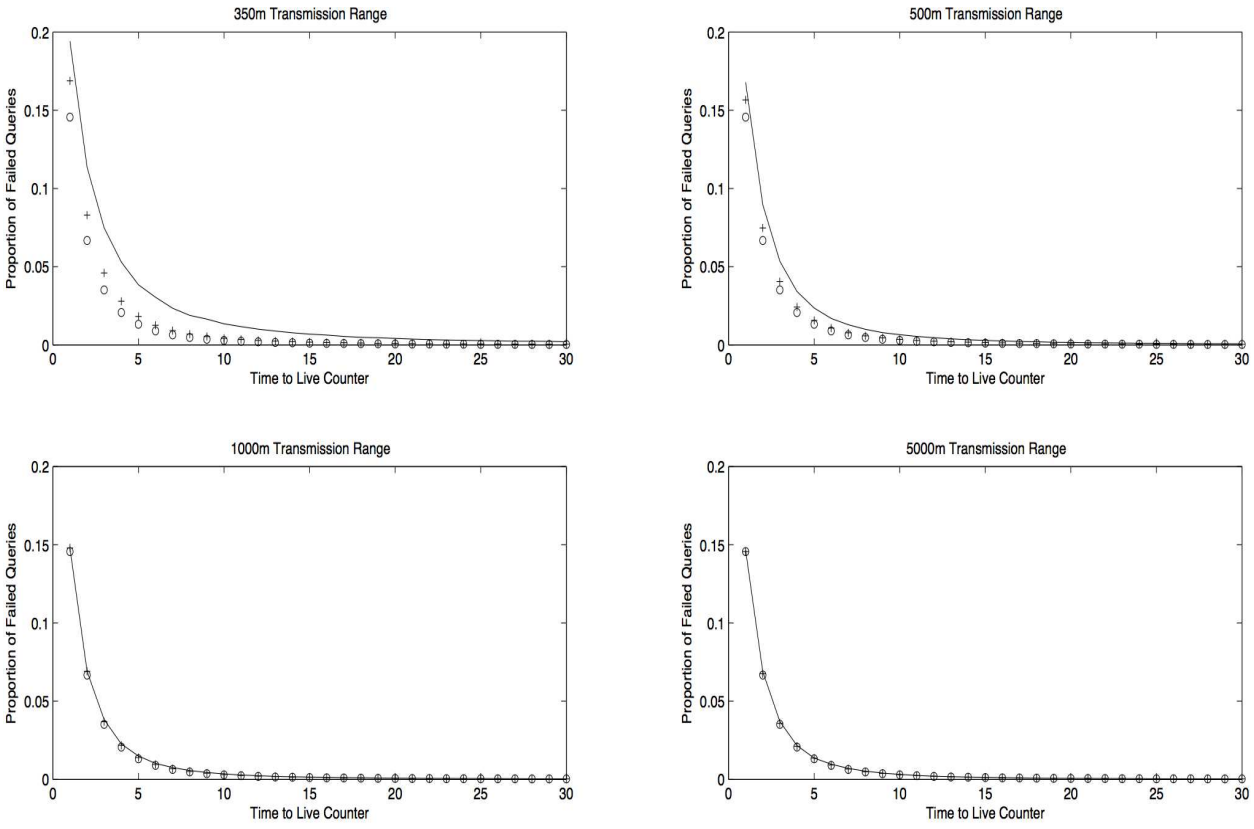


Figure 6: Comparison of Δ values with Weibull query lifetimes ($N = 1000$): (-) OPNET; (o) $r = \infty$; (+) $r < \infty$.

Example 2: 5000-Node Network: Here, we consider a 5000-node wireless sensor network with nodes deployed in the same region as the 1000-node case but with node density $\psi \approx 4.50 \times 10^{-4}$ nodes per square meter. To ensure a connected network with probability 0.9999, the minimum required sensor transmission range is $r = 112$m. Therefore, we considered the following transmission ranges: 115m, 350m, 500m, and 5000m. Table 4 illustrates the quality of both approximations for the 5000-node network. The maximum absolute deviation for the proportion of time uninformed is less than 0.082 for $r = \infty$, and it is reduced to, at most, 0.0174 when the revisiting effect is included. As before, the superiority of the finite range approximation is generally more pronounced for smaller transmission ranges. Figure 7 depicts the simulated and approximated values of π_0 when the query lifetime follows a triangular distribution. When the transmission range is small (115m), we see some discrepancy between the two approximation schemes. However, for the other three cases, the approximations nearly coincide and are very similar to the simulated results ($D_\pi < 0.011$).

Table 4: MAD in the proportion of time uninformed (D_π) when $N = 5000$.

Query lifetime	115m		350m		500m		5000m	
	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$
Exponential(0.2)	0.0605	0.0172	0.0107	0.0019	0.0082	0.0031	0.0040	0.0049
Triangular(0.1, 5.0, 9.9)	0.0612	0.0174	0.0100	0.0015	0.0068	0.0026	0.0053	0.0062
Uniform(0.1, 9.9)	0.0819	0.0054	0.0105	0.0013	0.0074	0.0025	0.0048	0.0058
Rayleigh(5.645)	0.0595	0.0172	0.0101	0.0016	0.0074	0.0026	0.0051	0.0060
Weibull(3.0, 5.6)	0.0611	0.0174	0.0099	0.0014	0.0068	0.0028	0.0073	0.0083

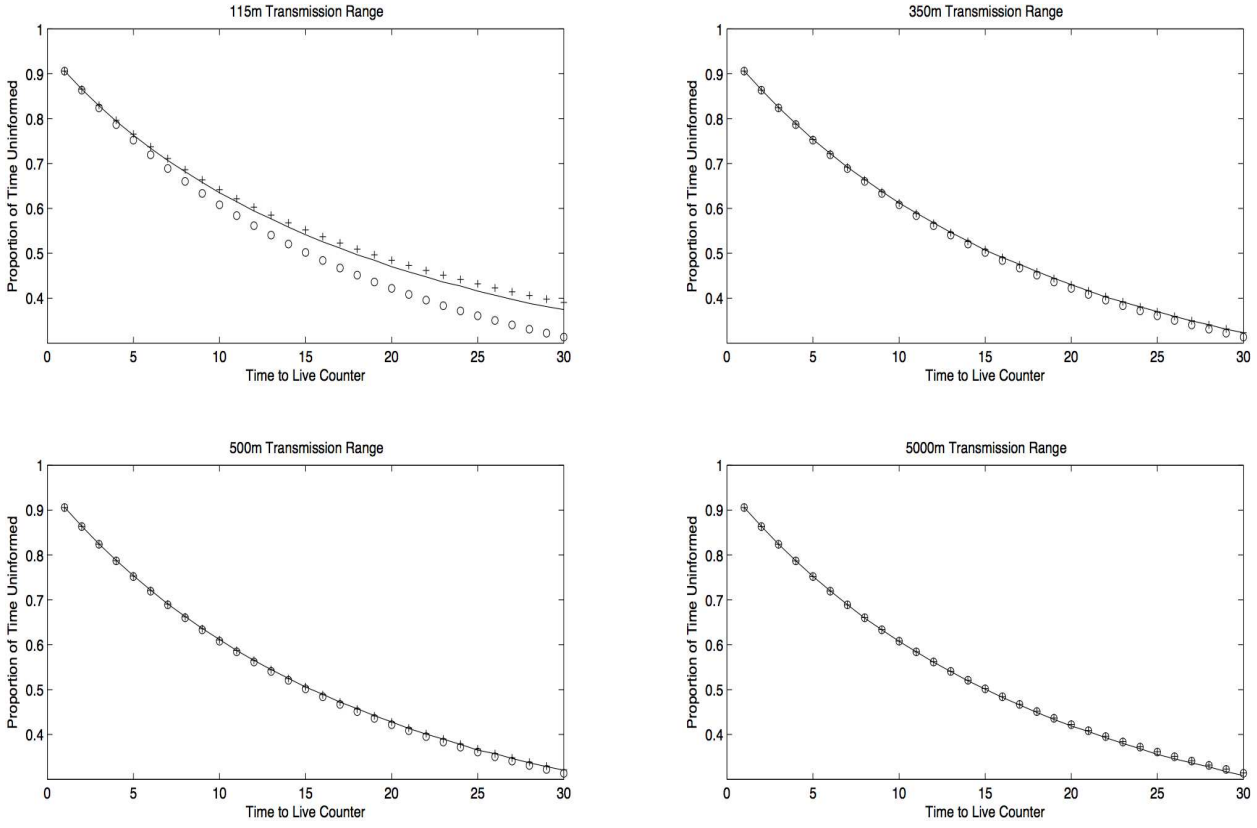


Figure 7: Comparison of π_0 values with triangular query lifetimes ($N = 5000$): (-) OPNET; (o) $r = \infty$; (+) $r < \infty$.

Table 5: MAD in the proportion of query failures (D_Δ) when $N = 5000$.

Query lifetime	115m		350m		500m		5000m	
	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$
Exponential(0.2)	0.0493	0.0283	0.0046	0.0022	0.0045	0.0043	0.0061	0.0044
Triangular(0.1, 5.0, 9.9)	0.0588	0.0333	0.0052	0.0026	0.0013	0.0024	0.0044	0.0045
Uniform(0.1, 9.9)	0.0724	0.0493	0.0044	0.0047	0.0051	0.0021	0.0078	0.0023
Rayleigh(5.645)	0.0371	0.0170	0.0214	0.0192	0.0265	0.0225	0.0286	0.0228
Weibull(3.0, 5.6)	0.0619	0.0351	0.0071	0.0064	0.0025	0.0041	0.0039	0.0021

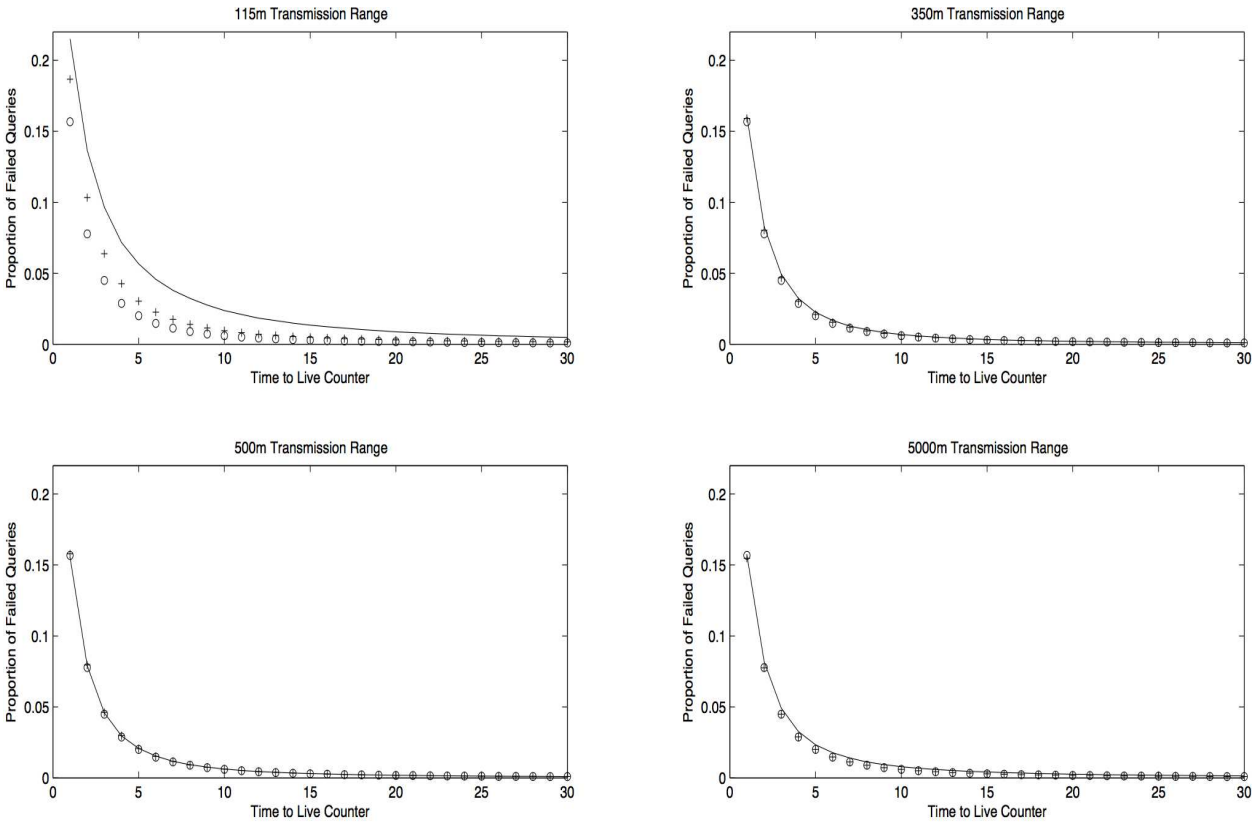


Figure 8: Comparison of Δ values with triangular query lifetimes ($N = 5000$): (-) OPNET; (o) $r = \infty$; (+) $r < \infty$.

Next, we compare the maximum absolute deviation of the proportion of query failures. Table 5 shows that the maximum deviation values are bounded above by 0.0725. Again, the finite range approximation outperforms the infinite range version when the transmission range is small. However, for larger ranges, the results nearly coincide and closely track the simulated values.

Figure 8 graphically depicts the simulated and approximated values of Δ and illustrates the high quality of the approximations. In the worst case (115m), the MAD is less than 0.0725 and 0.05 for $r = \infty$ and $r < \infty$, respectively.

Example 3: Model Validation: We conducted an experiment to validate the approximations when some of the model assumptions regarding the query and event lifetime distributions are violated. For the benchmark simulation experiments presented here, events arrive according to a renewal process with a specified (non-exponential) interarrival time distribution (i.e., the event arrival process is not Poisson). This experiment also employs non-exponential event agent and query lifetimes, both of which are used in the approximations of Sections 2.2 and 2.3.

Table 6 provides a summary of the numerical results for 45 distinct test cases using a 1000-node wireless sensor network with nodes distributed randomly in a 3335m $\times$ 3335m sensor field. The node density is $\psi \approx 9.00 \times 10^{-5}$ nodes per square meter. To ensure a connected network with probability 0.9999, the minimum required sensor transmission range is $r = 239$m; therefore, we set $r = 350$m.

Table 6 reveals some very interesting results. First, we note that the performance of the finite-range approximation is similar to that reported in Example 1 which assumed Poisson-generated events. Specifically, despite the fact that the event arrival process is distinctly non-Poisson, and the query and event lifetimes are not exponential, the benchmark proportion of failed queries is approximated very closely using the finite-range approximation. Over all 45 test cases, the observed maximum absolute deviation between the simulated proportion of query failures and the approximated values (using the finite-range model) is about 0.0319, and the average absolute deviation is about 0.0237. Considering the complexity of event agent and query dynamics, and the random nature of arrivals and transmissions, we consider these discrepancies to be quite acceptable. For example, if an engineer is interested in selecting the optimal TTL value that minimizes energy expenditure while satisfying a quality-of-service constraint based on the proportion of query failures, then our approximation can be used to quickly assess the query failure rate using alternative TTL values. Alternatively, one might consider jointly optimizing the TTL value and the transmission

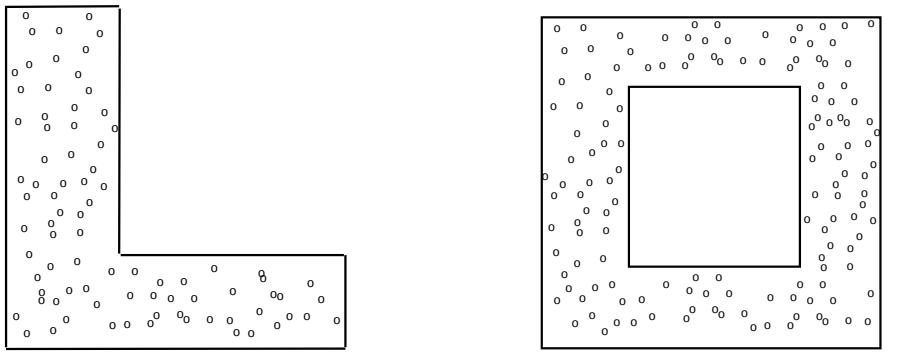
Table 6: MAD in the proportion of time uninformed (D_π) and proportion of failed queries (D_Δ).

Trial	Event interarrival time	Event lifetime	Query lifetime	D_π		D_Δ	
				$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$
1	Er(5, 40.0)	Er(4, 2.5)	Er(5, 1.0)	0.0711	0.0233	0.0421	0.0265
2	Er(5, 40.0)	Tri(0.1, 10.0, 19.9)	Er(5, 1.0)	0.0693	0.0215	0.0428	0.0273
3	Er(5, 40.0)	Uni(0.1, 19.9)	Er(5, 1.0)	0.0596	0.0124	0.0424	0.0273
4	Er(5, 40.0)	Er(4, 2.5)	Ray(5.645)	0.0696	0.0213	0.0201	0.0106
5	Er(5, 40.0)	Tri(0.1, 10.0, 19.9)	Ray(5.645)	0.0699	0.0220	0.0203	0.0106
6	Er(5, 40.0)	Uni(0.1, 19.9)	Ray(5.645)	0.0594	0.0130	0.0213	0.0106
7	Er(5, 40.0)	Er(4, 2.5)	Tri(0.1, 5.0, 9.9)	0.0687	0.0213	0.0399	0.0262
8	Er(5, 40.0)	Tri(0.1, 10.0, 19.9)	Tri(0.1, 5.0, 9.9)	0.0682	0.0206	0.0407	0.0267
9	Er(5, 40.0)	Uni(0.1, 19.9)	Tri(0.1, 5.0, 9.9)	0.0584	0.0100	0.0400	0.0259
10	Er(5, 40.0)	Er(4, 2.5)	Uni(0.1, 9.9)	0.0691	0.0218	0.0340	0.0209
11	Er(5, 40.0)	Tri(0.1, 10.0, 19.9)	Uni(0.1, 9.9)	0.0690	0.0211	0.0342	0.0205
12	Er(5, 40.0)	Uni(0.1, 19.9)	Uni(0.1, 9.9)	0.0589	0.0118	0.0344	0.0211
13	Er(5, 40.0)	Er(4, 2.5)	Weib(3.0, 5.6)	0.0699	0.0229	0.0424	0.0278
14	Er(5, 40.0)	Tri(0.1, 10.0, 19.9)	Weib(3.0, 5.6)	0.0677	0.0201	0.0427	0.0282
15	Er(5, 40.0)	Uni(0.1, 19.9)	Weib(3.0, 5.6)	0.0584	0.0096	0.0428	0.0278
16	Tri(1.0, 200.0, 399.0)	Er(4, 2.5)	Er(5, 1.0)	0.0739	0.0261	0.0438	0.0282
17	Tri(1.0, 200.0, 399.0)	Tri(0.1, 10.0, 19.9)	Er(5, 1.0)	0.0713	0.0235	0.0429	0.0274
18	Tri(1.0, 200.0, 399.0)	Uni(0.1, 19.9)	Er(5, 1.0)	0.0615	0.0145	0.0433	0.0282
19	Tri(1.0, 200.0, 399.0)	Er(4, 2.5)	Ray(5.645)	0.0718	0.0235	0.0213	0.0106
20	Tri(1.0, 200.0, 399.0)	Tri(0.1, 10.0, 19.9)	Ray(5.645)	0.0728	0.0250	0.0215	0.0106
21	Tri(1.0, 200.0, 399.0)	Uni(0.1, 19.9)	Ray(5.645)	0.0614	0.0150	0.0217	0.0114
22	Tri(1.0, 200.0, 399.0)	Er(4, 2.5)	Tri(0.1, 5.0, 9.9)	0.0713	0.0239	0.0410	0.0273
23	Tri(1.0, 200.0, 399.0)	Tri(0.1, 10.0, 19.9)	Tri(0.1, 5.0, 9.9)	0.0725	0.0249	0.0405	0.0265
24	Tri(1.0, 200.0, 399.0)	Uni(0.1, 19.9)	Tri(0.1, 5.0, 9.9)	0.0523	0.0052	0.0418	0.0277
25	Tri(1.0, 200.0, 399.0)	Er(4, 2.5)	Uni(0.1, 9.9)	0.0709	0.0237	0.0343	0.0212
26	Tri(1.0, 200.0, 399.0)	Tri(0.1, 10.0, 19.9)	Uni(0.1, 9.9)	0.0717	0.0238	0.0356	0.0219
27	Tri(1.0, 200.0, 399.0)	Uni(0.1, 19.9)	Uni(0.1, 9.9)	0.0618	0.0146	0.0296	0.0163
28	Tri(1.0, 200.0, 399.0)	Er(4, 2.5)	Weib(3.0, 5.6)	0.0718	0.0247	0.0437	0.0290
29	Tri(1.0, 200.0, 399.0)	Tri(0.1, 10.0, 19.9)	Weib(3.0, 5.6)	0.0721	0.0246	0.0431	0.0286
30	Tri(1.0, 200.0, 399.0)	Uni(0.1, 19.9)	Weib(3.0, 5.6)	0.0519	0.0037	0.0438	0.0288
31	Uni(1.0, 399.0)	Er(4, 2.5)	Er(5, 1.0)	0.0737	0.0259	0.0453	0.0297
32	Uni(1.0, 399.0)	Tri(0.1, 10.0, 19.9)	Er(5, 1.0)	0.0675	0.0201	0.0459	0.0304
33	Uni(1.0, 399.0)	Uni(0.1, 19.9)	Er(5, 1.0)	0.0630	0.0158	0.0457	0.0306
34	Uni(1.0, 399.0)	Er(4, 2.5)	Ray(5.645)	0.0742	0.0260	0.0243	0.0127
35	Uni(1.0, 399.0)	Tri(0.1, 10.0, 19.9)	Ray(5.645)	0.0737	0.0258	0.0234	0.0123
36	Uni(1.0, 399.0)	Uni(0.1, 19.9)	Ray(5.645)	0.0637	0.0167	0.0233	0.0133
37	Uni(1.0, 399.0)	Er(4, 2.5)	Tri(0.1, 5.0, 9.9)	0.0751	0.0276	0.0435	0.0298
38	Uni(1.0, 399.0)	Tri(0.1, 10.0, 19.9)	Tri(0.1, 5.0, 9.9)	0.0745	0.0269	0.0440	0.0300
39	Uni(1.0, 399.0)	Uni(0.1, 19.9)	Tri(0.1, 5.0, 9.9)	0.0634	0.0155	0.0444	0.0300
40	Uni(1.0, 399.0)	Er(4, 2.5)	Uni(0.1, 9.9)	0.0749	0.0277	0.0369	0.0255
41	Uni(1.0, 399.0)	Tri(0.1, 10.0, 19.9)	Uni(0.1, 9.9)	0.0749	0.0270	0.0356	0.0219
42	Uni(1.0, 399.0)	Uni(0.1, 19.9)	Uni(0.1, 9.9)	0.0634	0.0165	0.0379	0.0252
43	Uni(1.0, 399.0)	Er(4, 2.5)	Weib(3.0, 5.6)	0.0728	0.0257	0.0458	0.0311
44	Uni(1.0, 399.0)	Tri(0.1, 10.0, 19.9)	Weib(3.0, 5.6)	0.0746	0.0270	0.0464	0.0319
45	Uni(1.0, 399.0)	Uni(0.1, 19.9)	Weib(3.0, 5.6)	0.0648	0.0161	0.0469	0.0319

range of the sensors in order to maximize network lifetime, subject to an upper limit on the proportion of query failures. Here too, our approximations can be used, in lieu of a simulation model, to quickly evaluate alternative solutions. For such purposes, an average deviation on the order of 0.0237 is tolerable. The results of this section are significant because they provide empirical evidence that the approximations are not heavily influenced by the Poisson arrival assumption imposed at the event tables and the transmission queues. This hypothesis is also supported theoretically in

that the arrival streams at the event tables and transmission queues are superpositions of multiple independent sources. Albin [7] argued that such superpositions are well approximated by a Poisson process if the number of sources is large (say 10 or more), and the traffic intensity (the traffic arrival rate multiplied by the expected service time) at the node is light or moderate. However, the approximation can be poor if the traffic intensity is high, even if the number of sources is large. We conjecture that the Poisson assumption is adequate here because N is large, and the rates at which events are witnessed and/or queries are generated are moderate.

Example 4: Irregular topologies: In previous examples, the sensor field was assumed to be a square shape region. In this example, we explore the impact of boundaries and topology on the performance of approximations. We present results for a 1000-node wireless sensor network with nodes distributed randomly in an *L-shape* (see Figure 9a) and in a *square with a hole (SH)-shape* (see Figure 9b) sensor field. The node density is $\psi \approx 9.00 \times 10^{-5}$ nodes per square meter in both examples.



(a) L-shape deployment region.

(b) SH-shape deployment region.

Figure 9: Graphical depiction of irregular deployment regions ($\mathcal{R}$).

Example 4a: 1000-Node L-Shape Network: Table 7 summarizes the MAD in the proportion of time uninformed using 350m, 500m, 1000m, 5000m transmission ranges. The maximum absolute deviation for the proportion of time uninformed is less than 0.062 for $r = \infty$, and it is reduced to, at most, 0.008 when the revisiting effect is included. Revisiting effect is more pronounced in irregular topologies when the transmission range is small because boundary conditions are more significant. Figure 10 reveals that the finite range approximation is superior for small range values and the results are consistent for all TTL values. For larger ranges, both approximations are matching the OPNET simulation values.

Table 7: MAD in the proportion of time uninformed (D_π) when $N = 1000$ in L-shape deployment area.

Query lifetime	350m		500m		1000m		5000m	
	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$
Exponential(0.2)	0.0616	0.0056	0.0396	0.0073	0.0166	0.0029	0.0082	0.0042
Triangular(0.1, 5.0, 9.9)	0.0624	0.0094	0.0374	0.0055	0.0152	0.0027	0.0086	0.0054
Uniform(0.1, 9.9)	0.0623	0.0076	0.0404	0.0088	0.0149	0.0032	0.0091	0.0047
Rayleigh(5.645)	0.0610	0.0068	0.0385	0.0061	0.0140	0.0037	0.0069	0.0037
Weibull(3.0, 5.6)	0.0604	0.0073	0.0410	0.0074	0.0147	0.0028	0.0078	0.0038

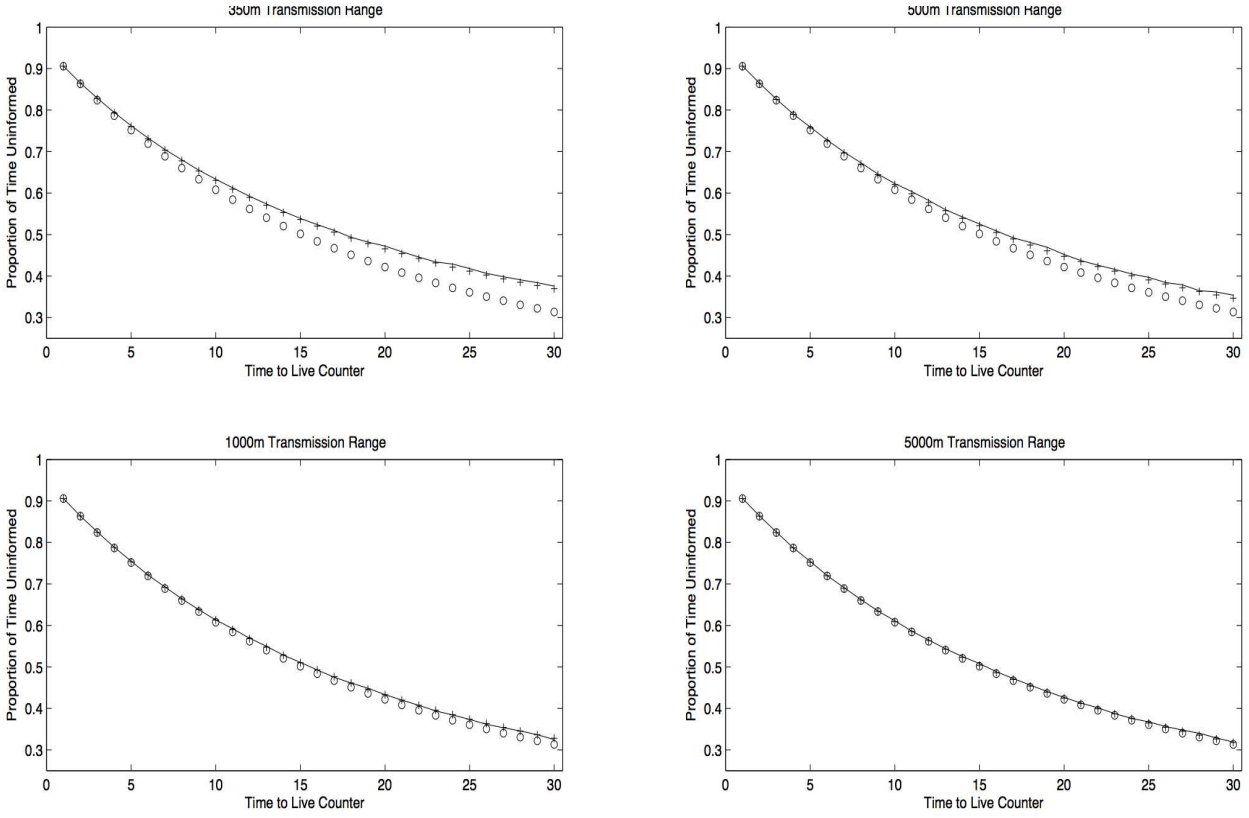


Figure 10: Comparison of π_0 values with uniform query lifetimes ($N = 1000$, L-Shape): (-) OPNET; (o) $r = \infty$; (+) $r < \infty$.

Table 8: MAD in the proportion of failed queries (D_Δ) when $N = 1000$ in L-shape deployment area.

Query lifetime	350m		500m		1000m		5000m	
	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$
Exponential(0.2)	0.0422	0.0297	0.0200	0.0132	0.0068	0.0042	0.0074	0.0045
Triangular(0.1, 5.0, 9.9)	0.0542	0.0367	0.0276	0.0199	0.0096	0.0090	0.0049	0.0008
Uniform(0.1, 9.9)	0.0457	0.0304	0.0225	0.0163	0.0075	0.0068	0.0080	0.0031
Rayleigh(5.645)	0.0312	0.0183	0.0091	0.0091	0.0188	0.0182	0.0284	0.0238
Weibull(3.0, 5.6)	0.0580	0.0400	0.0283	0.0195	0.0102	0.0091	0.0032	0.0014

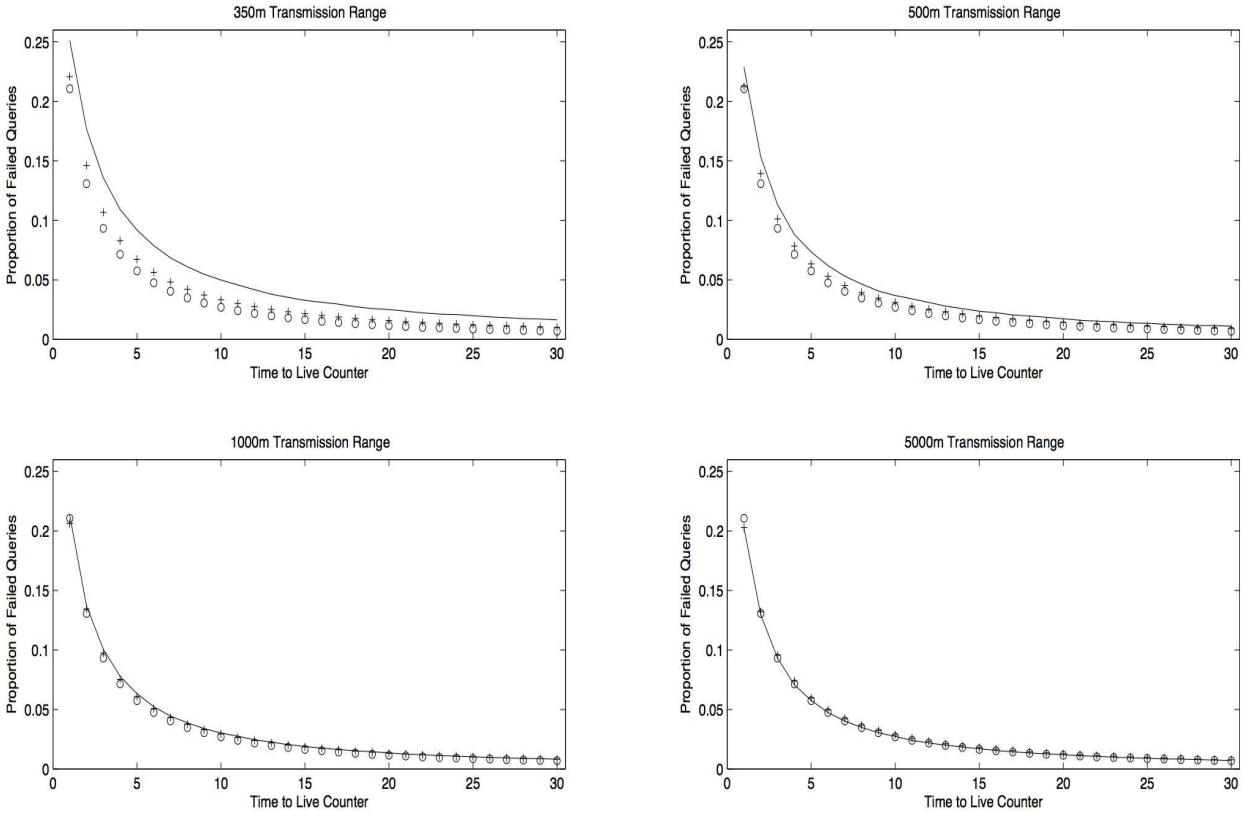


Figure 11: Comparison of Δ values with uniform query lifetimes ($N = 1000$, L-Shape): (-) OPNET; (o) $r = \infty$; (+) $r < \infty$.

Table 9: MAD in the proportion of time uninformed (D_π) when $N = 1000$ in SH-shape deployment area.

Query lifetime	350m		500m		1000m		5000m	
	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$
Exponential(0.2)	0.0615	0.0088	0.0367	0.0078	0.0133	0.0027	0.0026	0.0057
Triangular(0.1, 5.0, 9.9)	0.0602	0.0081	0.0363	0.0070	0.0156	0.0028	0.0032	0.0058
Uniform(0.1, 9.9)	0.0616	0.0091	0.0376	0.0070	0.0140	0.0041	0.0029	0.0070
Rayleigh(5.645)	0.0614	0.0089	0.0401	0.0090	0.0133	0.0034	0.0028	0.0050
Weibull(3.0, 5.6)	0.0594	0.0083	0.0380	0.0080	0.0157	0.0030	0.0023	0.0063

We compare the maximum absolute deviation of the proportion of query failures in Table 8. The performance of the finite range approximation is slightly superior than the infinite range version when the transmission range is small and the maximum deviation values are less than 0.0542. On the other hand, the deviation reduces up to 0.028 for larger ranges. Figure 11 graphically depicts the MAD of the uniform lifetime distribution for the 350m, 500m, 1000m, 5000m transmission range cases.

Example 4b: 1000-Node Square with a Hole (SH)-Shape Network: Here, we present results for a 1000-node wireless sensor network with nodes distributed randomly in a *SH-shape* sensor field (see Figure 9b). Similar to the L-shape example, the node density is $\psi \approx 9.00 \times 10^{-5}$ nodes per square meter. In this example, the effect of revisiting effect is expected to be significant because the boundary conditions are more substantial. Table 9 shows that the MAD in proportion of time uninformed is improved when the finite-range approximation is used, especially for small transmission range values. Because queries are more likely to revisit neighbors when the transmission range is small and boundary conditions are more pronounced. Figure 12 graphically depicts the simulated and approximated values of π_0 for triangular distributed query lifetimes and reveals that in the worst case (350m), the MAD is less than 0.0602 and 0.0081 for $r = \infty$ and $r < \infty$, respectively.

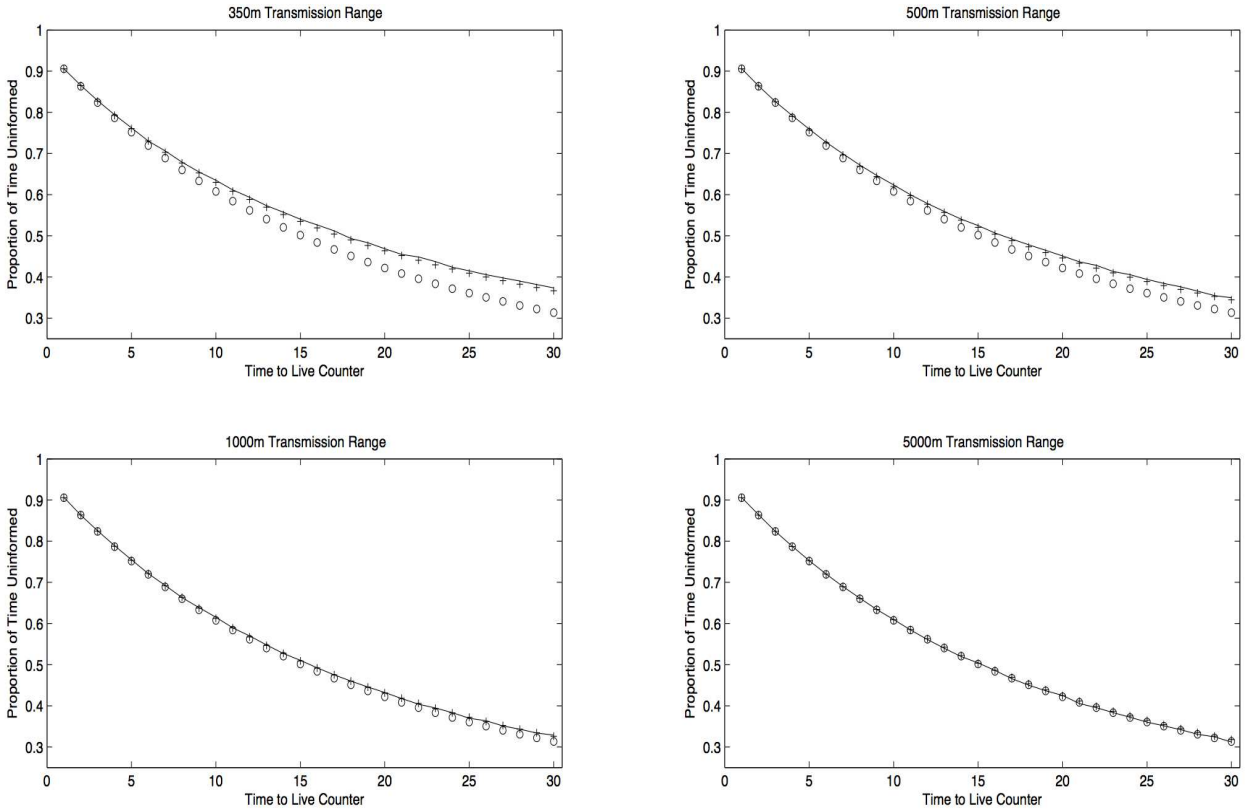


Figure 12: Comparison of π_0 values with triangular query lifetimes ($N = 1000$, SH-Shape): (-) OPNET; (o) $r = \infty$; (+) $r < \infty$.

Table 10: MAD in the proportion of failed queries (D_Δ) when $N = 1000$ in SH-shape deployment area.

Query lifetime	350m		500m		1000m		5000m	
	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$	$r = \infty$	$r < \infty$
Exponential(0.2)	0.0423	0.0290	0.0197	0.0127	0.0056	0.0031	0.0041	0.0012
Triangular(0.1, 5.0, 9.9)	0.0513	0.0357	0.0240	0.0176	0.0067	0.0084	0.0020	0.0044
Uniform(0.1, 9.9)	0.0448	0.0307	0.0204	0.0148	0.0058	0.0049	0.0052	0.0026
Rayleigh(5.645)	0.0306	0.0182	0.0073	0.0103	0.0205	0.0192	0.0254	0.0208
Weibull(3.0, 5.6)	0.0553	0.0377	0.0274	0.0192	0.0082	0.0075	0.0031	0.0039

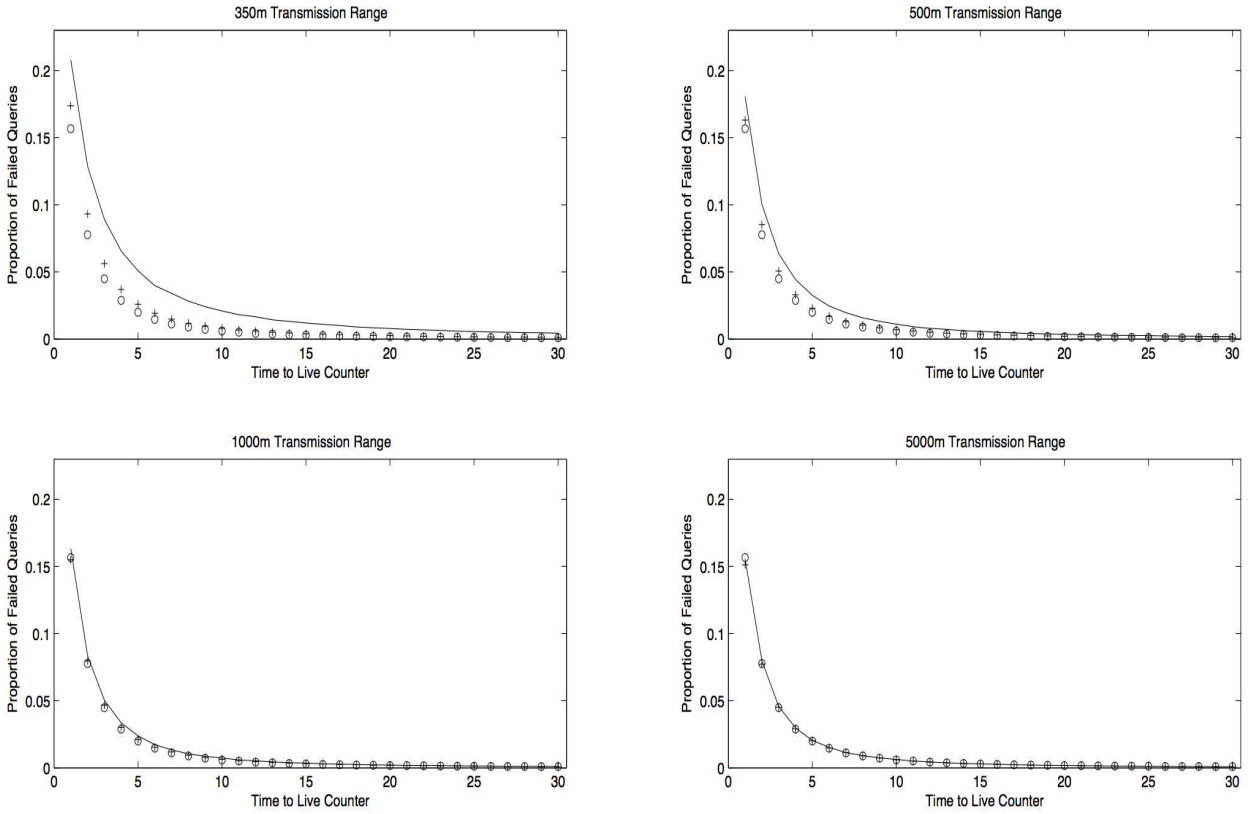


Figure 13: Comparison of Δ values with triangular query lifetimes ($N = 1000$, SH-Shape): (-) OPNET; (o) $r = \infty$; (+) $r < \infty$.

The maximum absolute deviation in proportion of query failures are summarized in Table 10. In the worst case, the MAD in proportion of query failures is 0.0553 and it is reduced to 0.0377 when the finite transmission range is used. The results are consistent for exponential and non-exponential query lifetimes. Figure 13 depicts the depicts performance of the approximations when the query lifetime follows a triangular distribution. We see some discrepancies however the MAD in proportion of query failures is less than 0.0513.

This section has demonstrated that the approximations of Sections 2.2 and 2.3 are remarkably accurate, even when some key model assumptions are violated. Moreover, for each instance, the approximated proportion of time uninformed and proportion of query failures were computed in less than 20 minutes as compared to the OPNET simulation results, which required a minimum of 2 hours.

3.0 MAXIMIZING THE LIFETIME OF A WSN

All of the measures of network lifetime mentioned in Chapter 2 are strongly correlated with the functionality of the sensor nodes which are failure prone due to energy depletion and/or environmental effects. (Our focus here is on failures due to energy depletion.) Furthermore, in many applications, the sensed information is time-sensitive; i.e., deadlines are imposed on significant witnessed events and on query packets transmitted throughout the network. To ensure the timely delivery of critical data or resources, it may be necessary to adjust network parameters (e.g., increasing the transmission range of nodes) at the expense of increased energy consumption. However, there exists a delicate tradeoff between meeting the performance requirements of the network and prolonging its useful lifetime by expeditious use of its energy reserves. The main purpose of this chapter is to formulate and solve optimization models for maximizing the lifetime of a WSN that detects and transmits time-critical data. Of particular concern is the problem of selecting a common transmission range for all of the sensor nodes, the resource replication level (or time-to-live counter) and the active/sleep schedule of nodes while satisfying connectivity and quality-of-service constraints. In our context, the network is no longer functional if it either fails to meet the QoS requirement or it becomes disconnected.

Remarkable energy savings can be achieved by turning-off the communication capability (radio) of a node during idle time-slots (see [5, 97, 103]). Two distinguished modes of a node are "sleep" and "active". When a node is in *active* mode, it performs all of its duties; when it is in *sleep* mode, it continues sensing the environment but does not communicate with other nodes to conserve energy. However, energy is conserved at the expense of network throughput capacity and increased response time. Niyato and Hossain [80] developed a queueing model to investigate the performance of different sleep and wake-up strategies. Chiasserini et al. [27] developed a fluid model to analyze WSNs with a random sleep scheme, where at a given time instant there is a certain probability for an arbitrary node to be active. Liu et al. [69] developed a queuing model-based framework

to study the interaction between random sleep schemes and packet delivery delay, and network throughput as a measure of network performance. Active/sleep schedules are further studied in [47, 90, 102, 24]. Ha et al. [47] developed an integer linear programming model based on a network flow model to schedule the active/sleep modes that satisfy a network coverage constraint. Sarkar and Cruz [90] proposed a dynamic programming formulation to numerically solve the problem of selecting optimal sleep times and durations, subject to an average delay constraint. Turkogullari et al. [102] developed a mixed integer linear programming model to maximize network lifetime by simultaneously setting sleep schedules, sensor locations and data routing, and they proposed heuristic solution algorithms. They concluded that, by simultaneously considering sleep schedules, node placement and routing protocols in a single model, the network lifetime can be substantially improved. Cerulli et al. [24] developed an exact column generation algorithm to maximize the network lifetime by determining subsets of sensors that can cover an entire set of targets in the sensor field and assigning the activation times of the covering sets. They also assume the sensing range is adjustable. Our work here differs from these prior models in that we focus our attention on network operating parameters that significantly influence the quality-of-service and average lifetime of the network. Specifically, we consider the impact of transmission range, the time-to-live (or hop) counter of event packets and active/sleep schedules for the nodes of the network. We explain each of these aspects in what follows.

The maximum one-hop transmission distance of a node is called its *transmission range*. Chen et al. [26] and Deng et al. [33] determined the optimal transmission range of all nodes that minimizes the total network energy expenditure and showed the impact of transmission range on WSN lifetime using a general energy consumption model. They showed that a larger transmission range decreases response time at the expense of higher energy consumption. Gao et al. [42] showed on a simple linear network (i.e., a WSN in which sensor nodes are arranged in a serial configuration) that energy expenditure can be reduced significantly if the transmission range is updated when the network node density changes due to the nodes' active/sleep schedules. Ata [12] considered the problem of dynamically choosing the transmission rate in a general wireless communications network such that the average energy consumption per time unit is minimized, subject to a QoS constraint. In that work, the transmission queue was modeled as a finite-buffer, $M/M/1$ queueing system. Aneja et al. [10] presented three model formulations to assign transmission power (transmission range indirectly) to each node such that there exists a (directed) path between each pair of sensor nodes.

A branch-and-cut algorithm was developed and its performance was evaluated empirically using a network with up to 150 nodes.

While some of the existing research to date has focused on adjusting the transmission range and active/sleep schedules of nodes to maximize the network lifetime, no existing studies simultaneously consider the selection of the transmission range, time-to-live counter, and node active/sleep schedules while taking into account the energy expended by the nodes' batteries. Moreover, (with the exception of [76, 32]) none of the existing models accounts explicitly for the limited lifetimes of event or query packets. The primary objective of this chapter is to formulate optimization models to maximize the number of consecutive time periods in which a query-based WSN simultaneously satisfies QoS and connectivity requirements by optimally selecting the transmission range, time-to-live counter and node active/sleep schedules for each period in a finite planning horizon. The optimal parameter values within each period are selected *a priori* by solving a linearized version of a nonlinear mixed integer programming model. We also derive a proxy for the probability that the network is connected by considering the border effects of a square deployment region. Finally, we examine a special case of the model that does not consider active/sleep decisions and show that the solution to this problem can be obtained by solving a sequence of single-period models.

The remainder of the chapter is organized as follows. Section 3.1 describes the WSN model, introduces essential notation and presents the QoS and connectivity constraints. By considering the border effects of a square deployment region, we also provide an approximation for the network connectivity probability. Section 3.1 concludes by presenting the formal nonlinear mixed integer programming model. Section 3.2 presents a linearized version of the model that is amenable to solution by a commercial solver (such as CPLEX [29]). In Section 3.3, we discuss a special case of the main model in which alive nodes are always in active mode. For this case, we present a simple algorithm for solving a sequence of single-period problems to obtain the optimal solution. Section 3.4 presents computational experiments that illustrate the advantages of optimizing the network parameter values.

3.1 OPTIMIZATION MODEL DESCRIPTION

As in Chapter 2, we assume that event agents are transmitted from node-to-node via a random walk until either the witnessed event expires (i.e., it reaches its deadline), or it exhausts its *time-to-live*

counter. Our model assumes that nodes transmit to a randomly selected node from the set of nodes within its transmission range. Nodes can also generate *queries* to request data or resources from the network. Recall that if a query cannot locate an informed node before its designated *query lifetime* expires, the query is said to have failed. The total proportion of generated queries that are not answered on time, i.e., the *limiting proportion of query failures*, is a critical QoS measure.

In addition to event agents and queries, the sensor nodes themselves have limited lifetimes. If a node fails due to battery drain, then we say the node is *failed*. Otherwise, the node is said to be *alive*. An alive node is either in *sleep* mode or *active* mode. When in the sleep mode, a node turns off its sensing and communication capabilities to conserve energy (see [9]). It is very common in dense networks to place a subset of network nodes in sleep mode for energy conversation to prolong the lifetime of the network. However, the performance of the network can suffer when only some of the nodes are available for sensing and forwarding event agents or query packets.

Recall that N is the number of nodes in the network. We consider a planning horizon $\mathcal{T} = \{1, 2, \ldots, T\}$ ($T < \infty$), where each element of $\mathcal{T}$ is a decision epoch and the time between two epochs is referred to as a period (e.g., one week). At the beginning of a period, each alive node can either switch to the sleep mode with probability $(1 - p_\tau)$, or stay active with probability p_τ. We assume that p_τ is the same for all alive nodes.

In this section, we present a model to determine the transmission range (r_τ), event time-to-live counter (ℓ_τ), and the probability that a node is in active mode (p_τ) for each $\tau \in \mathcal{T} = \{1, 2, \ldots, T\}$. The following notation will be used in our model:

- $\mathcal{A}$: The set of possible policy decisions.

$$\mathcal{A} \quad = \quad \{(r, \ell, p) : r \in (0, \overline{r}],\, \ell \in \mathcal{N} \setminus \{N\},\, p \in (0, 1]\}\,,$$

  where $\overline{r} = \sqrt{2L}$ is the maximum distance between two nodes in the square deployment region $\mathcal{R}$;

- $a_\tau \in \mathcal{A}$: The decision to make at the start of period τ. Note that the triplet a_τ includes decisions about the transmission range (r_τ), event time-to-live counter (ℓ_τ), and the proportion of active nodes (p_τ), i.e.,

$$a_\tau = (r_\tau, \ell_\tau, p_\tau) \in \mathcal{A};$$

- s_τ: The number of nodes that are alive at the start of period τ, $s_\tau \in \mathcal{N}$;

- n_τ: The expected number of active nodes at the start of period τ. To be conservative, we set

$$n_\tau = \lfloor s_\tau p_\tau \rfloor;$$

- $c(s_\tau, a_\tau)$: The expected battery energy expended by an active node during period τ. We assume that in period τ, the energy expenditures of all nodes are independent and identically distributed (i.i.d.) random variables denoted by U_τ with common mean $\mathbb{E}(U_\tau) = c(s_\tau, a_\tau)$, i.e.,

$$c(s_\tau, a_\tau) = (e_t + e_d\, r_\tau^\eta)\, \lambda_q,$$

where e_t is the energy/bit consumed by the transmitter electronics, e_d accounts for energy dissipated in the transmission, r_τ is the transmission range and η is the path-loss coefficient (see [42],[116]);

- b_τ: The expected available battery energy at a node at the start of period τ. To simplify matters, we will use the mean available energy as a proxy for the actual available energy at a single node. An alive node (in sleep mode) does not consume energy with probability $1 - p_\tau$; however, it is in active mode and consumes energy with probability p_τ. Therefore, the expected available energy during period $\tau + 1$ is obtained recursively by

$$b_{\tau+1} = b_\tau - p_\tau c(s_\tau, a_\tau);$$

- $f(s_\tau, a_\tau, b_\tau)$: The probability that a node alive at the start of period τ fails during this period. We assume that any active node fails at the end of period τ if the energy required during the period exceeds the *mean* energy available at the start of the period. Specifically,

$$f(s_\tau, a_\tau, b_\tau) = \mathbb{P}(U_\tau > b_\tau), \quad \tau \in \mathcal{T} \setminus \{T\}.$$

We use this probability to compute the conservative estimate

$$s_{\tau+1} = \lfloor s_\tau - f_\tau(s_\tau, a_\tau, b_\tau) n_\tau \rfloor,$$

since only active nodes can fail due to battery drain;

- x_τ: The status of the network at the start of period τ.

$$x_\tau = \begin{cases} 1, & \text{if the network is } \textit{functional}, \\ 0, & \text{otherwise.} \end{cases}$$

By *functional*, we mean that the network is connected and satisfies the QoS requirement at the start of period τ.

Let $\Delta(n_\tau, a_\tau)$ be the proportion of query failures at time τ. The limiting proportion of queries that fail to be answered on time is a critical QoS measure for query-based WSNs. We presented an approximation for this proportion that has been show to be very accurate and fairly insensitive to the event and query lifetime distributions in Chapter 2. It plays a crucial role in the optimization model. The proportion of query failures is obtained by considering the limiting behavior of a single query that is generated at an uninformed node and it is provided in Section 2.3.

The energy expended for transmitting event agents or forwarding queries serves as a proxy for the total energy expenditure at a node since transmitting is the primary energy-consuming activity. In the next section we first describe the expected energy expenditure.

3.1.1 Expected Energy Expenditure

Let $\lambda_q(r_\tau, \ell_\tau, n_\tau)$ be the total arrival rate of event agents and queries to a node's transmission queue in period τ. While this rate depends explicitly on r, ℓ, and n, we will suppress this notation and simply write λ_q. The energy expended for transmitting event agents or forwarding queries serves as a proxy for the total energy expenditure at a node. An approximation for λ_q was derived in Section 2.3 and is given by

$$\lambda_q \approx \lambda \left[\frac{1-(1-\alpha)^\ell}{\alpha} \right] + \gamma \pi_0(r) \left[\frac{2-\pi_0(r)}{1-\pi_0(r)} \right]. \tag{3.1}$$

The energy expended per transmission by a node can be computed as $E_t = (e_t + e_d r^\eta)$, where e_t is the energy/bit consumed by the transmitter electronics, e_d accounts for energy dissipated in the transmission, r is the transmission range and η is the path-loss coefficient (see [42],[116]). Therefore, the expected energy expenditure for transmissions is

$$c(s_\tau, a_\tau) = (e_t + e_d\, r_\tau^\eta)\, \lambda_q. \tag{3.2}$$

It is important to note that, due to the dependence of λ_q on the decision variables r_τ, ℓ_τ and n_τ, the structural properties of the function $c(s_\tau, a_\tau)$ (e.g., convexity, monotonicity, etc.) are not obvious.

At each time interval, in addition to the quality of service requirement, the network must always be connected, i.e., there are no disconnected subgroups of nodes. In the next section, we provide an approximate connectivity constraint.

3.1.2 Approximate Connectivity Constraint

Network connectivity is obviously an important consideration for static WSNs whose nodes are subject to failure and sleep schedules. In this subsection, we provide an approximate expression for the probability that the network is connected. Viewing the WSN as an undirected graph, the network is said to be *connected* if for each pair of nodes $i, j \in \mathcal{N}$, there exists at least one single- or multiple-hop path between i and j. As noted by Bettstetter [16], the connectivity of a WSN is closely related to the number of *isolated* nodes in the network. A node $i \in \mathcal{N}$ is said to be isolated if there are no other nodes within its transmission range (i.e., if its node degree is 0); the WSN contains no isolated nodes if the minimum node degree of the network is positive. The probability that there are no isolated nodes in a random graph can be used to bound the connectivity probability from above, and we use this fact to establish an approximate connectivity constraint.

To frame this discussion, we adopt the following definitions and notation, similar to those used in [16]. The WSN is a random graph $\mathcal{G}$ whose N stationary nodes are uniformly distributed in a two-dimensional region $\mathcal{R}$ with area L, i.e., the locations of the nodes can be viewed as points in $\mathcal{R}$ generated by a two-dimensional, spatial Poisson process with constant intensity (node density) N/L. Assuming that each sensor node uses the same transmission range r, and that $L \gg \pi r^2$, one can ignore the border effects of $\mathcal{R}$ to obtain the exact probability that no node is isolated. Let E_0 be the event that no node is isolated when the border effects can be ignored, and let $\widetilde{\Lambda}$ be the true probability that no node is isolated in the network. Clearly, $\widetilde{\Lambda} \leq \mathbb{P}(E_0)$ since the degrees of nodes near the borders of $\mathcal{R}$ are smaller than those whose transmission areas are in the interior of $\mathcal{R}$. Let C be the event that the network is connected. Using basic results for spatially homogeneous Poisson point processes in two dimensions (cf. Diggle [35]), Proposition 3.1 provides an upper bound for the probability that the network is connected, $\mathbb{P}(C)$.

Proposition 3.1. *Suppose $\mathcal{G}$ is an ad hoc WSN with N stationary sensor nodes distributed uniformly in a region of area L. Assuming each node uses the same transmission range r ($r > 0$), an upper bound for the connectivity probability of the network is*

$$\mathbb{P}(C) \leq \widetilde{\Lambda} \leq \mathbb{P}(E_0) = \left[1 - \exp(-N\pi r^2/L)\right]^N . \tag{3.3}$$

By Theorem 1 of [16], the minimum transmission range needed to ensure no isolated nodes with probability ζ is

$$r \geq \sqrt{\frac{-\ln(1-\zeta^{1/N})}{\pi N/L}}, \tag{3.4}$$

or equivalently, one may choose r such that

$$\zeta \leq \left[1-\exp(-N\pi r^2/L)\right]^N.$$

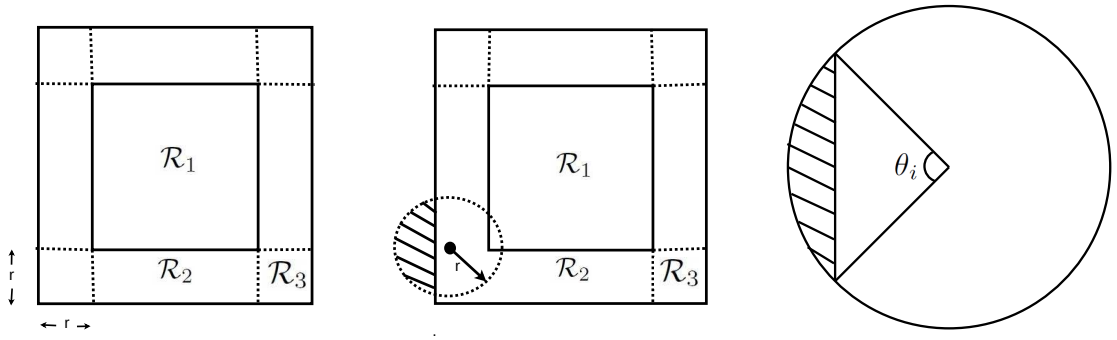
To simplify notation, let $\widehat{\Lambda}_1 := \left[1-\exp(-N\pi r^2/L)\right]^N$ serve as an approximation for $\widetilde{\Lambda}$. It is important to note that the upper bound of (3.3) and, consequently, the lower bound of (3.4), ignore the border effects of the deployment region. By accounting for these effects, our aim is to derive an improved approximation for $\widetilde{\Lambda}$ (call it $\widehat{\Lambda}_2$), thereby providing a tighter upper bound for $\mathbb{P}(C)$. Even though (3.3) provides only an upper bound for $\mathbb{P}(C)$, Bettstetter [17] has shown that when $\widetilde{\Lambda}$ is close to 1, the bound is in fact tight.

Let $d_i(r)$ be the degree of node $i \in \mathcal{N}$ when all the nodes use transmission range r. Our approximation is formed by partitioning $\mathcal{R}$ into three subareas, $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$, as depicted in Figure 14a. Nodes in $\mathcal{R}_1$ are located at least r units away from any of the borders. Nodes in $\mathcal{R}_2$ are located closer than r units to one of the borders but at least r units away from the others. Nodes in $\mathcal{R}_3$ are located closer than r units to two of the borders. To simplify matters slightly, we will use the degree of nodes in $\mathcal{R}_2$ to approximate the degree of nodes in $\mathcal{R}_3$. The expected degrees of nodes in $\mathcal{R}_1$ and $\mathcal{R}_2$ are provided in Lemma 3.1.

Lemma 3.1. *Consider a WSN with $N \geq 1$ nodes. Let $\mathcal{N}_1$ and $\mathcal{N}_2$ be the set of nodes that are located in $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively. Then,*

$$\mathbb{E}[d_i(r)] \approx \begin{cases} \dfrac{N\pi r^2}{L}, & i \in \mathcal{N}_1, \\ \dfrac{(3\pi^2+4)r^2}{4\pi} \cdot \dfrac{N}{L}, & i \in \mathcal{N}_2. \end{cases}$$

Proof. Nodes in $\mathcal{R}_1$ are located at least r units away from any of the borders; therefore, their transmission area is πr^2, and the expected node degree is just the node density multiplied by the area, or $(N/L)\pi r^2$. Nodes in $\mathcal{R}_2$ are located closer than r units to the border, so they have a smaller transmission area, leading to a smaller degree. Figures 14b and 14c illustrate the transmission area of a node located in $\mathcal{R}_2$. In these two figures, no nodes are located in the shaded regions. Let $A_0(i)$ be the area of intersection of node i's transmission area and the network's deployment region. Moreover, let θ_i be a random variable denoting the central angle of the circle surrounding the



(a) Partitioning the region. (b) The effect of $\mathcal{R}$'s borders. (c) Transmission area of a node.

Figure 14: Graphical depiction of the partitioning of $\mathcal{R}$ and the border effects.

transmission area of node i so that θ_i sweeps the arc that remains of the network's deployment region as seen in Figure 14c. Then, for $i \in \mathcal{N}_2$

$$A_0(i) = \pi r^2 - \left(\pi r^2 \frac{\theta_i}{2\pi} - \frac{r^2 \sin\theta_i}{2}\right),$$

and the expectation of $A_0(i)$ is

$$\begin{aligned} \mathbb{E}[A_0(i)] &= \pi r^2 - \frac{r^2}{2}\int_0^{\pi} (u - \sin u)\,\frac{1}{\pi}\,\mathrm{d}u \\ &= \frac{(3\pi^2+4)r^2}{4\pi}. \end{aligned} \tag{3.5}$$

Therefore, the expected degree of a node in $\mathcal{R}_2$ is $(N/L)[(3\pi^2+4)r^2]/4\pi$. □

We use (3.5) as an approximation for $\mathbb{E}[A_0(i)]$ for each $i \in \mathcal{N}_2 \cup \mathcal{N}_3$, where $\mathcal{N}_3$ is the set of nodes located in $\mathcal{R}_3$. It is worth noting that a similar approach was taken in [107]; however, in [107], the authors derive and use a bound for $\mathbb{E}[A_0(i)]$, the expected transmission area of a node in $\mathcal{N}_2$, which directly affects the result. Next, we state the probability that there is no isolated node in the network.

Lemma 3.2 (Equation (23) of [17]). *Consider a node at a given location $\boldsymbol{x}$ in the deployment area. It is randomly placed according to p.d.f. $h(\boldsymbol{x})$. Let $g(r,\boldsymbol{x})$ be the expected degree of a node located at $\boldsymbol{x}$. Given a WSN with $N \geq 120$ and $\pi r^2/L \leq 0.08$, the probability that there is no isolated node in the network is*

$$\exp\left(-N\int_{\mathcal{R}} e^{-g(r,\boldsymbol{x})} h(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right).$$

Proposition 3.2. *Assuming that nodes are spatially uniformly distributed in $\mathcal{R}$, considering the boundary effects, the probability that there is no isolated node in the network, $\widetilde{\Lambda}$, is approximated by*

$$\widetilde{\Lambda} \approx \widehat{\Lambda}_2 = \exp\left[-Ne^{-\frac{N\pi r^2}{L}}\left(\frac{(\sqrt{L}-2r)^2}{L}\right)\right]\exp\left[-Ne^{-\frac{N(3\pi^2+4)r^2}{4\pi L}}\left(1-\frac{(\sqrt{L}-2r)^2}{L}\right)\right]. \quad (3.6)$$

Proof. We employ a uniform node distribution over a finite, square deployment region $\mathcal{R}$; therefore,

$$h(\boldsymbol{x}) = \frac{1}{L}, \quad \boldsymbol{x} \in \mathcal{R}.$$

By Lemmas 3.1 and 3.2,

$$\begin{aligned}\widehat{\Lambda}_2 &= \exp\left[-N\left(\int_{\mathcal{R}_1} e^{-\frac{N\pi r^2}{L}}\frac{1}{L}\mathrm{d}\boldsymbol{x} + \int_{\mathcal{R}_2\cup\mathcal{R}_3} e^{-\frac{N(3\pi^2+4)r^2}{4\pi L}}\frac{1}{L}\mathrm{d}\boldsymbol{x}\right)\right] \\ &= \exp\left[-Ne^{-\frac{N\pi r^2}{L}}\left(\frac{(\sqrt{L}-2r)^2}{L}\right)\right]\exp\left[-Ne^{-\frac{N(3\pi^2+4)r^2}{4\pi L}}\left(1-\frac{(\sqrt{L}-2r)^2}{L}\right)\right].\end{aligned}$$

□

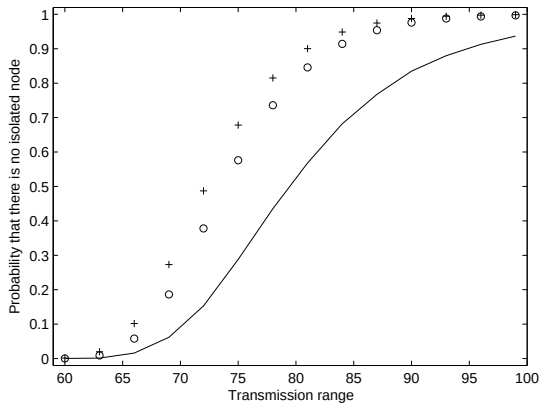
We assess the quality of approximations $\widehat{\Lambda}_1$ and $\widehat{\Lambda}_2$ by comparing them to simulated values of $\widetilde{\Lambda}$ by generating a large number of random networks. We deployed, respectively, 1000 and 5000 nodes on a 1500 m × 1500 m region and determined if the network contains at least one isolated node (for a given transmission range r). The experiment was replicated K times and the proportion of networks with no isolated nodes estimated. Specifically, let $\mathcal{G}_i$ denote the ith generated network and let

$$\mathbf{1}(\mathcal{G}_i) = \begin{cases} 1, & \text{if network } i \text{ has no isolated nodes,} \\ 0, & \text{if network } i \text{ has at least one isolated node.} \end{cases}$$

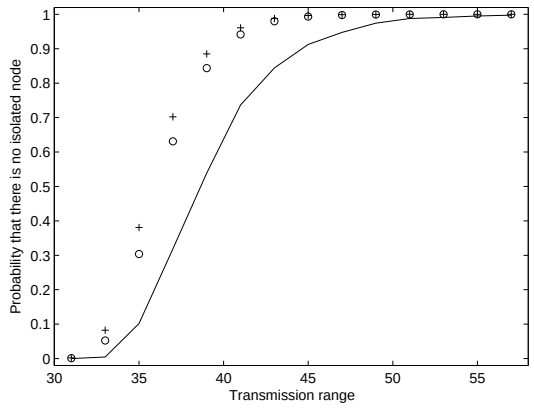
Assuming existence of the limit,

$$\frac{1}{K}\sum_{i=1}^{K}\mathbf{1}(\mathcal{G}_i) \to \widetilde{\Lambda} \quad \text{w.p. } 1$$

as $K \to \infty$ by the strong law of large numbers. Therefore, for K sufficiently large, we obtain a reasonable (relative frequency) estimate of $\widetilde{\Lambda}$. For both scenarios ($N = 1000$ and $N = 5000$), and for each selected transmission range, we randomly generated $K = 10^4$ networks. Figure 15, which depicts the relative performance of the approximations, provides some important insights.



(a) $N = 1000$ nodes.



(b) $N = 5000$ nodes.

Figure 15: Comparison of $\widetilde{\Lambda}$ values: (-) Simulated values; (+) approximation $\widehat{\Lambda}_1$; (o) approximation $\widehat{\Lambda}_2$. Nodes deployed on a 1500 m×1500 m square region.

First, both Figures 15a and 15b indicate that the upper bound with border effects ($\widehat{\Lambda}_2$) is tighter than the one that ignores border effects ($\widehat{\Lambda}_1$) for most of the transmission ranges. Second, as expected, the approximations are close to the simulated values when $\widetilde{\Lambda}$ is close to 1. This latter observation is significant because the minimum threshold for the connectivity probability will be set close to 1 (e.g., 0.99). Therefore, it can be argued that either approximation is suitable for the connectivity constraint, but we prefer $\widehat{\Lambda}_2$ since it exhibits better performance over the set of transmission range values.

Finally, we let $\Psi(n_\tau, a_\tau)$ be the probability that the network is connected at time τ and note that Ψ is an explicit function of n_τ and r_τ. Based on the empirical tests presented in this subsection, we use the result of Proposition 3.2 to ensure the network remains connected with high probability in the optimization model. Specifically,

$$\Psi(n_\tau, a_\tau) \approx \widehat{\Lambda}_2 = \exp\left[-n_\tau e^{-\frac{n_\tau \pi r_\tau^2}{L}}\left(\frac{(\sqrt{L}-2r_\tau)^2}{L}\right)\right] \times \exp\left[-n_\tau e^{-\frac{n_\tau(3\pi^2+4)r^2}{4\pi L}}\left(1-\frac{(\sqrt{L}-2r_\tau)^2}{L}\right)\right]. \quad (3.7)$$

We pause here to remark that the approximation (3.7) assumes that at the start of period τ, the node density n_τ/L is constant throughout $\mathcal{R}$. However, in general the node density will not be constant as some nodes fail over time due to battery depletion, while others enter sleep mode during

certain periods of the planning horizon. Nevertheless, we use (3.7) as the approximate connectivity due to its tractability and the fact that it accounts for the border effects of the region. The next subsection describes the main optimization model.

3.1.3 Optimization Model

Let φ be the maximum allowable proportion of query failures, ζ be the minimum allowable probability that the network is connected and $\bar{b}$ is the initial available energy at a node. In general, ζ should be close to 1 and φ should be close to 0. We propose the following model to determine the optimal transmission range, time-to-live counter and proportion of active modes (among those that are not failed) for each decision epoch in $\mathcal{T}$. We label the first formulation as problem $\mathbf{P}_1$.

$$
\begin{aligned}
(\mathbf{P}_1) \quad \max \quad & \sum_{\tau=1}^{T} x_\tau && && \text{(3.8a)}\\
\text{s.t.} \quad & \Delta(n_\tau, a_\tau) \leq \varphi + M(1 - x_\tau), \quad \tau \in \mathcal{T} && && \text{(3.8b)}\\
& \Psi(n_\tau, a_\tau) \geq \zeta - M(1 - x_\tau), \quad \tau \in \mathcal{T} && && \text{(3.8c)}\\
& n_\tau \leq s_\tau p_\tau, \quad \tau \in \mathcal{T} && && \text{(3.8d)}\\
& s_{\tau+1} \leq s_\tau - f(s_\tau, a_\tau, b_\tau) n_\tau, \quad \tau \in \mathcal{T} \setminus \{T\} && && \text{(3.8e)}\\
& b_{\tau+1} = b_\tau - p_\tau c(s_\tau, a_\tau), \quad \tau \in \mathcal{T} \setminus \{T\} && && \text{(3.8f)}\\
& x_\tau \geq x_{\tau+1}, \quad \tau \in \mathcal{T} \setminus \{T\} && && \text{(3.8g)}\\
& a_\tau = \{r_\tau, \ell_\tau, p_\tau\} \in \mathcal{A}, \quad \tau \in \mathcal{T} && && \text{(3.8h)}\\
& x_\tau \in \{0, 1\}, \, b_\tau \in [0, \bar{b}], \quad \tau \in \mathcal{T} && && \text{(3.8i)}\\
& s_\tau \in \{0, 1, \ldots, N\}, \, n_\tau \in \{0, \ldots, N\}, \, \tau \in \mathcal{T}. && && \text{(3.8j)}
\end{aligned}
$$

The objective function (3.8a) represents the number of *consecutive* time periods in which the network satisfies the QoS and connectivity requirements. In constraints (3.8b) and (3.8c), M is a large positive constant which forces x_τ to assume the value 1 if and only if the QoS and connectivity constraints are simultaneously satisfied in the τth period. For the sake of completeness, we also include constraints (3.8d) and (3.8e) to set the number of active and alive nodes, respectively. Constraint (3.8f) ensures that the expected available energy at each time period is computed considering the energy expenditures in the prior time period. Constraint (3.8g) ensures that the network satisfies the QoS and connectivity constraints in consecutive periods. The admissible ranges of variables are given in (3.8h) through (3.8j).

Solving problem $\mathbf{P}_1$ is nontrivial for at least the following reasons. First, constraints (3.8b) through (3.8f) are nonlinear, and the left-hand side of (3.8b) requires numerical integration. Second, the structure of the constraints, as well as the convexity of $c(s_\tau, a_\tau)$ in ℓ_τ, are difficult to prove; therefore, it is unclear if the feasible region is convex. Fortunately, this model can be viewed as a nonlinear knapsack-like problem with additional constraints. One possible approach for addressing this type of problem is to linearize the model (a typical example is given in [53]) and solve it using a commercial solver (e.g., CPLEX [29]). In the next section, we show how to linearize $\mathbf{P}_1$ and solve it.

3.2 LINEARIZED MODEL FOR MAXIMIZING WSN LIFETIME

In this section, we linearize problem $\mathbf{P}_1$ to set the transmission range (r_τ), event time-to-live counter (ℓ_τ) and the proportion of nodes that are in active mode (p_τ), for each $\tau \in \mathcal{T} = \{1, 2, \ldots, T\}$. Piecewise linear approximations of the functions $c(s_\tau, a_\tau)$ and $f(s_\tau, a_\tau, b_\tau)$ (see (3.8e) and (3.8f)) are used to convert the nonlinear problem ($\mathbf{P}_1$) into a linear 0–1 problem. In addition to the previous notation, we introduce the following:

- $\mathcal{A}(s) \subset \mathcal{A}$: For a given of number of alive nodes s, $\mathcal{A}(s)$ is some subset of policy decisions $\{r, \ell, p\}$ satisfying the following requirements:
  (i) The proportion of query failures and connectivity requirements are satisfied, i.e.,

$$\Delta(n, \{r, \ell, p\}) \leq \varphi \text{ and } \Psi(n, \{r, \ell, p\}) \geq \zeta;$$

  (ii) The transmission range is set such that the energy expenditure is minimized for a given ℓ and p. That is, there does not exist an $r' \neq r$ such that

$$\begin{aligned} \Delta(n, \{r', \ell, p\}) &\leq \varphi, \\ \Psi(n, \{r', \ell, p\}) &\geq \zeta, \\ c(s, \{r', \ell, p\}) &< c(s, \{r, \ell, p\}). \end{aligned}$$

Model parameters:

- c_{sa}: The expected energy expenditure at an alive node for $a \in \mathcal{A}(s)$ and $s \in \mathcal{N}$, i.e.,

$$c_{sa} = c(s, a), \text{ where } a = \{r, \ell, p\} \in \mathcal{A}(s);$$

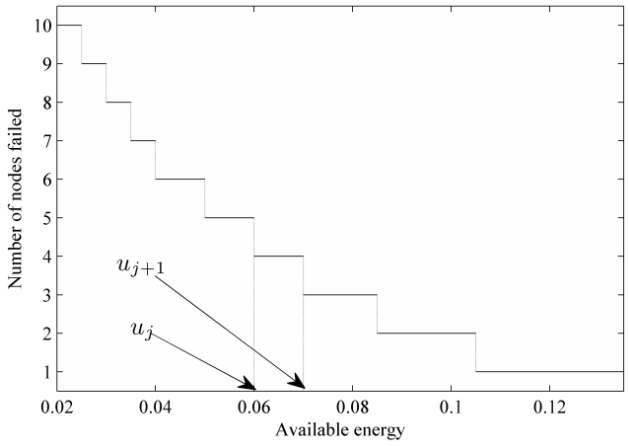


Figure 16: Discretization of the interval $[0, \bar{b}]$.

- u_j, $j \in J = \{1, 2, \ldots, B\}$, where B is a finite positive integer so that $|J| < \infty$: A split-point in $[0, \bar{b}]$ to discretize the interval in which the expected available energy is defined. Note that $u_j < u_{j+1}$ and $u_{j+1} - u_j$ is assumed to be small enough to ensure that, for any given $a \in \mathcal{A}(s)$ and $s \in \mathcal{N}$, the number of failed nodes is constant for $b_\tau \in [u_j, u_{j+1})$. Moreover, $u_1 = 0$ and $u_{|J|} = \bar{b}$. Figure 16 depicts the relationship of u_j values to the node failures for a given $a \in \mathcal{A}(s)$ and $s \in \mathcal{N}$, which is denoted by m_{sa}^τ (see the definition of m_{sa}^τ that follows);

- k_{saj}: The number of nodes that fail when number of alive nodes is s, the policy decision is a and $b_\tau \in [u_j, u_{j+1})$. Note that the expected number of nodes failed is used as a proxy for this variable, i.e., $k_{saj} = spf(s, a, u_j)$, for $a \in \mathcal{A}(s)$;

Model variables:

- y_{sa}^τ: A binary variable to set the policy decisions, i.e., for $\tau \in \mathcal{T}$, $a \in \mathcal{A}(s)$ and $s \in \mathcal{N}$

$$y_{sa}^\tau = \begin{cases} 1, & \text{if } a_\tau = a \text{ and } s_\tau = s, \\ 0, & \text{otherwise;} \end{cases}$$

- z_j^τ: A binary variable to set the expected available energy, i.e., for $\tau \in \mathcal{T}$ and $j \in J$

$$z_j^\tau = \begin{cases} 1, & \text{if } u_j \leq b_\tau < u_{j+1}, \\ 0, & \text{otherwise;} \end{cases} \tag{3.9}$$

- m_{sa}^{τ}: An integer variable to set the number of nodes failed in the τth period, i.e.,

$$m_{sa}^{\tau} = \begin{cases} k_{saj}, & \text{if } y_{sa}^{\tau} = 1 \text{ and } z_j^{\tau} = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Using these variables, we next present a mixed integer linear model to determine the optimal transmission range, time-to-live counter and proportion of active modes (among those that are alive) for each time in $\mathcal{T}$. We label this as Problem $\mathbf{P}_2$.

$$(\mathbf{P}_2) \quad \max \quad \sum_{\tau=1}^{T} x_\tau \tag{3.10a}$$

$$\text{s.t.} \quad \sum_{\tau} \sum_{s \in \mathcal{N}} \sum_{a \in \mathcal{A}(s)} c_{sa} y_{sa}^{\tau} \leq \overline{b} \tag{3.10b}$$

$$\sum_{a \in \mathcal{A}(s)} y_{sa}^{\tau} = x_\tau, \ \tau \in \mathcal{T} \tag{3.10c}$$

$$x_{\tau+1} \leq x_\tau, \ \tau \in \mathcal{T} \setminus \{T\} \tag{3.10d}$$

$$m_{sa}^{\tau} \leq s y_{sa}^{\tau}, \ \tau \in \mathcal{T}, \ s \in \mathcal{N}, \ a \in \mathcal{A}(s) \tag{3.10e}$$

$$m_{sa}^{\tau} \geq \sum_{j \in J} k_{saj} z_j^{\tau} - s\left(1 - y_{sa}^{\tau}\right), \ \tau \in \mathcal{T}, \ s \in \mathcal{N}, \ a \in \mathcal{A}(s) \tag{3.10f}$$

$$m_{sa}^{\tau} \leq \sum_{j \in J} k_{saj} z_j^{\tau}, \ \ \tau \in \mathcal{T}, \ s \in \mathcal{N}, \ a \in \mathcal{A}(s) \tag{3.10g}$$

$$\sum_{a \in \mathcal{A}(s)} s y_{sa}^{1} = N \tag{3.10h}$$

$$\sum_{\substack{s \in \mathcal{N}, \\ a \in \mathcal{A}(s)}} s\, y_{sa}^{\tau} \leq \sum_{\substack{s \in \mathcal{N}, \\ a \in \mathcal{A}(s)}} (s y_{sa}^{\tau-1} - m_{sa}^{\tau-1}), \ \tau \in \mathcal{T} \setminus \{1\} \tag{3.10i}$$

$$b_1 = \overline{b}, \tag{3.10j}$$

$$b_\tau = b_{\tau-1} - \sum_{\substack{s \in \mathcal{N}, \\ a \in \mathcal{A}(s)}} c_{sa} y_{sa}^{\tau-1}, \ \tau \in \mathcal{T} \setminus \{1\} \tag{3.10k}$$

$$b_\tau \geq \sum_{j \in J} u_j z_j^{\tau}, \ \tau \in \mathcal{T} \tag{3.10l}$$

$$\sum_{j \in J} z_j^{\tau} = 1, \ \tau \in \mathcal{T} \tag{3.10m}$$

$$x_\tau \in \{0,1\}, \ b_\tau \in [0, \overline{b}], \ \tau \in \mathcal{T} \tag{3.10n}$$

$$y_{sa}^{\tau} \in \{0,1\}, \tau \in \mathcal{T}, \ s \in \mathcal{N}, \ a \in \mathcal{A}(s) \tag{3.10o}$$

$$z_j^{\tau} \in \{0,1\}, \tau \in \mathcal{T}, \ j \in J \tag{3.10p}$$

$$m_{sa}^{\tau} \in \{0,1,\ldots,s\}, \tau \in \mathcal{T}, \ s \in \mathcal{N}, \ a \in \mathcal{A}(s) \tag{3.10q}$$

The objective function (3.10a) of $\mathbf{P}_2$ is the same as (3.8a) in $\mathbf{P}_1$. For the sake of completeness, we also include constraint (3.10b) to account for the limit on remaining useful battery life. With constraint (3.10c), x_τ is forced to be 1 if and only if there is a feasible policy in the τth period. Constraint (3.10d) (which mirrors constraint (3.8g) of problem $\mathbf{P}_1$) ensures that the network satisfies the QoS and connectivity requirements in consecutive periods. Constraints (3.10e), (3.10f) and (3.10g) ensure that the number of node failures in the τth period is set. With constraints (3.10h) and (3.10i), node failures due to energy depletion are accounted for, and the number of alive nodes remaining in the network can be restricted. With constraints (3.10j) and (3.10k), the remaining available energy at a node in the τth period is determined. With constraints (3.10l) and (3.10m), z_j^τ is forced to be 1 to satisfy (3.9). The admissible ranges of variables are set in (3.10n) through (3.10q). Before presenting numerical results for problem $\mathbf{P}_2$, we first discuss a simplified version of $\mathbf{P}_1$ in Section 3.3.

3.3 A SPECIAL CASE

When all alive sensor nodes are active for all time periods, i.e., $p_\tau = 1$, for all $\tau \in \mathcal{T}$, we can solve a *single-period* model to determine the optimal transmission range and time-to-live counter for each period that maximizes the network's lifetime. In this section, we have the following assumptions for each τ:

A1. $\Delta(n_\tau, a_\tau)$ is nonincreasing in n_τ;

A2. $\Psi(n_\tau, a_\tau)$ is increasing in n_τ.

The former assumption can be explained by the revisiting effect, such that the probability that a query or an event agent revisits a node can be significant. This revisiting effect increases the proportion of time nodes are uninformed, the time to locate an informed node, and consequently, the proportion of failed queries (see [32]). As the number of active nodes increases, the revisiting effect is less pronounced. The latter assumption can be intuitively justified; however, we provide a sufficient condition using (3.7).

Proposition 3.3. *Assumption A2 holds when* $\sqrt{L} > 2r_\tau$ *and*

$$\frac{n_\tau \left(4 + 3\pi^2\right) r_\tau^2}{4L\pi} \geq 1. \tag{3.11}$$

Proof. First note that when (3.11) holds,

$$1 \leq \frac{n_\tau \left(4+3\pi^2\right) r_\tau^2}{4L\pi} = \frac{n_\tau r_\tau^2}{L\pi} + \frac{3n_\tau \pi r_\tau^2}{4L} < \frac{n_\tau \pi r_\tau^2}{4L} + \frac{3n_\tau \pi r_\tau^2}{4L} = \frac{n_\tau \pi r_\tau^2}{L}.$$

Therefore, using (3.7),

$$\begin{aligned}\frac{\partial \Psi(n_\tau, a_\tau)}{\partial n_\tau} = \exp &\left[-e^{-\frac{n_\tau \left(4+3\pi^2\right) r_\tau^2}{4L\pi}} n_\tau \left(1 - \frac{\left(\sqrt{L} - 2r_\tau\right)^2}{L}\right) - \frac{e^{-\frac{n_\tau \pi r_\tau^2}{L}} n_\tau \left(\sqrt{L} - 2r_\tau\right)^2}{L} \right] \\ &\times \left(-e^{-\frac{n_\tau \left(4+3\pi^2\right) r_\tau^2}{4L\pi}} \left(1 - \frac{\left(\sqrt{L} - 2r_\tau\right)^2}{L}\right) \left(1 - \frac{n_\tau \left(4+3\pi^2\right) r_\tau^2}{4L\pi}\right) \right. \\ &\left. -e^{-\frac{n_\tau \pi r_\tau^2}{L}} \left(\frac{\left(\sqrt{L} - 2r_\tau\right)^2}{L}\right) \left(1 - \frac{n_\tau \pi r_\tau^2}{L}\right) \right) \geq 0.\end{aligned}$$

□

Therefore, $\Psi(n_\tau, a_\tau)$ is increasing in n_τ.

Note that the condition $\sqrt{L} > 2r_\tau$ is usually met in practice since the sensor transmission range is relatively small compared to the sensor field dimensions. We can interpret (3.11) as follows: If the expected node degree of the nodes in $\mathcal{R}_2$ is at least unity, then assumption A2 holds.

By these assumptions, the optimal solution of $\mathbf{P}_1$ is obtained when the number of active nodes for each $\tau \in \mathcal{T}$ is maximized. This can be achieved by sequentially solving the following energy minimization problem (the *single-period* model) for each time $\tau \in \mathcal{T}$.

$$\begin{aligned}(\mathbf{P}_3) \quad \min \quad & c(n_\tau, \{r_\tau, \ell_\tau, 1\}) \\ \text{s.t.} \quad & \Delta(n_\tau, \{r_\tau, \ell_\tau, 1\}) \leq \varphi \\ & \Psi(n_\tau, \{r_\tau, \ell_\tau, 1\}) \geq \zeta \\ & \{r_\tau, \ell_\tau, 1\} \in \mathcal{A}.\end{aligned}$$

Note that after solving the above model for each $\tau \in \mathcal{T}$, both the number of alive nodes and available energy must be updated as follows:

$$\begin{aligned} n_{\tau+1} &= \lfloor n_\tau - f_\tau(n_\tau, a_\tau, b_\tau) n_\tau \rfloor, && \tau \in \mathcal{T} \setminus \{T\} \\ b_{\tau+1} &= b_\tau - c(n_\tau, a_\tau), && \tau \in \mathcal{T} \setminus \{T\}. \end{aligned}$$

The optimal solution of problem $\mathbf{P}_3$ can be obtained by exhaustive enumeration. However, we propose an algorithm which is significantly more efficient than exhaustive enumeration. In addition to assumptions A1 and A2 we impose the following assumptions for each τ:

A3. $c_\tau(n_\tau, a_\tau)$ is continuous and increasing in r_τ;

A4. $\Delta(n_\tau, a_\tau)$ is decreasing in ℓ_τ;

A5. $\Delta(n_\tau, a_\tau)$ is decreasing in r_τ;

A6. $\Psi(n_\tau, a_\tau)$ is increasing in r_τ.

Unfortunately, assumptions A1 through A6 are difficult to prove analytically because the structures of $c_\tau(n_\tau, a_\tau)$, $\Delta(n_\tau, a_\tau)$ and $\Psi(n_\tau, a_\tau)$ are unknown. However, after extensive empirical testing using (3.2), (2.21) and (3.6), we have failed to identify a case in which these assumptions are violated. We propose the following algorithm to solve $\mathbf{P}_3$:

Step 0: *Initialize*

$\tau := 1$; $b_\tau := \overline{b}$; $r_0 := 0$;

Step 1: *Check for optimal solution*

$r_{\min} := \min\{r : r_{\tau-1} \le r \le \overline{r},\ \Psi(n_\tau, \{r, 0, 1\}) \ge \zeta\}$;

$\ell' := \arg\min\{c(n_\tau, \{r_{\min}, \ell, 1\}) : \ell \in \{1, \ldots, n_\tau - 1\},\ \Delta(n_\tau, \{r_{\min}, \ell, 1\}) \le \varphi\}$;

If $\exists\, \ell'$, then

$\ell_\tau := \ell'$; $r_\tau := r_{\min}$;

$c^* := c(n_\tau, \{r_\tau, \ell_\tau, 1\})$;

Go to *Step 3*.

Else

$\ell := n_\tau - 1$; $c^* := M$;

Go to *Step 2*.

Step 2: *Search for optimal solution*

$r' := \arg\min\{c(n_\tau, \{r, \ell, 1\}) : r_{\min} \le r \le \overline{r},\ \Delta(n_\tau, \{r, \ell, 1\}) \le \varphi\}$;

If $\exists\, r'$, then

If $c(n_\tau, \{r', \ell, 1\}) < c^*$

$\ell^* := \ell$; $r^* = r'$;

$c^* := c(n_\tau, \{r', \ell, 1\})$;

$r_{\min} := r'$; $\ell = \ell - 1$;

Go to *Step 2*.

Else

$\ell_\tau := \ell^*$; $r_\tau := r^*$;

Go to *Step 3*.

Step 3: *Check for feasibility*

If $c^* \leq b_\tau$, then

$x_\tau := 1;\ b_{\tau+1} := b_\tau - c^*;$

$\tau := \tau + 1;$

Go to *Step 1*.

Else

$x_\tau := 0,$

End.

In Step 1 of the algorithm, $r_{\min}$ is fixed considering the connectivity constraint, and if $c(n_\tau, \{r_{\min}, \ell, 1\})$ is convex in ℓ, then a simple bisection algorithm can be used to search for an ℓ that satisfies the QoS constraint. If there exists an ℓ' for given $r_{\min}$, then the solution is optimal due to assumptions A3 and A6.

On the other hand, if there does not exist an ℓ', then we set $\ell = n_\tau - 1$, because in this case, $\Delta(n_\tau, \{r, \ell, 1\})$ is minimized for any given r by A4. Therefore, if there exists a feasible solution for the problem, it can be obtained at $\ell = n_\tau - 1$. In Step 2, a simple bisection algorithm can be used to search for an r that satisfies both the QoS and connectivity constraints. In this step, ℓ is decremented by 1. By A4 and A5, in order to maintain feasibility, we have $r_{\min} \leq r$. Note that ℓ is decremented until there is no feasible (ℓ, r) that simultaneously satisfies the QoS and connectivity constraints. In this algorithm, we make use of the structural properties assumed in A3–A6, hence, set of admissible values for r is made smaller at each iteration of Step 2. Therefore, the performance of the algorithm is improved as compared to exhaustive enumeration.

3.4 COMPUTATIONAL EXPERIMENTS

3.4.1 Description of Experiments

In this section, we explore the effect of the optimal choices of transmission range, time-to-live counter and active/sleep schedules on WSN lifetime. To see the significance of parameter settings for each period, we compare four cases on a network with N nodes ($N \in \{600, 700, 800, 900, 1000\}$)

deployed randomly on a two-dimensional sensor field with node density 6.25×10^{-3} nodes/m^2 (cf. [25]). The four cases we consider are as follows:

1. **Optimized/Optimized (OO)**: optimized transmission range, active/sleep decisions and resource replication level for each period of the planning horizon;
2. **Fixed/Optimized (FO)**: fixed transmission range, optimized active/sleep decisions and resource replication level for each period of the planning horizon;
3. **Optimized/Fixed (OF)**: optimized transmission range and resource replication level for each period of the planning horizon and sensor nodes are always active;
4. **Fixed/Fixed (FF)**: fixed transmission range, sensor nodes are always active and optimized resource replication level.

For each of these four cases, the optimal time-to-live counter (ℓ) is selected for each time period. Although scenarios OO, FO and OF are termed optimal, it is important to note that the optimal decisions for each time period are made *a priori* – not dynamically. Nonetheless, these decisions do, in fact, account for the time-varying topology of the network due to sensor node battery depletion. In cases FO and FF, the transmission range is fixed at its minimum value that is chosen when the transmission range is optimized for each time period.

In the test instances, we assume each node initially has 10 Joules (J) of battery energy (cf. [25]). The node energy expenditures in the τth period are assumed to be i.i.d. truncated normal random variables (Tr-N) with mean $c(s_\tau, a_\tau)$ and standard deviation σ. Typical values for currently available radio transceivers are $e_t = 50 \times 10^{-9}$ J/bit, $e_d = 100 \times 10^{-12}$ J/bit/m^2 (cf. [42, 116]), and $\eta = 2$ (path-loss coefficient). The length of each period in the optimization model is one unit of time (e.g., one week). To analyze the effect of the variance of energy expenditure, we also compare the network lifetimes of four cases when $\sigma \in \{0.2, 0.4, \ldots, 2.0\}$. Larger dispersion of the energy expenditures values is likely to prevail due to a nonhomogeneous environment, or if some nodes serve as relays more frequently than others. For the test instances, the parameter values are summarized in Table 13. In all test instances, the event and query lifetimes follow triangular and uniform distributions, respectively.

The analytical approximations were coded in the C programming language and executed in Microsoft® Visual Studio® 2008 on a personal computer equipped with an Intel® Core™ 2 Duo CPU operating at 3.00GHz with 2.00 GB of RAM.

Table 11: Parameter values for the test instances.

Parameter description	Value
Transmitter's exponential transmission rate	5.000
Poisson rate of locally-witnessed events	0.012
Poisson rate of locally-generated queries	0.050
Event lifetime distribution	$\text{Tri}(0.1, 10.0, 19.9)$
Query lifetime distribution	$\text{U}(0.1, 9.9)$
Energy expenditure during the τth period	$\text{Tr-N}(c_\tau(n_\tau, a_\tau), \sigma^2)$
Proportion of query failure limit (φ)	0.025
Minimum probability of being connected (ζ)	0.99
Planning horizon ($\mathcal{T}$)	$\{1, 2, \ldots, 20\}$

For these test instances, the optimal solution of $\mathbf{P}_2$ is an approximate solution because both $\Delta(n_\tau, a_\tau)$ and $\Psi(n_\tau, a_\tau)$ are evaluated using approximations given in (2.21) and (3.6), respectively. Therefore, for computational expedience, $\mathbf{P}_2$ can be solved by considering only a subset of the policy alternatives without significant effect on the quality of solutions obtained. Consequently, we allow s_τ to assume only one of 20 integer values between 0 and N. Similarly, n_τ may assume one of 10 integer values between 0 and s_τ.

3.4.2 Results and Discussion

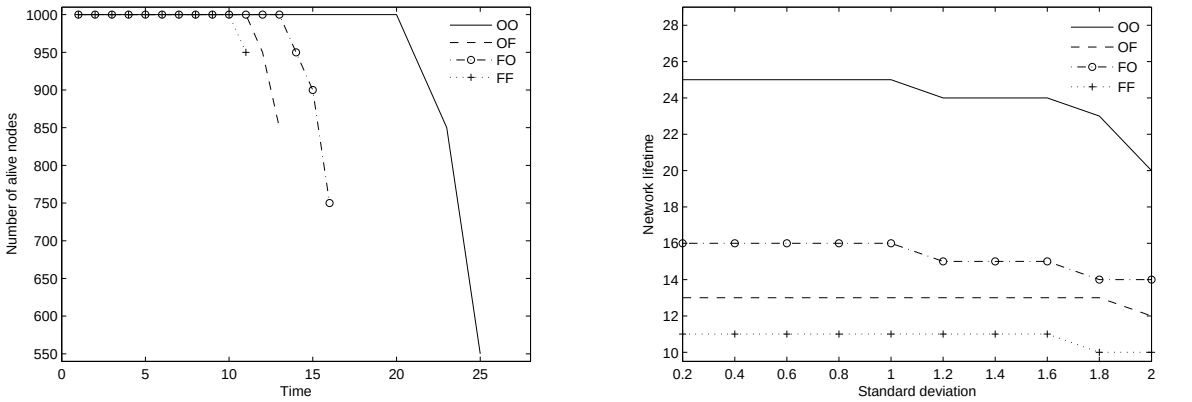
Figure 17a shows the effect of adjusting transmission range and sleep schedules on the number of alive nodes over time when $N = 1000$. When the transmission range and active/sleep decisions are fixed over time, the number of alive nodes in the network decreases rapidly. When either one of transmission range and active/sleep decisions is optimized for each period, the node failure rate is smaller than the case when the decisions are static. On the other hand, when the transmission range and the active/sleep schedules are optimized, the network lifetime increases, because nodes fail with a smaller rate over time. Additionally, when the transmission range and sleep schedules are optimized considering node failures, the network satisfies the QoS and connectivity requirements even after many nodes fail. Similar results are observed when $N \in \{600, 700, 800, 900\}$.

Table 12: Average lifetime over 10 test instances.

N	OO	FO	OF	FF
600	27.4	17.1	12.9	10.9
700	27.8	13.6	13.3	10.5
800	18.6	15.3	10.8	10.8
900	26.9	13.4	12.9	10.5
1000	24.0	15.3	12.9	10.8

The results summarized in Table 12 are the average lifetimes of 10 cases, where the standard deviation varies from 0.2 to 2.0. This table illustrates the significant effect of optimized decisions and reveals that for different sizes of networks, setting optimal transmission range and active/sleep decisions increases the average network lifetime.

Figure 17b reveals some very interesting results. First, we observe that making the transmission range and active/sleep decisions in each period of the planning horizon is superior to the other cases in terms of the average maximum network lifetime for almost all of the standard deviation values. Second, the network lifetime decreases as the standard deviation of the energy expenditure



(a) Number of alive nodes ($N = 1000$, $\sigma = 1.0$). (b) Network lifetime as a function of σ ($N = 1000$).

Figure 17: Effect of optimal decisions on network lifetime.

increases. One possible explanation for the reduced network lifetime is that, as the variance of energy expenditure increases, a greater proportion of nodes can experience failure times that are significantly smaller than the mean failure time, thereby increasing the instances in which the connectivity and/or QoS constraints are violated early in the planning horizon.

4.0 CLUSTER HEAD LOCATION AND RELOCATION IN MOBILE WSNS

One strategy for dealing with limited energy storage in WSNs is to allow the sensor nodes to aggregate sensed data at a particular node (or set of nodes) in the network known as *cluster heads*. A cluster head typically possesses a larger energy supply and larger transmission range than other nodes in the network (see [112]) and is responsible for aggregating and disseminating data received from sensors within its own cluster. Therefore, two important design issues emerge when cluster heads are used: (1) where should cluster heads be located in a particular region; and (2) how should the sensor nodes be assigned to cluster heads, given that the positions of the sensors evolve dynamically over time? This problem is further complicated by the fact that communication links between sensor nodes and their cluster heads may be destroyed due to a harsh operating environment or attacks on the network by an adversary (e.g., in a military theater). However, by optimally clustering the sensor nodes and dynamically locating cluster heads, it is possible to ensure that each sensor is within a single hop of at least one cluster head.

The problem of clustering sensor nodes in a WSN has been analyzed from a variety of perspectives including load balancing, fault-tolerance, connectivity, maximizing network lifetime and cluster count, to name only a few. Some useful survey papers related to clustering in WSNs include Liu and Shi [70], Kumar et al. [67], Younis et al. [112] and Abbasi et al. [1]. Liu and Shi [70] surveyed popular clustering algorithms and categorized them into three groups: cluster head election algorithms, cluster formation and data transmission. In this chapter, the most commonly discussed clustering approaches are classified as: (1) decentralized (distributed) algorithms, (2) centralized algorithms, or (3) optimization methods.

Low Energy Adaptive Clustering Hierarchy (LEACH) [51] is one of the most popular clustering algorithms that uses a distributed algorithm in which cluster heads are selected and rotated randomly to balance the energy expenditure among the sensors in the network. A node becomes a cluster head with a certain probability, which is assigned by the algorithm. LEACH assumes

that all sensor locations are known and the number of clusters in the network is fixed *a-priori* in the algorithm. In the Hybrid Energy Efficient Distributed (HEED) clustering algorithm [111] and Time Delay Based Clustering (TDC) algorithm [117], cluster heads are selected based on their residual energy, and explicit sensor location information is not required. However, the optimal selection of cluster heads is not guaranteed. In HEED [111], intra-cluster communication costs – the energy expenditure due to communication from a sensor to its cluster head – are not considered, while Ding et al. [37] proposed DWEHC, a distributed clustering algorithm, that does consider the intra-communication costs. Yu et al. [115] also presented a clustering algorithm for large-scale networks that minimizes both intra- and inter-cluster communications. Xia and Vlajic [106] and Manisekaran et al. [74] developed algorithms wherein the similarity of sensor readings is used as the main clustering criterion to minimize the in-network data reporting traffic. Dimoskas et al. [36] considered the significance of a sensor with respect to its contribution in relaying messages as a metric for clustering. Koucheryavy and Salim [63] developed a combined criteria metric according to connectivity, coverage, mobility and residual energy and their distributed algorithm uses predicted values of their combined metric. Each of these techniques can be viewed as distributed algorithms that make decisions based on *local* observations. However, they have extensive data collection and storage requirements (see [28]).

By contrast, centralized algorithms have been proposed to reduce the energy expenditure in the network. LEACH-C [50] is a centralized version of LEACH wherein the rotation of cluster heads is controlled by a base station. LEACH-C is superior to LEACH in that its energy consumption is less (based on computational results). Ci et al. [28] proposed a centralized clustering algorithm wherein cluster heads are rotated based on mining the sensor energy data without localization information. Khan et al. [59] proposed Multiple Parameter-based Clustering (MPC) which makes clustering decisions based on the residual energy of sensors, proximity to the base station, and latency of data to the base station. Gupta et al. [46] proposed a fuzzy-logic based clustering algorithm that considers the residual energy, the number of neighboring nodes and centrality of the nodes.

Relevant to our work here are models that apply both exact and heuristic optimization techniques to the optimal clustering problem in WSNs. Slama et al. [98] developed an optimization model that maximizes the network lifetime by balancing energy expenditure over the sensors' activities. Islam et al. [55] proposed heuristic methods for the same problem. Krivitski et al. [30] proposed a facility location-based heuristic algorithm for sensor networks where resources can be placed in any of the k out of m possible locations. Furuta et al. [41] formulated the WSN clus-

tering problem as an uncapacitated facility location problem and incorporated the residual energy of sensor nodes. Their objective is to extend the network lifetime by finding the optimal number of cluster heads and simultaneously selecting the cluster head candidates. Youssef et al. [113] proposed a heuristic that considers cluster-overlapping in addition to connectivity and coverage. Their heuristic provides the set of cluster heads such that every node in the network is within some distance k from a cluster head. Aioffi et al. [3] proposed algorithms to minimize message delivery latency by considering topology constraints that reduce energy consumption. Shanbehzadeh et al. [93] developed a genetic algorithm and particle swarm-based heuristic algorithm to determine the number of clusters, to elect the cluster heads and to cluster the members. Sevgi and Kocyigit [92] developed a heuristic to determine the size of clusters and initial energy level of sensors to maintain network lifetime and coverage requirements.

Perhaps most relevant to our model is the one described by Patel et al. [84] which considers the optimal clustering of nodes in an ad hoc network with mobile nodes and unreliable communication links. They proposed a mixed integer linear programming (MILP) model to maximize the expected data coverage minus the cluster head reassignment costs. They developed a column-generation heuristic to solve the problem and suggested several interesting extensions. Their models, however, assume that sensor locations (or at least proximities to cluster heads) are known *a priori* and do not consider the time evolution of sensor locations in the region.

The primary objective of this chapter is to formulate and solve optimization problems that seek to maximize the demand coverage and minimize the costs of locating, and relocating, cluster heads in a WSN with mobile nodes and unreliable links. Specifically, we will determine the cluster head locations and assignment of sensor nodes to particular cluster heads over a finite planning horizon. Due to node mobility and unreliability of the links, we also consider the optimal timing of sensor location updates to relocate cluster heads and maintain connectivity of the network. Specifically, cluster heads are relocated to ensure that each sensor node can send information to a cluster head in a single hop. The model can be viewed as a generalization of the model in [84] in that we allow the position of sensor nodes to evolve randomly over time, and we explicitly determine the best time(s) at which to update the sensor locations. This dynamic updating of information is shown to improve demand coverage and reduce the network traffic associated with updating locations in each period of the planning horizon. Perhaps most important is the fact that we are able to linearize our model and solve it with relative ease using a commercial solver.

The remainder of the chapter is organized as follows. Section 4.1 introduces essential notation and provides the initial formulation for the problem of locating and relocating cluster heads. Section 4.2 considers an extension of the main model to consider the problem of optimally timing updates while Section 4.3 provides a summary of computation experiments.

4.1 MAIN PROBLEM FORMULATION

Because this chapter addresses a different class of problems than those addressed in Chapters 2 and 3, we first review and introduce some new notation. Consider a WSN with N sensor nodes that are initially randomly distributed in a subset of Euclidean two-dimensional space; however, the nodes are mobile and their locations evolve according to a continuous-time, continuous-state stochastic process (described in greater detail later). For simplicity, we consider only a square sensor field R ($R \subseteq \mathbb{R}^2$), but the same model can be used if $R \subseteq \mathbb{R}^3$. Let $\mathcal{N} = \{1, 2, \ldots, N\}$ be the set of sensor nodes and assume that each node in the network uses the same finite transmission range r (in meters). Additionally, there are finitely many points in R that are candidate locations for cluster heads. Let $\mathcal{K} = \{1, 2, \ldots, K\}$ be the set of labels for potential cluster head locations. At most n ($n \leq K$) of the K potential locations can be selected to host cluster heads. Figure 18 graphically depicts this scenario where nodes marked by $\otimes$ represent potential cluster head locations and the black dots represent the mobile sensor nodes.

The planning horizon consists of T ($T < \infty$) periods, each of equal duration, i.e. the planning horizon is the finite set $\mathcal{T} = \{1, 2, \ldots, T\}$, where each element of $\mathcal{T}$ represents a decision epoch, and the time between two epochs is referred to as a period. During each period $t \in \mathcal{T}$, sensor node $j \in \mathcal{N}$ generates a deterministic demand d_j that must be satisfied by its cluster head. For example, this demand might represent a summary of sensed data that must be reported to the cluster head (e.g., the average temperature reading observed during period j). If a sensor node is within range of at least one cluster head, the demand of that node is said to be *covered*. It is assumed that sensors only report to cluster heads. That is, there is no routing involved – one sensor does not act as a relay for other sensors in this model. The following notation will be used in the main model:

Parameters:

- d_j: the per period demand generated by sensor $j \in \mathcal{N}$;
- p: the probability of link failure between any cluster head and sensor $(0 < p < 1)$;
- C: the unit cost of placing a single cluster head (or changeover cost);
- Z_{ijt}: an indicator random variable for the proximity of cluster head i to sensor j in period t,

$$Z_{ijt} = \begin{cases} 1, & \text{if cluster head location } i \text{ is within range of sensor } j \text{ in period } t, \\ 0, & \text{otherwise.} \end{cases}$$

In the main model, the objective is to locate and relocate cluster heads and to assign sensors to cluster heads during each period. Therefore, the decisions variables are as follows:

Decision Variables:

- x_{it}: a binary variable for cluster head assignment at a candidate location,

$$x_{it} = \begin{cases} 1, & \text{if a cluster head is placed at location } i \text{ in period } t, \\ 0, & \text{otherwise;} \end{cases}$$

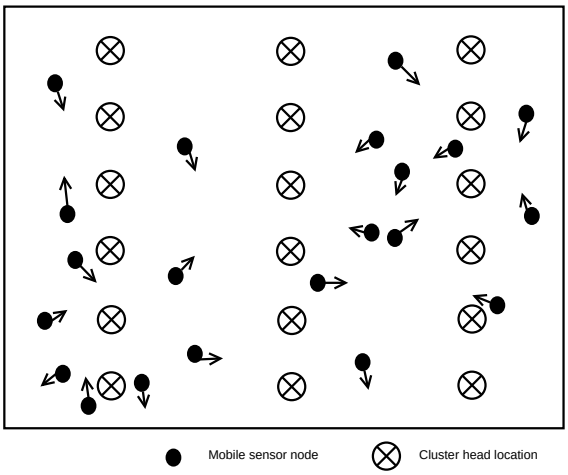


Figure 18: An illustrative example of a WSN with mobile nodes and cluster head locations.

- υ_{jkt}: a binary variable indicating the sensor coverage by at least k cluster heads in period t,

$$\upsilon_{jkt} = \begin{cases} 1, & \text{if sensor } j \text{ is covered by at least } k \text{ cluster heads in period } t, \\ 0, & \text{otherwise;} \end{cases}$$

- w_{it}: a binary variable describing whether cluster head i is relocated at period t,

$$w_{it} = \begin{cases} 1, & \text{if a cluster head is located at } i \text{ in period } t-1 \text{ and not in period } t, \\ 1, & \text{if a cluster head is located at } i \text{ in period } t \text{ and not in period } t-1, \\ 0, & \text{otherwise.} \end{cases}$$

We propose the following model to maximize the demand the coverage and minimize the relocation costs by locating and relocating cluster heads.

$$(\mathbf{P}_4) \quad \max \sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{N}}\sum_{k=1}^{K}(1-p)p^{k-1}d_j\upsilon_{jkt} - C\sum_{i\in\mathcal{K}}\sum_{t\in\mathcal{T}} w_{it} \tag{4.1}$$

$$\text{s.t.} \quad \mathbb{P}\left(\sum_{k=1}^{K}\upsilon_{jkt} - \sum_{i\in\mathcal{K}} Z_{ijt}\, x_{it} > \psi\right) \le \kappa, \qquad j\in\mathcal{N},\, t\in\mathcal{T} \tag{4.2}$$

$$\sum_{i\in\mathcal{K}} x_{it} \le n, \qquad t\in\mathcal{T} \tag{4.3}$$

$$w_{it} \ge x_{it-1} - x_{it}, \qquad i\in\mathcal{K},\, t\in\mathcal{T}\setminus\{1\} \tag{4.4}$$

$$w_{it} \ge x_{it} - x_{it-1}, \qquad i\in\mathcal{K},\, t\in\mathcal{T}\setminus\{1\} \tag{4.5}$$

$$x_{it}\in\{0,1\},\, w_{it}\ge 0, \qquad i\in\mathcal{K},\, t\in\mathcal{T} \tag{4.6}$$

$$\upsilon_{jkt}\in\{0,1\}, \qquad j\in\mathcal{N},\, k\in\mathcal{K},\, t\in\mathcal{T} \tag{4.7}$$

The objective function (4.1) represents the expected demand covered by cluster heads minus the total relocation costs. The term $\sum_{k=1}^{K}(1-p)p^{k-1}d_j\upsilon_{jkt}$ represents the expected demand coverage of sensor j in period t, considering the link quality. Constraint (4.2) ensures that if sensor j is covered by k' cluster heads at time t, then each of the variables $\upsilon_{j1t}, \upsilon_{j2t}, \ldots, \upsilon_{jk't}$ is assigned a value of 1 since the objective function contains the term υ_{jkt}. Note in constraint (4.2) that Z_{ijt} is a random variable describing the connectivity of cluster head i and sensor j at time t. Here, ψ ($\psi > 0$) can be viewed as a "small" amount of infeasibility, and κ ($0 \le \kappa \le 1$) is a user-specified probability that bounds the likelihood that the chance constraint is violated. Larger values of κ serve to relax the problem but may not yield high-quality solutions. On the other hand, when $\kappa = 0$,

there is no chance of violating the constraint, yielding the most conservative solution. Constraint (4.3) ensures that the maximum number of cluster heads to be located cannot exceed n during any period. Constraints (4.4) and (4.5) determine the cluster head relocations and force w_{it} to be 1 if there is a change in cluster head location i in terms of cluster head assignments at time t. The admissible values of decision variables are given by constraints (4.6) through (4.7).

The mathematical programming formulation ($\mathbf{P}_4$) can be viewed as a generalization of the model presented by Patel et al. [84]. Specifically, we consider the stochastic evolution of sensor locations over time rather than assuming the positions are known at all times. Therefore, rather than specifying a deterministic constraint for the connectivity of sensors and cluster heads, we use a probabilistic constraint (namely (4.2)) that allows us to bound the likelihood that the connectivity requirement is violated. If the sensor locations are known at the start of each time period in the planning horizon, then (4.2) becomes a deterministic constraint, making $\mathbf{P}_4$ equivalent to the model proposed by Patel et al. [84].

Now, constraint (4.2) requires the distribution of $\sum_{i\in\mathcal{K}} Z_{ijt}\, x_{it}$, which cannot be easily derived in general. But if the distribution function of Z_{ijt} can be obtained, it can be used to rewrite (4.2) in a deterministic form that depends on the parameter κ. We next present an equivalent representation of $\mathbf{P}_4$ by first defining the following additional variables and parameters:

- z_{ijt}: a binary variable describing the connectivity of cluster head i and sensor j in period t,

$$z_{ijt} = \begin{cases} 1, & \text{if a cluster head at location } i \text{ is within range of sensor } j \text{ at time } t, \\ 0, & \text{otherwise;} \end{cases}$$

- y_{jt}: a binary variable describing the coverage of sensor j in period t,

$$y_{jt} = \begin{cases} 1, & \text{if sensor } j \text{ is covered by at least 1 cluster head with at least probability } \kappa \text{ at time } t, \\ 0, & \text{otherwise;} \end{cases}$$

- $\mathcal{K}_{jt} \in \mathcal{K}$: the set of cluster heads within range of sensor j at time t,

$$\mathcal{K}_{jt} = \{i \in \mathcal{K} : \mathbb{P}(Z_{ijt} = 1) > \kappa\};$$

- q_{jt}: probability that sensor j is covered by at least 1 cluster head and the link is not failed.

Let Q_{ijt} be an indicator (random) variable describing the status of the link between sensor j and cluster head i at time t. That is, $Q_{ijt} = 1$ if the link between cluster head i and sensor j is failed and 0, otherwise. Note that $\mathbb{P}(Q_{ijt} = 1) = p$. Proposition 4.1 shows how to compute the time-dependent probabilities, q_{jt}.

Proposition 4.1. *If the random variables* $\{Q_{ijt} : i \in \mathcal{K},\, j \in \mathcal{N},\, t \in \mathcal{T}\}$ *are mutually independent, then*

$$1 - q_{jt} = \prod_{i \in \mathcal{K}_{jt}} \left(\mathbb{P}(Z_{ijt} = 1)p + \mathbb{P}(Z_{ijt} = 0)\right)$$

for each $j \in \mathcal{N}$ *and* $t \in \mathcal{T}$.

Next, our aim is to make the model deterministic. To that end, let $\mathbb{1}(x)$ be an indicator function that assumes the value 1 if condition x holds and 0, otherwise. Then the probabilistic constraint (4.2) can be replaced by

$$z_{ijt} - \mathbb{1}(\mathbb{P}(Z_{ijt} = 1) > \kappa)\, x_{it} \leq 0.$$

The revised model is problem $\mathbf{P}_5$:

$$(\mathbf{P}_5) \quad \max \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{N}} q_{jt} d_j y_{jt} - C \sum_{i \in \mathcal{K}} \sum_{t \in \mathcal{T}} w_{it} \tag{4.8}$$

$$\text{s.t. } z_{ijt} - \mathbb{1}\left(\mathbb{P}(Z_{ijt} = 1) > \kappa\right) x_{it} \leq 0, \qquad i \in \mathcal{K},\, j \in \mathcal{N},\, t \in \mathcal{T} \tag{4.9}$$

$$y_{jt} - \sum_{i \in \mathcal{K}} z_{ijt} \leq 0, \qquad j \in \mathcal{N},\, t \in \mathcal{T} \tag{4.10}$$

$$y_{jt} \in \{0, 1\},\ z_{ijt} \in \{0, 1\}, \qquad j \in \mathcal{N},\, t \in \mathcal{T} \tag{4.11}$$

$$(4.3),\ (4.4),\ (4.5),\ (4.6). \tag{4.12}$$

As in problem $\mathbf{P}_4$, the objective function (4.8) represents the expected demand covered by cluster heads minus the relocation costs. Constraints (4.9) and (4.10) ensure that $y_{jt} = 0$ if no cluster head located within the transmission range of sensor j with probability at least as large as κ in period t. It is worth mentioning that the column generation algorithm proposed by Patel et al. [84] can be applied to solve problem $\mathbf{P}_5$ when $\mathbb{1}(\mathbb{P}(Z_{ijt} = 1) > \kappa)$ is known.

So far, we have assumed that the planning horizon is known *a priori*, and only the *initial* positions of the sensors are known. However, to improve WSN performance, the mobile sensors can broadcast their locations at the start of each period. This updating can be costly because (1) the additional network traffic serves to increase energy expenditure [104], and (2) covert sensor locations might be inadvertently revealed. The next subsection describes an extension of problems $\mathbf{P}_4$ and $\mathbf{P}_5$ to determine the optimal time to *next* update the sensor locations in order to maximize the expected demand coverage and minimize the location/relocation costs. Additionally, in the next subsection, we modify the objective function to prevent the double counting of coverage as the demand of a given sensor can be covered by multiple cluster heads.

4.2 OPTIMALLY TIMING LOCATION UPDATES

The optimal solutions of problems $\mathbf{P}_4$ and $\mathbf{P}_5$ yield myopic policies, i.e., they maximize the demand coverage and minimize location costs using only the initial sensor locations. However, the initial sensor locations may not be adequate for decision making in each of the subsequent periods of the planning horizon. In this section, we propose a non-myopic approach to relocate the sensors. The objective of non-myopic cluster head relocation is to determine optimal relocation decisions until the next time to update the sensor locations. The solution approach is repeated after all decisions have been executed. That is, at the beginning of period $t = 1$, the model will determine not only cluster head and sensor assignments, but also the optimal time t^* at which to next update the sensor locations. Subsequently, the problem can be solved sequentially starting with the updated sensor coordinates at time t^* until the end of the planning horizon. Updating the sensor locations at these optimal times results in increased expected demand coverage and lower relocation costs.

Define s_t as a binary variable describing the first time that the node locations are updated after time 0, that is,

$$s_t = \begin{cases} 1, & \text{if sensor locations are updated for the first time at time } t, \\ 0, & \text{otherwise,} \end{cases}$$

and let C' be the cost of updating the sensor locations. The new formulation is as follows:

$$(\mathbf{P}_6) \quad \max \left(\sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{N}} q_{jt} d_j y_{jt} - C \sum_{i\in\mathcal{K}} \sum_{t\in\mathcal{T}} w_{it} - C' \right) \cdot \sum_{t\in\mathcal{T}} \frac{s_t}{t} \tag{4.13}$$

$$\text{s.t.} \quad y_{jt} + \sum_{t'\le t} s_{t'} \le 1, \qquad j \in \mathcal{N},\, t \in \mathcal{T} \tag{4.14}$$

$$\sum_{t\in\mathcal{T}} s_t = 1, \tag{4.15}$$

$$s_t \in \{0,1\}, \qquad t \in \mathcal{T} \tag{4.16}$$

$$(4.3), (4.4), (4.5), (4.6), (4.9), (4.10), (4.11). \tag{4.17}$$

The objective function is identical to (4.8) except that here we maximize the average expected demand covered by cluster heads minus the relocation and updating costs from the prior updating time until the next updating time. Constraint (4.14) ensures that $y_{jt} = 0$ for $t \ge t^*$ where t^* is the next optimal updating time. This constraint is required because sensor locations must be updated at time t^* and $\mathbf{P}_6$ can be solved with the updated information for time periods $t \in \mathcal{T} \setminus \{1, 2, \ldots, t^* - 1\}$.

With constraint (4.15), only the next update time is determined. The sequential solution procedure is summarized below to explain how to obtain relocation decisions for time periods $t \in \mathcal{T}$.

Step 0: Initialize/update q_{jt}, $j \in \mathcal{N}$, $t \in \mathcal{T}$;

Step 1: Solve $\mathbf{P}_6$ to determine cluster head locations for $t \in \mathcal{T}$ and time to update, t^*;

Step 2: At time t^*, update $\mathcal{T} := \mathcal{T} \setminus \{1, 2, \ldots, t^* - 1\}$;

Step 3: If $|\mathcal{T}| > 1$, then

Go to *Step 0*;

Else

End.

Solving $\mathbf{P}_6$ is nontrivial due to the nonlinearity of the objective function. This complication can be avoided by linearizing the model and solving it using a commercial solver (e.g., CPLEX [29]). To that end, define the following two variables, both of which are continuous on $[0, 1]$:

$$u_{jt} = y_{jt} \sum_{m \in \mathcal{T}} \frac{s_m}{m}, \quad j \in \mathcal{N}, t \in \mathcal{T}; \tag{4.18}$$

$$v_{it} = w_{it} \sum_{m \in \mathcal{T}} \frac{s_m}{m}, \quad i \in \mathcal{K}, t \in \mathcal{T}. \tag{4.19}$$

Using these variables, we next present a mixed integer linear program (MILP) to determine the optimal cluster head locations and the location updating times over the planning horizon $\mathcal{T}$.

$$(\mathbf{P}_7) \quad \max \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{N}} q_{jt} d_j u_{jt} - C \sum_{i \in \mathcal{K}} \sum_{t \in \mathcal{T}} v_{it} - C' \sum_{t \in \mathcal{T}} \frac{s_t}{t} \tag{4.20}$$

$$\text{s.t. } u_{jt} - \sum_{t \in \mathcal{T}} \frac{s_t}{t} \leq 0, \qquad j \in \mathcal{N}, t \in \mathcal{T} \tag{4.21}$$

$$u_{jt} - y_{jt} \leq 0, \qquad j \in \mathcal{N}, t \in \mathcal{T} \tag{4.22}$$

$$w_{it} - v_{it} + \sum_{t \in \mathcal{T}} \frac{s_t}{t} \leq 1, \qquad i \in \mathcal{K}, t \in \mathcal{T} \tag{4.23}$$

$$v_{it} \in [0, 1], \qquad i \in \mathcal{K}, t \in \mathcal{T} \tag{4.24}$$

$$u_{jt} \in [0, 1], \qquad j \in \mathcal{N}, t \in \mathcal{T} \tag{4.25}$$

$$(4.3), (4.4), (4.5), (4.6), (4.9), \tag{4.26}$$

$$(4.10), (4.11), (4.14), (4.15), (4.16). \tag{4.27}$$

The objective function (4.20) of $\mathbf{P}_7$ is identical to that of problem $\mathbf{P}_6$ except that (4.20) is linear. Constraints (4.21) through (4.23) ensure that u_{jt} and v_{it} are assigned their defined values via (4.18) and (4.19). The admissible ranges of variables are set in (4.24) and (4.25). Most important, the linearized model $\mathbf{P}_7$ is amenable to solution by a commercial solver, such as CPLEX [29]. The next section highlights the advantages of updating the sensor locations over time when the sensors move throughout the region according to a two-dimensional Brownian motion process.

4.3 COMPUTATIONAL EXPERIMENTS

4.3.1 Description of Experiments

The problem $\mathbf{P}_7$ is generic in terms sensor locations and movements. Here, for illustrative purposes, we assume that each sensor's movements in the sensor field can be modeled as a two-dimensional Brownian motion (BM) process with drift. A continuous-time, continuous-state stochastic process, $\{X(t) : t \geq 0\}$, is called a one-dimensional BM process on $\mathbb{R}$ with drift μ if

$$X(t) = \mu t + \sigma B(t),$$

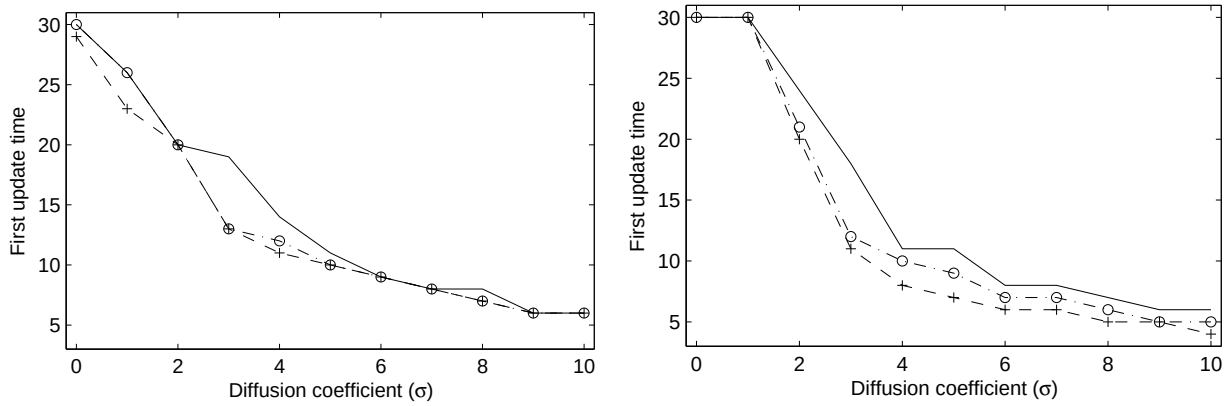
where μ is the drift parameter, σ is the diffusion coefficient and $B(t)$ denotes standard Brownian motion (i.e., $B(t) \sim N(0, t)$ for each $t \geq 0$). The parameter μ can be viewed as the mean behavior of the process whereas σ magnifies the random error term $B(t)$. BM with drift can be viewed as a linear motion which is observed when nodes move towards a target, and can be used to account for randomness due to either environmental or functional effects (cf. [38]). Extending this idea, $\{(B_x(t), B_y(t)) : t \geq 0\}$ is called a two-dimensional BM process if $\{(B_x(t)\}$ and $\{(B_y(t)\}$ are independent, and each component is a one-dimensional BM process on $\mathbb{R}$. We consider mobile sensors moving according to a two-dimensional BM process with drift in the sensor field. Denote the coordinates of sensor j at time t by $(X_j(t), Y_j(t))$ where,

$$X_j(t) = \mu t + \sigma B_x(t) \quad \text{and} \quad Y_j(t) = \mu t + \sigma B_y(t).$$

Two problem instances will be solved to illustrate the usefulness of updating sensor locations within the optimization framework of $\mathbf{P}_7$. Specifically, we present a small problem instance ($N = 25$) and a large problem instance ($N = 200$). The specific parameters for each case are summarized in Table 13.

Table 13: Parameter values for the test instances.

Parameter description	Small Instance	Large Instance
Planning horizon ($\mathcal{T}$)	$\{1, 2, \ldots, 30\}$	$\{1, 2, \ldots, 30\}$
Number of sensors (N)	25	200
Number of potential cluster head locations (K)	25	196
Maximum number of cluster heads (n)	5	10
Cluster head relocation cost (C)	5.00	5.00
Location updating cost (C')	100.00	100.00
Per period demand of sensor j (d_j)	$U(10, 20)$	$U(10, 20)$
Probability of link failure (p)	$\{0.1, 0.2, \ldots, 0.9\}$	$\{0.1, 0.2, \ldots, 0.9\}$
Coverage radius	13.00	13.00
Drift parameter (μ)	1	1
Diffusion coefficient (σ)	$\{0, 1, \ldots, 10\}$	$\{0, 1, \ldots, 10\}$



(a) Small instance.

(b) Large instance.

Figure 19: Comparison of first time time to update: (-) $p = 0.1$; (-·o) $p = 0.5$; (- - +) $p = 0.9$.

4.3.2 Results and Discussion

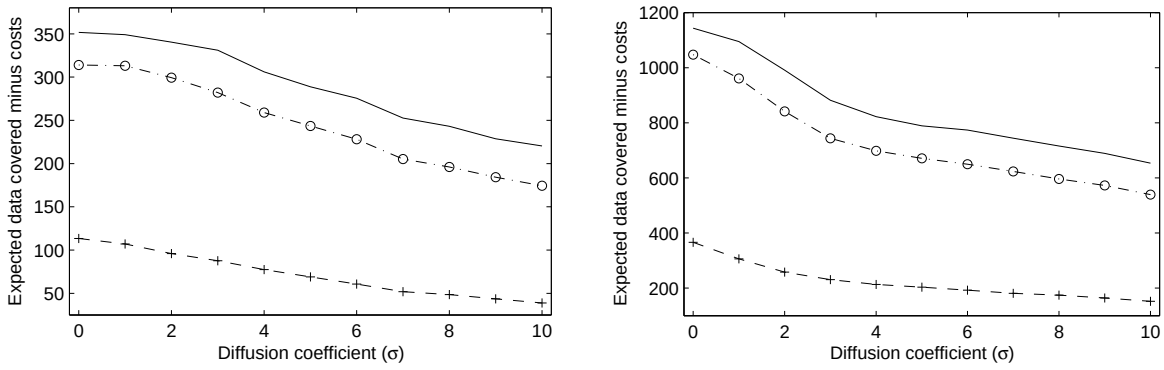
We plotted the optimal *first* time to update the sensor locations as a function of the diffusion coefficient (σ) in Figure 19. The behavior of this function is intuitive because as the diffusion parameter increases, the variation in the sensor movements increases, thereby reducing the predictability of the sensors' locations. The updating frequency is decreasing in the diffusion parameter for both the small and large problem instances. Additionally, we note that the optimal first update time decreases as the probability of link failure (p) increases. This is because the gradual decrease in the expected data coverage becomes more pronounced over time. Therefore, node locations need to be updated more frequently to ensure that the rate of total data coverage minus the costs is maximized. These conclusions are further corroborated by Tables 14 and 15.

Table 14: Summary of results for small problem instance.

p	σ	Solution time (s)	Objective value	Optimal update time
0.1	0	50.86	351.74	30
0.1	1	64.95	349.01	26
0.1	2	83.37	340.49	20
0.1	3	47.16	338.08	19
0.1	4	29.80	281.46	14
0.1	5	21.92	311.20	11,21
0.1	6	13.98	294.95	9,16,24
0.1	7	13.90	284.31	8,15,22
0.1	8	8.50	272.99	8,15,22
0.1	9	6.96	253.70	6,11,17,23
0.1	10	4.52	239.89	6,11,17,22
0.5	0	53.40	314.00	30
0.5	1	67.10	313.03	26
0.5	2	83.51	299.43	20
0.5	3	46.94	252.84	13
0.5	4	27.60	267.61	12,24
0.5	5	18.44	255.31	10,20
0.5	6	13.10	238.94	9,16,24
0.5	7	13.46	225.13	8,15,22
0.5	8	8.22	212.38	7,14,20
0.5	9	6.82	196.40	6,11,17,23
0.5	10	4.60	182.45	6,11,17,22
0.9	0	61.68	113.45	30
0.9	1	69.94	102.47	23
0.9	2	68.94	95.22	20
0.9	3	46.27	89.64	13
0.9	4	27.41	77.59	11,23
0.9	5	19.55	69.93	10,20
0.9	6	13.20	62.64	9,16,24
0.9	7	12.36	56.49	8,15,22
0.9	8	8.02	50.78	7,13,19
0.9	9	6.44	45.03	6,11,17,23
0.9	10	4.46	39.87	5,9,14,19,23

Table 15: Summary of results for large problem instance.

p	σ	Solution time (s)	Objective value	Optimal update time
0.1	0	1371.20	1143.36	30
0.1	1	1372.45	1094.84	30
0.1	2	1317.68	1004.25	24
0.1	3	1257.17	956.47	18
0.1	4	972.46	914.89	11, 22
0.1	5	651.44	861.12	11,19
0.1	6	427.51	824.48	8,15,23
0.1	7	299.49	790.13	8,14,19
0.1	8	258.32	755.86	7,13,18,24
0.1	9	253.52	718.91	6,11,16,21
0.1	10	158.12	677.61	6,10,15,19,24
0.5	0	1364.40	1047.35	30
0.5	1	1370.05	961.05	30
0.5	2	1284.22	875.53	21
0.5	3	1202.54	716.01	12
0.5	4	960.64	745.77	10,17
0.5	5	637.28	717.89	9,15,21
0.5	6	420.40	672.46	7,13,18
0.5	7	294.16	646.36	7,12,17,22
0.5	8	252.79	613.68	6,11,16,21
0.5	9	252.10	571.51	5,9,13,17,21
0.5	10	155.28	543.28	5,9,13,17,21
0.9	0	1414.03	366.32	30
0.9	1	1358.54	306.39	30
0.9	2	1275.22	271.14	20
0.9	3	1201.55	244.98	11,21
0.9	4	948.77	225.32	8,15,21
0.9	5	625.76	205.33	7,13,18,25
0.9	6	416.81	193.89	6,12,16,21
0.9	7	291.43	184.32	6,11,16,20,25
0.9	8	250.37	170.79	5,9,14,18,23
0.9	9	249.67	162.24	5,9,14,18,23
0.9	10	155.19	150.82	4,8,11,14,18,22,26



(a) Small problem instance.

(b) Large problem instance.

Figure 20: Comparison of objective values: (-) $p = 0.1$; (-·o) $p = 0.5$; (- - +) $p = 0.9$.

Figure 20 shows the effect of sensor movement and link failure probability on the expected data coverage minus the costs. As the diffusion parameter increases, the objective value decreases because, as time progresses, the likelihood that a cluster head is within the range of a particular sensor is decreasing. Figure 20 also shows that the value of the objective function decreases as the probability of link failure increases.

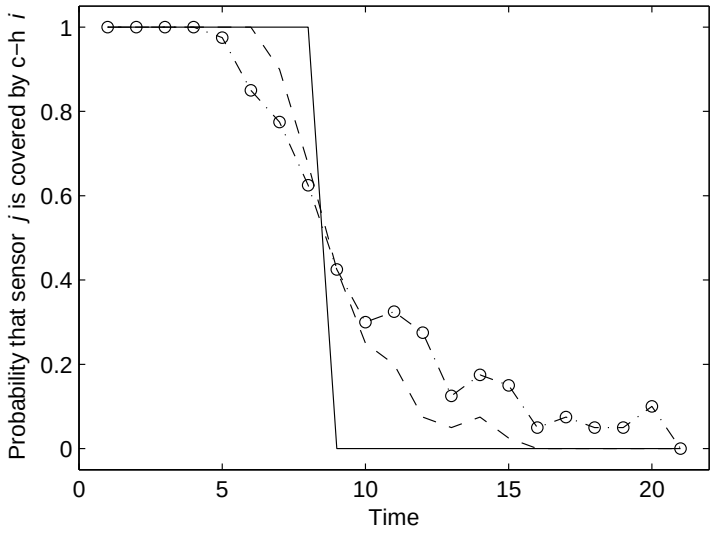
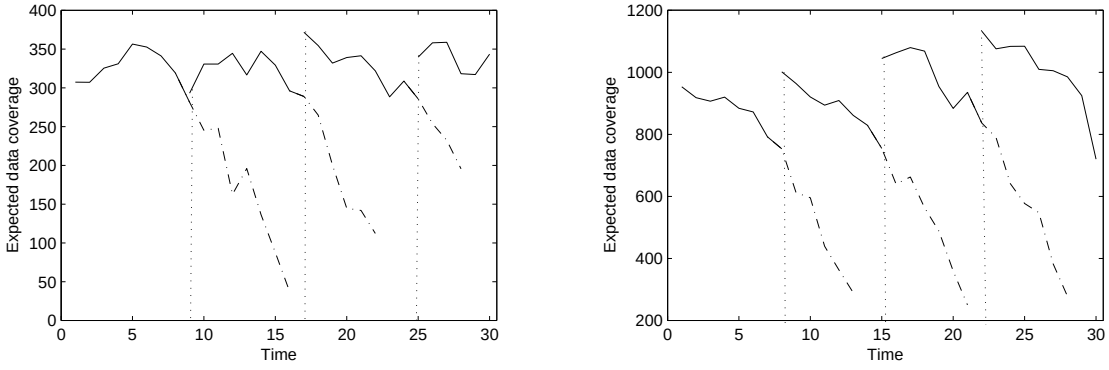


Figure 21: Coverage probabilities (small instance) ($p = 0.1$): (-) $\sigma = 0.0$; (- -) $\sigma = 3.0$; (-o) $\sigma = 6.0$.

Figure 21 reveals insights into the effect of the diffusion coefficient on the optimal decisions, which is enforced by constraint (4.9). Here, we consider cluster head i that is within the range of sensor j at time 0. When the diffusion coefficient is ignored (i.e. $\sigma = 0$), the sensors follow a deterministic path. Therefore, the probability that cluster head i remains within the range of sensor j is either 1 or 0. On the other hand, as σ increases, the probability that cluster head i and sensor j are connected decreases with time; therefore, expected data coverage decreases.



(a) Small problem instance.

(b) Large problem instance.

Figure 22: Data coverage with location updates ($\sigma = 6.0$, $p = 0.1$).

Figure 22 shows the effect of sensor location updates on expected data coverage. For example, at times 0, 9, 17, and 25 in the small instance, the exact sensor locations are known. After update times, the expected data coverage decreases due to the uncertainties in the sensor locations. The optimal cluster head assignments and update times are presented in Table 16. When the sensor locations are updated, more cluster heads are relocated because the availability of exact sensor locations leads to opportunities for improving the demand coverage by relocating cluster heads.

Table 16: Optimal cluster head assignments ($\sigma = 6.0$, $p = 0.1$).

Time	Cluster head assignments				
1	7	15	19	21	25
2	7	15	19	21	25
3	7	15	19	21	25
4	7	15	9	21	24
5	7	15	9	22	24
6	7	19	9	17	24
7	7	19	9	17	24
8	7	19	9	17	23
9	7	14	10	22	25
10	12	14	10	22	25
11	12	14	10	22	25
12	12	20	10	22	25
13	12	20	10	22	25
14	12	20	10	22	25
15	12	20	23	22	25
16	12	19	23	15	25
17	4	17	22	15	25
18	4	17	22	15	25
19	4	17	22	15	25
20	4	17	23	15	25
21	4	17	23	15	25
22	19	17	23	15	25
23	19	17	23	15	25
24	19	17	23	15	25
25	19	9	22	15	25
26	19	9	22	15	25
27	19	9	22	15	25
28	17	9	23	15	25
29	17	9	23	15	25
30	17	9	23	15	25

Figure 23 shows the dependence of sensor location update times on the cost of location updates and relocation costs for the large instance. With increasing relocation cost, we observe that the first update time decreases. This is intuitive since, as the relocation costs increase, the sensor location information becomes more significant. On the other hand, the first update time is expected to increase when the update cost increases, which can be observed when update cost is 3 in Figure 23. Obviously, the object function value is decreasing in both the relocation and updating costs.

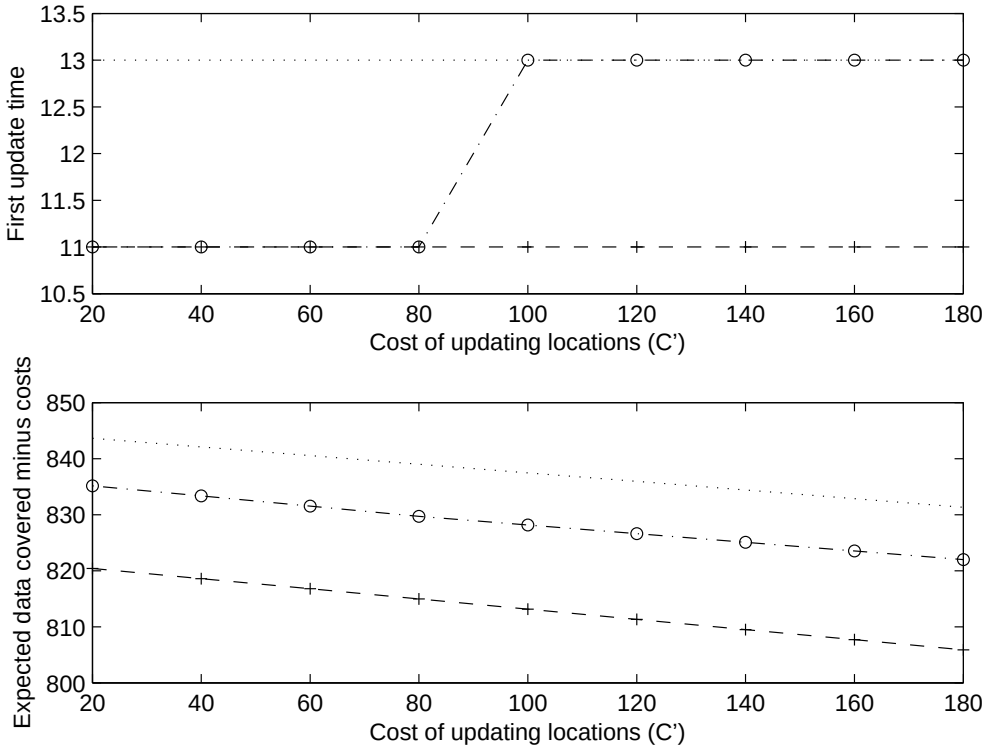


Figure 23: Impact of updating cost on update time and objective value (large problem instance) ($\sigma = 6.0$, $p = 0.1$): $(\cdot\,\cdot)$ $C = 1$; $(\text{-}\cdot\,\text{o})$ $C = 3$; $(\text{-}\,\text{o}\,\text{-})$ $C = 9$.

In this chapter, we developed a stochastic optimization model to maximize the demand coverage and minimize the costs of locating, and relocating, cluster heads in a WSN with mobile nodes and unreliable links. In the optimization model, we consider the stochastic evolution of sensor locations over time rather than assuming the positions are known at all times. The mathematical models are amenable to solution by a commercial solver. The numerical results demonstrated that the non-myopic cluster head relocation approach improves the data coverage when the sensors are mobile and their locations in the planning horizon is uncertain. Moreover, relocation and sensor location update time decisions are shown to be sensitive to the level of uncertainty of the sensor locations.

5.0 CONCLUSIONS AND FUTURE RESEARCH

Chapter 2 presented both single- and multi-hop models for evaluating the performance of large-scale WSNs with time-limited events and queries using a queueing-theoretic approach. With the former model, we approximate the steady state proportion of query failures as a measure of quality-of-service and the approximation is shown to be insensitive to the network size. The latter model leads to an approximation of the same measure by considering the realistic effects of a limited transmission range. We showed that this approximation converges to its infinite range (single-hop) counterpart as the sensor transmission range tends to infinity. Our approach is unique in that it considers time-limited event agents and queries and is not limited to memoryless (exponentially-distributed) lifetimes. The numerical results demonstrated that the approximations are remarkably accurate, even when distribution and network topology assumptions were violated. The approximations of Chapter 2 can be used for optimization of large-scale, query-based WSNs. The energy expended for transmissions serves as a proxy for the total energy expenditure at a node because event agent or query transmissions are the primary energy-consuming activity. Our models lead to an approximated proxy for energy expenditure in the form of traffic rates, which can be used to obtain optimal operating parameters so that a quality-of-service constraint is satisfied.

Although the models of Chapter 2 are mathematically valid, and the approximations are easy to compute, they currently lack the flexibility to account for some realistic features of WSNs. First, in the present framework, we assumed that all transmissions are perfect (i.e., there are no fading effects or packet collisions) so that retransmissions are not necessary. In future work, it may be possible to model each transmission queue as a single-server retrial queueing station to account for event agents and queries that require retransmission. Second, it was assumed that event agents and queries are transmitted in the order in which they are received. However, it is more realistic to incorporate the deadlines of packets in the transmission queue so as to prioritize transmissions (e.g., giving preference to those queries with the smallest remaining lifetime). One approach is to

consider real-time queueing network theory (see Lehoczky [44,45]). Third, and finally, event agents and queries were assumed to use a random-walk routing protocol that does not exploit additional state information that can be used to improve routing and potentially reduce the overall proportion of query failures. In the future, it will be instructive to develop similar approximations for WSNs that use other common routing protocols.

In Chapter 3, we addressed the problem of network lifetime maximization in query-based WSNs subject to connectivity and quality-of-service requirements. An approximate probability of network connectivity considering the impact of border effects of a square region was derived. Subsequently, we exploited the approximations of Chapter 2 to formulate nonlinear and linear mixed integer problems to maximize the network lifetime by choosing the optimal time-to-live counter, transmission range and active/sleep schedules for each time period of a finite planning horizon. Moreover, we presented a solution algorithm for solving a special case of the model in which all of the alive nodes are in active mode in the planning horizon. The numerical results indicated that the network lifetime can be extended by selecting optimal parameters. In addition, we showed that high variability in the energy expended by sensor nodes can lead to shorter network lifetimes on average.

Our research on maximizing WSN lifetime suggests a number of future directions. First, it remains for future to develop extensive simulation models to assess the quality of the optimal solutions of the mathematical models. A common WSN simulation environment such as OPNET does not account for node failures and does not allow parameter selection for each time period; therefore, one can seek to create simulation models that incorporate selecting time-to-live counter, transmission range and active/sleep schedules for each time period of a finite planning horizon. The impact of the model assumptions can be studied by comparing the model solutions with simulated values. Second, it will be imperative to develop efficient solution algorithms for the most general lifetime maximization problem. The proposed linear model has the form of a knapsack problem with additional constraints and is, therefore, amenable to solution by a commercial solver. However, obtaining the model parameters and solving the linear model requires significant computational time, so the existing model does not scale well for large-scale networks. While we developed an algorithm for the special case, developing an efficient algorithm to reduce the size of the linearized model and heuristic methods to solve the linearized model will be critical. Third, one can seek to develop stochastic models to maximize network lifetime considering the time sensitivity of information and

randomness of node deployment, event agents and query moving patterns. Stochastic optimization methods are relatively new in WSN research and therefore proposing algorithms with chance constraints would be appealing.

Chapter 4 explored the problem of cluster head location and relocation in mobile wireless sensor networks while focusing on relocation and information updating costs. We first developed a nonlinear stochastic optimization model to maximize the coverage of the network. Then, we described a way to transform the nonlinear stochastic model into a deterministic linear model, which can be solved using a commercial solver. Finally, the numerical results revealed the effect of relocation decisions on the coverage and energy expenditure related to the relocation and location updates. In addition, the effect of certain network parameters, including variability in the movement patterns of sensor nodes, and the costs of relocating cluster heads, was explored.

The proposed cluster head relocation optimization models can be solved using commercial solvers; however, heuristic algorithms are needed to solve these problems for large networks. The clustering models and algorithms in the data processing and wired sensor network literature are not applicable to the WSNs because of operational characteristics of these networks. Specifically, sensor nodes in mobile WSNs are typically unaware of their locations. Moreover, the movement of sensors without perfect location information increases the complexity of the problems. Therefore, future developments for these models should focus on sensor movement models, stochastic optimization models and solution techniques to handle general sensor movement patterns. Application-specific algorithms are also required to improve the lifetime and coverage in WSNs, where sensors move according to a specification of the network. Additionally, other than coverage and relocation costs, depending on the application, selection of cluster heads should take into account uniform distribution of energy consumption and quality of service. Finally, another area for future work involves locating and relocating cluster heads when the sinks are mobile and data aggregation is performed by a mobile agent.

BIBLIOGRAPHY

[1] A. A. Abbasi and M. Younis. A survey on clustering algorithms for wireless sensor networks. *Computer Communications*, 30:2826–2841, 2007.

[2] J. Ahn and B. Krishnamachari. Modeling search costs in wireless sensor networks. In *Proceedings of the 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 1–6, 2007.

[3] W. Aioffi, C. Valle, G. Mateus, and A. Cunha. Balancing message delivery latency and network lifetime through an integrated model for clustering and routing in wireless sensor networks. *Computer Networks*, 55:2803–2820, 2011.

[4] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Ad hoc Networks*, 3:325–349, 2005.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38:393–422, 2002.

[6] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11:6–28, 2004.

[7] S. L. Albin. On Poisson approximations for superposition arrival processes in queues. *Management Science*, 28:126–137, 1982.

[8] G. Anastasi, M. Conti, and M. D. Francesco. Extending the lifetime of wireless sensor networks through adaptive sleep. *IEEE Transactions on Industrial Informatics*, 5:351–365, 2009.

[9] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7:537–568, 2009.

[10] Y. P. Aneja, R. Chandrasekaran, X. Li, and K. P. K. Nair. A branch-and-cut algorithm for the strong minimum energy topology in wireless sensor networks. *European Journal of Operational Research*, 204(3):604–612, 2010.

[11] P. Antoniou, A. Pitsillides, A. Engelbrecht, T. Blackwell, and L. Michael. Applying swarm intelligence to a novel congestion control approach for wireless sensor networks. In *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, pages 1–7, 2011.

[12] B. Ata. Dynamic power control in a wireless static channel subject to a quality-of-service constraint. *Operations Research*, 53:842–851, 2005.

[13] T. Banka, G. Tandon, and A. P. Jayasumana. Zonal rumor routing for wireless sensor networks. In *Proceedings of the International Conference on Information Technology: Coding and Computing*, pages 562–567, 2005.

[14] A. Barroso, U. Roeding, and C. Sreenan. Maintenance efficient routing in wireless sensor networks. In *Proceedings of IEEE Workshop on Embedded Networked Sensors*, pages 97–106, 2005.

[15] P. Bellavista, A. Corradi, and E. Magisretti. Comparing and evaluating lightweight solutions for replica dissemination and retrieval in dense manets. In *Proceedings of the 10th IEEE Symposium on Computers and Communications*, pages 43–50, 2005.

[16] C. Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *Proceedings of MobiHoc '02: The 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 80–91, 2002.

[17] C. Bettstetter. On the connectivity of ad hoc networks. *The Computer Journal*, 47(4):432 –447, 2004.

[18] N. Bisnik and A. A. Abouzeid. Queuing network models for delay analysis of multihop wireless ad hoc networks. *Ad Hoc Networks*, 7:79–97, 2009.

[19] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 22–31, 2002.

[20] N. Bulusu and S. Jha. *Wireless Sensor Networks: A Systems Perspective*. Artech House, Norwood, MA, 2005.

[21] R. L. Burden and J. D. Faires. *Numerical Analysis*. PWS Publishing Company, Boston, MA, 1993.

[22] J. M. Caicedo, J. Marulanda, P. Thomson, and S. Dyke. Monitoring of bridges to detect changes in structural health. In *Proceedings of the 2001 American Control Conference*, pages 453–458, 2001.

[23] M. Cardei, W. Jie, L. Mingming, and M. O. Pervaiz. Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In *Proceedings of the IEEE International Conference Wireless and Mobile Computing, Networking and Communications*, pages 438–445, 2005.

[24] R. Cerulli, R. DeDonato, and A. Raiconi. Exact and heuristic methods to maximize network lifetime in wireless sensor networks with adjustable sensing ranges. *European Journal of Operational Research*, 220(1):58–66, 2012.

[25] I. Chen, A. P. Speer, and M. Eltoweissy. Adaptive fault-tolerant QoS control algorithms for maximizing system lifetime of query-based wireless sensor networks. *IEEE Transactions on Dependable and Secure Computing*, 8(2):161–176, 2011.

[26] P. Chen, B. O'Dea, and E. Callaway. Energy efficient system design with optimum transmission range for wireless ad hoc networks. In *Proceedings of the IEEE International Conference on Communications*, pages 945–952, 2002.

[27] C. F. Chiasserini, R. Gaeta, M. Garetto, M. Gribaudo, D. Manini, and M. Sereno. Fluid models for large scale wireless sensor networks. *Performance Evaluation*, 64:715–736, 2007.

[28] S. Ci, M. Guizani, and H. Sharif. Adaptive clustering in wireless sensor networks by mining sensor energy data. *Computer Communications*, 30:2968–2975, 2007.

[29] IBM ILOG: CPLEX, 2011. http://www.ilog.com/products/cplex.

[30] A. S. D. Krivitski and R. Wolff. A local facility location algorithm for sensor networks. In *Proceedings of the Distributed Computing in Sensor System: First IEEE International Conference*, pages 368–375, 2005.

[31] I. Daubechies and J. C. Lagarias. Sets of matrices all infinite products of which converge. *Linear Algebra and its Applications*, 161:227–263, 1992.

[32] G. Degirmenci, J. P. Kharoufeh, and R. O. Baldwin. On the performance evaluation of query-based wireless sensor networks. *Performance Evaluation*, 70:124–147, 2013.

[33] J. Deng, Y. S. Han, P. Chen, and P. K. Varshney. Optimum transmission range for wireless ad hoc networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, pages 1024 – 1029, 2004.

[34] I. Dietrich and F. Dressler. On the lifetime of wireless sensor networks. *ACM transactions on Sensor Networks*, 5:1–39, 2009.

[35] P. Diggle. *Statistical Analysis of Spatial Point Patterns.* Arnold, London, 2003.

[36] N. Dimokas, D. Katsaros, and Y. Manolopoulos. Energy-efficient distributed clustering in wireless sensor networks. *Journal of Parallel and Distributed Computing*, 70:371–383, 2010.

[37] P. Ding, J. Holliday, and A. Celik. Distributed energy-efficient hierarchical clustering for wireless sensor networks. *Distributed Computing in Sensor Systems*, 3560:322–339, 2005.

[38] O. Dousse, C. Tavoularis, and P. Thiran. Delay of intrusion detection in wireless sensor networks. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 155–165, 2006.

[39] P. Eftekhari, H. Shokrzadeh, and A. T. Haghighat. Cluster-base directional rumor routing protocol in wireless sensor network. *Communications in Computer and Information Science*, 101:394–399, 2010.

[40] F. Flammini, A. Gaglione, F. Ottello, A. P. C. Pragliola, and A. Tedesco. Towards wireless sensor networks for railway infrastructure monitoring. In *Proceedings of the Electrical Systems for Aircraft, Railway and Ship Propulsion (ESARS)*, pages 1–6, 2010.

[41] T. Furuta, M. Sasaki, F. Ishizaki, A. Suzuki, and H. Miyazawa. A new clustering model of wireless sensor networks using facility location theory. *Journal of the Operations Research Society of Japan*, 52:366–376, 209.

[42] Q. Gao, K. J. Blow, D. J. Holding, I. W. Marshall, and X. H. Peng. Radio range adjustment for energy efficient wireless sensor networks. *Ad Hoc Networks*, 4(1):75–82, 2006.

[43] D. Gross and C. Harris. *Fundamentals of Queueing Theory.* John Wiley & Sons, New York, NY, 1998.

[44] Y. Gu, Y. Ji, and B. Zhao. Maximize lifetime of heterogeneous wireless sensor networks with joint coverage and connectivity requirement. In *Proceedings of the Eighth IEEE International Conference on Embedded Computing*, pages 226–231, 2009.

[45] V. C. Gungor and G. P. Hancke. Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Transactions on Industrial Electronics*, 56(10):4258–4265, 2009.

[46] I. Gupta, D. Riordan, and S. Sampalli. Cluster-head election using fuzzy logic for wireless sensor networks. In *Proceedings of the 3rd Annual Communication Networks and Services Research Conference*, pages 255–260, 2005.

[47] R. W. Ha, P. H. Ho, X. S. Shen, and J. Zhang. Sleep scheduling for wireless sensor networks via network flow model. *Computer Communications*, 29(13-14):2469–2481, 2006.

[48] C. M. Harris and W. G. Marchal. Distribution estimation using Laplace transforms. *INFORMS Journal on Computing*, 10:448–458, 1998.

[49] S. Hedetniemi and A. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18:319–346, 1988.

[50] W. R. Heinzelman, A. P. Chandrakasan, and H. Blakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transaction on Wireless Communications*, 1:660–670, 2002.

[51] W. R. Heinzelman, A. P. Chandrakasan, and H. Blakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, pages 660–670, 2002.

[52] C. Herring and S. Kaplan. Component-based software systems for smart environments. *IEEE Personal Communications*, 7(5):60–61, 2000.

[53] D. S. Hochbaum. A nonlinear knapsack problem. *Operations Research Letters*, 17(3):103–110, 1995.

[54] L. Hu and D. Evans. Localization for mobile sensor networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pages 45–57, 2004.

[55] A. A. Islam, C. S. Hyder, H. Kabir, and M. Naznin. Stable sensor network(ssn): a dynamic clustering technique for maximizing stability in wireless sensor networks. *Wireless Sensor Network*, 2:538–554, 2010.

[56] F. Jiang, D. Huang, C. Yang, and F. Leu. Lifetime elongation for wireless sensor network using queue-based approaches. *The Journal of Supercomputing*, 59(3):1312–1335, 2012.

[57] P. N. K. Kalpakis, K. Dasgupta. Maximum lifetime data gathering and aggregation in wireless sensor networks. In *Proceedings of IEEE International Conference on Networking*, pages 685–696, 2002.

[58] R. D. Kane, D. C. Eden, S. Amidi, and D. Delve. Implementation of real-time corrosion monitoring with industrial process control and automation. In *Proceedings of Corrosion*, pages 1–16, 2007.

[59] A. R. Khan, S. A. Madani, K. Hayat, and S. U. Khan. Clustering-based power-controlled routing for mobile wireless sensor networks. *International Journal of Communication Systems*, 25:529–542, 2012.

[60] T. Kijewski-Correa, M. Haenggi, and P. Antsaklis. Wireless sensor networks for structural health monitoring: A multi-scale approach. In *Proceedings of the 2006 ASCE Structures Congress, 17th Analysis and Computation Specialty Conference*, pages 1–16, 2006.

[61] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health honitoring of civil infrastructures using wireless sensor networks. In *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks*, pages 254–263, 2007.

[62] L. Kleinrock. *Queuing Systems, Volume I: Theory.* A Wiley-Interscience Publication, New York, NY, 1975.

[63] A. Koucheryavy and A. Salim. Prediction-based clustering algorithm for mobile wireless sensor networks. In *Proceedings of the 12th International Conference on Advanced Communication Technology*, pages 1209–1215, 2010.

[64] B. Krishnamachari. *Networking Wireless Sensors.* Cambridge University Press, New York, NY, 2005.

[65] B. Krishnamachari and J. Ahn. Optimizing data replication for expanding ring-based queries in wireless sensor networks. In *Proceedings of the 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 1–10, 2006.

[66] V. G. Kulkarni. *Modeling and Analysis of Stochastic Systems.* Chapman and Hall, New York, NY, 1995.

[67] V. Kumar, S. Jain, and S. Tiwari. Energy efficient clustering algorithms in wireless sensor networks: A survey. *International Journal of Computer Science Issues*, 8:259–268, 2011.

[68] F. L. Lewis. *Wireless Sensor Networks.* John Wiley & Sons, 2005.

[69] J. Liu, X. Jiang, S. Horiguchi, and T. T. Lee. Analysis of random sleep scheme for wireless sensor networks. *International Journal of Sensor Networks*, 7(1):71–84, 2010.

[70] X. Liu and J. Shi. Clustering routing algorithms in wireless sensor networks: An overview. *KSII Transactions on Internet and Information Systems*, 6:1735–1755, 2012.

[71] K. S. Low, W. N. Win, and M. J. Er. Wireless sensor networks for industrial environments. In *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation*, pages 271 –276, 2005.

[72] J. P. Lynch and K. J. Loh. A summary review of wireless sensors and sensor networks for structural health monitoring. *The Shock and Vibration Digest*, 38(2):91–128, 2006.

[73] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pages 88–97, 2002.

[74] S. V. Manisekaran, R. Venkatesan, and G. Deivanai. Mobile adaptive distributed clustering algorithm for wireless sensor networks. *International Journal of Computer Applications*, 20:12–19, 2011.

[75] C. R. Mann, R. O. Baldwin, J. P. Kharoufeh, and B. E. Mullins. A trajectory-based selective broadcast query protocol for large-scale, high-density wireless sensor networks. *Telecommunication Systems*, 35:67–86, 2007.

[76] C. R. Mann, R. O. Baldwin, J. P. Kharoufeh, and B. E. Mullins. A queueing approach to optimal resource replication in wireless sensor networks. *Performance Evaluation*, 65:689–700, 2008.

[77] A. Milenkovic, C. Otto, and E. Jovanov. Wireless sensor networks for personal health monitoring: Issues and an implementation. *Computer Communications*, 29(13-14):2521–2533, 2006.

[78] H. Miranda, S. Leggio, L. Rodrigues, and K. Raatikainen. An algorithm for dissemination and retrieval of information in wireless ad hoc networks. In *Proceedings of the 13th International Euro-Par Conference (Euro-Par 2007)*, pages 891–900, 2007.

[79] A. Mishra, F. M. Gondal, A. A. Afrashteh, R. R. Wilson, R. D. Moffitt, R. K. Kapania, and S. Bland. Embedded wireless sensors for aircraft/automobile tire structural health monitoring. In *Proceedings of the 2nd IEEE Workshop on Wireless Mesh Networks*, pages 163–165, 2006.

[80] D. Niyato and E. Hossain. Sleep and wakeup strategies in solar-powered wireless sensor/mesh networks:performance analysis and optimization. *IEEE Transactions on Mobile Computing*, 6:221–236, 2007.

[81] C. Ok, S. Lee, P. Mitra, and S. Kumara. Distributed energy balanced routing for wireless sensor networks. *Computers and Industrial Engineering*, 57:125–135, 2009.

[82] P. Padhy, K. Martinez, A. Riddoch, J. K. Hart, and H. L. Ong. Glacial environment monitoring using sensor networks. In *Proceedings of the First Real-World Wireless Sensor Networks Workshop (REALWSN05)*, pages 10–14, 2005.

[83] K. Padmanabh, P. Gupta, and R. Roy. Transmission range management for lifetime maximization in wireless sensor network. In *Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, pages 138–142, 2008.

[84] D. J. Patel, R. Batta, and R. Nagi. Clustering sensors in wireless ad hoc networks operating in a threat environment. *Operations Research*, 53:432–442, 2005.

[85] C. Patra, P. Bhaumik, and D. Chakroborty. Modified rumor routing for wireless sensor networks. *International Journal of Computer Science Issues*, 7:31–34, 2010.

[86] A. Perrig, J. Stankovic, and D. Wagner. Security in wireless sensor networks. *Communications of the ACM*, 47(6):53 – 57, 2004.

[87] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, 12:63–78, 2006.

[88] L. Rodero-Merino, A. F. Anta, L. López, and V. Cholvi. Performance of random walks in one-hop replication networks. *Computer Networks*, 54:781–796, 2010.

[89] K. Romer and F. Mattern. The design space of wireless sensor networks. *IEEE Wireless Communications*, 2004.

[90] M. Sarkar and R. L. Cruz. Analysis of power management for energy and delay trade-off in a WLAN. In *Proceedings of the Conference on Information Sciences and Systems*, pages 1–6, 2004.

[91] C. Schurgers and M. B. Srivastava. Energy efficient routing in wireless sensor networks. In *Proceedings of the IEEE Military Communications Conference*, pages 357–361, 2001.

[92] C. Sevgi and A. Kocyigit. An optimal network dimensioning and initial energy assignment minimizing the monetary cost of a heterogeneous wsn. In *Proceedings of the 6th International Symposium on Wireless Communication Systems*, pages 517–521, 2009.

[93] J. Shanbehzadeh, S. Mehrjoo, and A. Sarrafzadeh. An intelligent energy efficient clustering in wireless sensor networks. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, pages 1–5, 2011.

[94] L. Shi, A. Capponi, K. H. Johansson, and R. M. Murray. Resource optimisation in a wireless sensor network with guaranteed estimator performance. *IET Control Theory and Applications*, 4(5):710–723, 2010.

[95] H. Shokrzadeh, A. T. Haghighat, and A. Nayebi. New routing framework base on rumor routing in wireless sensor networks. *Computer Communications*, 32:86–93, 2009.

[96] J. F. Shortle, P. H. Brill, M. J. Fischer, and D. Gross. An algorithm to compute the waiting time distribution for the M/G/1 queue. *INFORMS Journal on Computing*, 16:152–161, 2004.

[97] A. Sinha and A. P. Chandrakasan. Dynamic power management in wireless sensor networks. *IEEE Design and Test of Computers Magazine*, 18(2):62–74, 2001.

[98] I. Slama, M. C. Ghedira, B. Jouaber, and H. Afifi. Cluster based wireless sensor networks' optimization under energy constraints. In *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, pages 745–750, 2007.

[99] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications*, pages 472–476, 2001.

[100] J. A. Stankovic. Research challenges for wireless sensor networks. *ACM SIGBED Review*, 1(2):9–12, 2004.

[101] J. A. Stankovic, Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, S. Lin, S. Son, R. Stoleru, and A. Wood. Wireless sensor networks for in-home healthcare: Potential and challenges. In *Proceedings of High Confidence Medical Device Software and Systems Workshop (HCMDSS)*, pages 1–4, 2005.

[102] Y. B. Turkogullari, N. Aras, I. K. Altinel, and C. Ersoy. Optimal placement, scheduling, and routing to maximize lifetime in sensor networks. *Journal of Operational Research Society*, 61(6):1000–1012, 2010.

[103] J. H. W. Ye and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3):493–506, 2004.

[104] G. Wang, T. Wang, W. Jia, M. Guo, H. Chen, and M. Guizani. Local update-based routing protocol in wireless sensor networks with mobile sinks. In *Proceedings of the IEEE International Conference on Communications*, pages 3094 –3099, 2007.

[105] M. Welsh. Harvard Sensor Networks Lab: Volcano monitoring, 2004. http://fiji.eecs.harvard.edu/Volcano.

[106] D. Xia and N. Vlajic. Near-optimal node clustering in wireless sensor networks for environment monitoring. In *Proceedings of the 21st International Conference on Advanced Networking and Applications*, pages 43–50, 2007.

[107] X. Xing, G. Wang, J. Wu, and J. Li. Square region-based coverage and connectivity probability model in wireless sensor networks. In *Proceedings of the 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 1–8, 2009.

[108] C. Yan-rong, C. Jia-heng, H. Ning, and Z. Fan. Rumor routing based on ant colony optimization for wireless sensor networks. *Application Research of Computers*, 3:1033–1035, 2009.

[109] G. Yang. *Body Sensor Networks*. Springer-Verlag, London, 2006.

[110] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52:2292–2330, 2008.

[111] O. Younis and S. Fahmy. Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3:366–379, 2004.

[112] O. Younis, M. Krunz, and S. Ramasubramanian. Node clustering in wireless sensor networks: Recent developments and deployment challenges. *IEEE Network*, 20:20–25, 2006.

[113] M. Youssef, A. Youssef, and M. Younis. Overlapping multihop clustering for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 20:1844–1856, 2009.

[114] B. Yu and B. Xiao. Detecting selective forwarding attacks in wireless sensor networks. In *Proceedings of the 20th International Parallel and Distributed Processing Symposium*, pages 8–16, 2006.

[115] M. Yu, K. Leung, and A. Malvankar. A dynamic clustering and energy efficient routing technique for sensor networks. *IEEE Transactions on Wireless Communications*, pages 3069–3079, 2005.

[116] Z. Zhang, G. Mao, and B. D. Anderson. On the effective energy consumption in wireless sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2010.

[117] T. Zhong, S. Wang, S. Xu, H. Yu, and D. Xu. Time delay based clustering in wireless sensor networks. In *Proceedings of the 2007 IEEE Wireless Communication and Networking Conference*, pages 3959–3963, 2007.

[118] Y. Zhu, W. Wu, J. Pan, and Y. Tang. An energy-efficient data gathering algorithm to prolong lifetime of wireless sensor networks. *Computer Communications*, 33:639–647, 2010.