9-2007

Energy-Efficient Querying of Wireless Sensor Networks

Christopher R. Mann

ENERGY-EFFICIENT QUERYING OF

WIRELESS SENSOR NETWORKS

DISSERTATION

Christopher R. Mann, Lieutenant Colonel, USAF

AFIT/DS/ENG/07-19

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/DS/ENG/07-19

ENERGY-EFFICIENT QUERYING OF WIRELESS SENSOR NETWORKS

DISSERTATION

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Christopher R. Mann, BSEE, MSEE

Lieutenant Colonel, USAF

September 2007

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/DS/ENG/07-19

ENERGY-EFFICIENT QUERYING OF
WIRELESS SENSOR NETWORKS

Christopher R. Mann, BSEE, MSEE

Lieutenant Colonel, USAF

Approved:

	Date
Dr. Rusty O. Baldwin (Chairman)	23 Aug 07
Dr. Barry E. Mullins (Member)	21 Aug 07
Dr. Jeffrey P. Kharoufeh (Member)	21 Aug 07
Dr. Michael A. Marciniak (Dean's Representative)	21 Aug 07

Accepted:

M. U. Thomas Dean, Graduate School of Engineering and Management	31 Aug 07 Date

AFIT/DS/ENG/07-19

Abstract

Due to the distributed nature of information collection in wireless sensor networks and the inherent limitations of the component devices, the ability to store, locate, and retrieve data and services with minimum energy expenditure is a critical network function. Additionally, effective search protocols must scale efficiently and consume a minimum of network energy and memory reserves.

A novel search protocol, the Trajectory-based Selective Broadcast Query protocol, is proposed. An analytical model of the protocol is derived, and an optimization model is formulated. Based on the results of analysis and simulation, the protocol is shown to reduce the expected total network energy expenditure by 45.5 percent to 75 percent compared to current methods.

This research also derives an enhanced analytical node model of random walk search protocols for networks with limited-lifetime resources and time-constrained queries. An optimization program is developed to minimize the expected total energy expenditure while simultaneously ensuring the proportion of failed queries does not exceed a specified threshold.

Finally, the ability of the analytical node model to predict the performance of random walk search protocols in large-population networks is established through extensive simulation experiments. It is shown that the model provides a reliable estimate of optimum search algorithm parameters.

To Isabel, Natalie, and Alexander.

Acknowledgements

Although the pursuit of doctoral research is frequently a lonely process, it is hardly a solitary undertaking. Were it not for the understanding and sacrifices of many people, to whom I am greatly indebted, this research would not have been possible. In particular, I owe a great deal of thanks to my advisor, Dr. Rusty Baldwin, for his seemingly endless enthusiasm and patience. Dr. Baldwin's mentorship and guidance have been invaluable—a steadfast compass helping me to navigate the oftentimes unfamiliar oceans of academia and research for the past three years. I am also sincerely grateful to my committee members, Dr. Jeffrey Kharoufeh and Dr. Barry Mullins, for the selfless contribution of their valuable time, effort, and feedback to improve the quality of this research. Thank you!

Christopher R. Mann
August 2007

Table of Contents

Page

Page

Page

List of Figures

Page

Page

Page

List of Tables

Page

List of Symbols and Notation

α	Proportion of network nodes informed by an agent
α_i	Proportion of nodes informed by a type-*i* agent
$\alpha_{i,\max}$	Largest value of α_i that can be supported by the network
α^*	The proportion of informed nodes that results in least total energy expenditure
α_i^*	The optimum energy-centric replication level for a type-*i* agent
$\alpha_{\kappa_i}^*$	Minimum resource replication level capable of meeting the network's highest tolerable bound for the proportion of query failures while simultaneously minimizing the rate of received transmissions
β	The number of hops a query completes before encountering a network boundary
β_i	Lead time of an arriving type-*i* query
γ_i	Rate at which individual nodes generate type-*i* queries
δ	The cardinality of a node's one-hop neighbor set
δ_i	Rate of expiration of type-*i* agents
δ'	The number of receivers designated to receive a query transmission
δ'^*	The optimum number of receivers designated to receive a query transmission
ε_i	Total proportion of type-*i* query failures
ζ	Term used to account for the probability of a node being polled more than once by a reissued query

θ	Rate of arrival of packets other than the agents or queries of interest to a node's transmission queue
κ_i	Maximum allowable proportion of type-*i* query failures
λ_i	Rate of generation of type-*i* agents at a node
μ	Rate at which a node can successfully transmit packets contained in its transmission queue
$\pi_{0,i}$	Proportion of time that a node is *i*-uninformed, i.e., has no type-*i* events in its event table
ρ	The probability a node located along the response path adds the information contained in the response to its event table
τ_i	Arrival rate of externally-generated type-*i* queries to a node
$f(\alpha,\delta')$	The expected total energy expended by the network to inform the network, locate the resource, and return the response to the OQN for a specific (α,δ') pair
(α^*,δ'^*)	The (α,δ') pair that minimizes the total expected energy expenditure of the TSBQ search algorithm
A	The total energy expended by the network to advertise a resource to a subset of the network
A_i	Rate of type-*i* agent arrivals to a node
A_i^{xmt}	Total arrival rate of agents to a node's transmission queue
c_1	Average proportion of one-hop neighbors shared by each QN-PQN pair
D	Maximum effective node transmission range
D'	Mean distance between transmitter-receiver pairs
d	Straight-line distance between a randomly-chosen node and a random point on the network boundary
E	Maximum number of unique agent/query types in the network

E_{xmt}	Mean energy expended by a node to successfully transmit a single packet to the intended receiver(s)
E_{rcv}	Mean energy expended by a node to receive a single packet
$E[X]$	Expected value of the random variable X
k	Maximum number of query transmissions that can be made to unique neighboring nodes before locating at least one informed node
l	Number of agents in the node's event table
m	Number of agents awaiting transmission in the node's transmission queue
N	Total number of nodes in the network
n	Total number of unique queries generated by n OQNs to locate a particular agent
n'	Expected number of query attempts required to locate an informed node
OQN	*Origin Query Node*. The node that initiates the search for a specific network resource
$p_{l,m,q}$	Steady-state proportion of time the node spends in state (l, m, q)
PQN	*Potential Query Node*. The node that will be next to transmit a query if the current QN fails to locate an informed node
Q	The energy expended by the network to locate an informed node
q	Number of queries awaiting transmission in the node's transmission queue
QN	*Query Node*. A node that transmits a query on behalf of the OQN
R	The energy used by the network to return the response to the OQN
s_i	Number of type-i agents present in a node's event table
T	Sum of A, Q, and R—the total energy consumed to advertise a resource, locate the resource, and return the response to the OQN
TTL	*Time-to-Live*. The maximum number of times a packet may be transmitted. In general, $\text{TTL} = \alpha N - 1$

$Var[X]$	The variance of the random variable X
$X_{\alpha,\delta'}$	The random number of transmissions required to find an informed node for fixed values of α and δ'
$x_{\alpha,\delta'}$	The total number of query transmissions required to locate a specific resource using fixed values of α and δ'
$X^{\beta}_{\alpha,\delta'}$	Random number of hops required to locate an informed node when network boundaries limit the maximum distance each query may traverse
Y	The number of informed nodes within one-hop distance of the QN
Z_j	Bernoulli random variable denoting success or failure of the jth query hop

ENERGY-EFFICIENT QUERYING OF
WIRELESS SENSOR NETWORKS

1. Introduction

1.1 Introduction to Wireless Sensor Networks

From the beginning of the Information Age, the push in technology has been toward smaller, faster devices that are cheaper to produce than their predecessors. Additionally, the growth of the Internet and the success of wireless technologies in the last decade finally permit access to real-time information from nearly any location in the world. Accessibility to timely information creates a competitive advantage and, as a result, the demand to be constantly and instantly "connected" continues to increase the need for real-time data. The manpower and cost required to maintain real-time data is expensive, so automated sensing devices have been adapted to collect data autonomously. A natural evolution of this approach is toward smaller devices capable of collecting more information in less time and, thus, small sensing devices found their niche. As the number and scope of applications for these sensing devices increases, the number of devices needed to perform a particular task grows, leading to the development of sensor networks. Today, the scope of wireless sensor networks (WSN) is vast and increasing. Among their many uses, today's WSNs check the structural integrity of buildings, keep track of warehouse inventory, perform reconnaissance and surveillance of enemy territory, and monitor vital signs of hospital patients [ASC02].

The design of WSNs is driven by the unique characteristics of the sensor nodes (Figure 1). In their most basic form, sensor nodes consist of one or more sensors configured to collect data of interest, a processor, a limited amount of memory, a receiver/transmitter, and a power source. Deployed sensor nodes, in many ways, are not unlike several laptop computers connected to an IEEE 802.11 (WiFi) wireless network. Both node and computer collect/process data and communicate over a wireless medium, and both may change location. However, sensor nodes, even in relatively sparsely populated sensor networks, typically have many more "neighbors" than their 802.11 counterparts. While computers in an 802.11 network can communicate with each other through access points if necessary, sensor nodes cannot rely on being within range of such a device. Instead, every device has routing capabilities, and nodes cooperatively relay information to nearby nodes until it reaches its final destination. Finally, in addition to being power-limited due to their small size, nodes are often deployed to locations

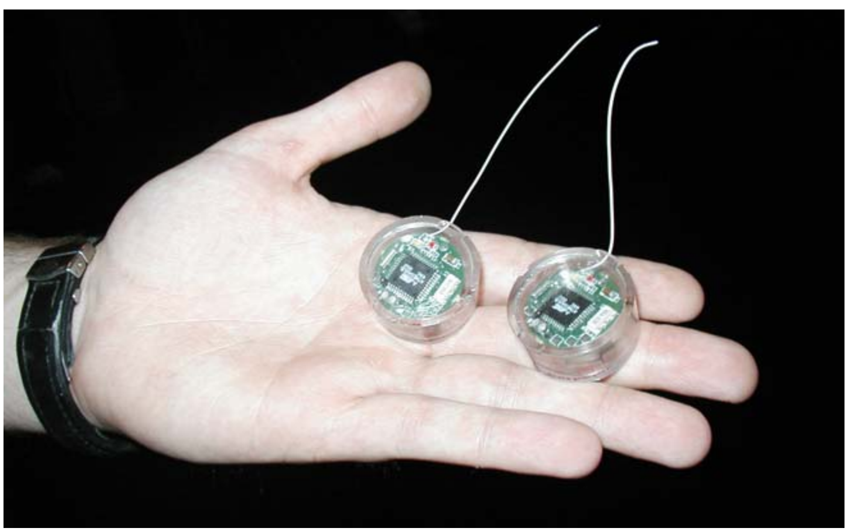


Figure 1: Typical Example of Wireless Sensor Nodes [UCB06].

where replenishing their energy supplies is extremely difficult or impossible. Consequently, power consumption becomes an important, if not the most important, issue driving WSN design and research [ASC02].

Three activities consume the majority of available power in a WSN: transmitting, receiving, and computing. Transmitting and receiving require the greatest expenditure of energy, with transmission being almost twice as costly as receiving in present-day devices [ROG06]. Computation is relatively cheap by comparison: 3,000 instructions can be performed for the same energy cost as transmitting a single bit a distance of 100 meters [TAH02].

In an ideal WSN, nodes consume power for transmitting, receiving, or computing only when necessary to accomplish network functions. If not otherwise required to perform a network function, nodes enter a low-power state, or *sleep mode*, to conserve energy. Because computing consumes the least energy of all node tasks, computation at the individual node level should be used whenever possible, especially if such computation can prevent the expenditure of the network's energy resources on more costly activities. Regardless, it must always be remembered that a wireless sensor network is useless unless it has the capability to gather the data of interest and communicate this information to the end-user (i.e., the entity that consumes the information gathered by the network). To this end, reliable communication between the data collector(s) and the data-consumer(s) is a critical function of every wireless sensor network.

1.2 Problem Statement

As the size and scale of wireless sensor networks continue to grow, two characteristics will be critical to maintaining their viability. First, high node densities (i.e., each node has a large number of one-hop neighbors) will be necessary to meet an increasing demand for high-precision sensor data while simultaneously providing redundant communication paths throughout the network. High node density also results in increased average lifetime per unit density of the network, a favorable property in networks composed of large numbers of low-cost, unreliable nodes [ZH04].

Second, small-footprint, scalable, energy-efficient applications will remain a critical enabling technology. Due to the distributed nature of data collection and storage in WSNs, no single node is likely to have all the information necessary to complete a particular task. Therefore, key among these critical applications is the ability of individual nodes to locate data and services within the network when on-board resources are insufficient. However, locating information requires nodes to expend precious energy reserves thereby reducing both node and network lifetime. Unfortunately, although several search algorithms are proposed in the open literature, much of the analysis of these algorithms is limited to the results obtained from simulation; few have been studied using analytical methods and even fewer from measuring the performance of an actual WSN. Additionally, there are currently no analytical models to examine the effects of limited resource lifetimes on optimal resource replication levels, aggregate network storage requirements, and energy efficiency. Furthermore, there is no literature on resource requests with deadlines nor are there any analytical models that predict the proportion of resource requests that will fail to locate the desired resource within an allotted timeframe.

1.3 Research Goals

The focus of this research is to overcome the deficiencies noted above. The research goals of this dissertation are summarized as follows:

1. Develop, model, analyze, and optimize an energy-efficient, scalable, small-footprint search protocol suitable for use in wireless sensor networks.
2. Develop an analytical node model for determining energy-efficient resource replication levels when (1) network resources have limited lifetimes, (2) deadlines are associated with resource requests, and (3) the proportion of failed requests may not exceed a specified level.
3. Evaluate the efficacy of the analytical node model to predict the performance of a search algorithm in large-population wireless sensor networks.

1.4 Dissertation Overview

This chapter provided an introduction to wireless sensor networks, their unique limitations, and the challenges they present for efficient design. The necessity of energy-efficient search algorithms in large-scale, high-density networks was discussed, and a short summary of the research goals of this dissertation was provided. Chapter 2 presents a survey of the relevant literature. Chapter 3 describes the specific goals of this research, characterizes the system under test, defines and analyzes key performance parameters, and discusses specific performance metrics. Chapter 4 details the development and analysis of a new search algorithm, the Trajectory-based Selective Broadcast Query (TSBQ) protocol. A mathematical model of TSBQ is developed, analyzed, and optimized for energy-efficient performance, and the performance of the protocol is evaluated via simulation experiments. In Chapter 5, a node model based on queueing theory is developed for analyzing search algorithm performance in networks with lifetime-limited resources and time-constrained queries. This node model is used to ascertain the resource replication levels required to minimize total expected network energy expenditure while simultaneously ensuring a specified maximum proportion of query failures is not exceeded. In Chapter 6, the utility of the node model developed in the previous chapter is examined in networks with large node populations. Chapter 7 provides a summary of the major results and contributions of this research.

2. Background

The field of wireless sensor networks is relatively new, and the study of search algorithms for these networks is newer still. However, there is no scarcity of available literature on this topic. In general, the body of search algorithm literature can be categorized into one or more classes based on the manner in which information is stored within the network and the means by which information is extracted from the network. Section 2.1 provides an overview of the general classes of WSN search algorithms and a detailed discussion of specific algorithms relevant to this research.

Mathematical modeling, analysis, and optimization of WSN search algorithms are key parts of this research. Section 2.2 describes the most common approaches for analyzing and optimizing the performance of WSN search algorithms.

Finally, no discussion of WSN search algorithms would be complete without an understanding of the necessary supporting services: localization algorithms, medium access control protocols, and routing algorithms. A broad survey of each of these areas is provided in Section 2.3.

2.1 Search Algorithms in Wireless Sensor Networks

When discussing the exchange of information between data collectors/providers and data consumers within a wireless sensor network, there are two distinctly orthogonal means to facilitate communication. These methods are referred to as *push* and *pull*. Classification of a network into a specific category is dependent on the mechanism which triggers a node to transmit its data. The majority of existing networks use search algorithms that fall somewhere in the middle of the spectrum between pure push and pull. These hybrid push-pull protocols are of particular interest to this research because their parameters can often be readily adjusted based on the requirements and characteristics of the network.

In the remainder of this document, the naming conventions of graph theory will be used to simplify the discussion. Nodes that provide resources (i.e., data and/or services) to the network are called *source* nodes, and nodes that require/request access to resources are *sink* nodes. Intermediate nodes that pass information and/or requests on behalf of the sink and source nodes are called the *transmitting* node or the *receiving* node, depending on the communication mode being used.

2.1.1 "Push" Networks

A push network assumes source nodes are aware of the presence and location of the sink node(s) and are also capable of making independent judgments regarding the sink's utility of collected data. However, if the source node cannot make these types of judgments (e.g., because the sink's data requirements frequently vary), then the only prudent alternative for the push-based network is for each source node to transmit all of its data to the sink. Push-based networks are preferred when the end-user's information requirements and the designation of sink nodes are relatively static, and the end-user is concerned with minimizing the amount of elapsed time between the moment the data is gathered by the source and its arrival at the sink. However, the transmitted information may or may not be useful to the sink. If much of the information transmitted by each source node has little utility to the sink, then the network is wasting its limited energy reserves. An alternative is for each source node to hold its information locally until it receives a specific data request from the sink. Networks that operate in this manner are called *pull* or *query-based* networks.

2.1.2 "Pull" Networks

When a node observes an event in a typical wireless sensor network employment scenario, the node determines locally whether the information will be transmitted through the network to the end-user(s). This decision, however, should not be made lightly since transmitting data is the most energy-expensive operation a node undertakes [ASC02]. When a node transmits information an end-user cannot use, energy is expended not only by the node that originally transmitted the data, but also by every node that forwarded the data. Thus, the total energy cost for poor transmission decisions is significant and decreases the useful lifetime of the network.

If the end-user's information requirements are well-defined or change infrequently, a local decision to transmit is appropriate. The decision can also be further simplified by limiting the type of data collected and the frequency of observations. In other applications, however, nodes may be required to observe a diverse or dynamic set of phenomena on a frequent basis. Unless latency is a concern, it is not feasible nor is it appropriate from an energy-efficiency perspective for nodes to transmit their data through the network. Rather, nodes should be notified by an end-user when and what type of data to transmit. This type of network is called *pull* or *query-based* because nodes transmit data only in direct response to an end-user's request.

The challenge with this approach is the end-user's query must be routed to the node that has the desired information; however, the end-user will likely not know which node(s) hold data of interest. Furthermore, the information requested by the end-user may not be in the network at all (i.e., no node has observed an event related to the end-user's request). Unfortunately, it is difficult for the query node to determine the specific failure mode of a query. It is unlikely that the query node will be capable of distinguishing between queries that fail due to non-existent information, routing failure within the network, or inability to find an informed node.

Given that the desired information exists in the network, the goal of query-based routing is to minimize the probability of a query failure. Therefore, if a query is answered with a negative reply, the end-user has a high degree of confidence the information does not exist in the network and another query need not be sent. Additionally, the number of transmissions required to locate the node(s) that possess the data of interest should be minimized to reduce the energy expended by the network.

The dual goals of reducing network energy expenditure while simultaneously maximizing the probability of query success are often at odds. The end-user prefers to search every node in the network for the desired data, but this is clearly not in the best interest of the energy-constrained nodes. To save energy, nodes should not transmit unless specifically requested; however, this hampers the ability to discover nodes with the desired data, especially in sensor networks with hundreds or thousands of nodes. A compromise is for each node that has information (i.e., a *witness node*) to share its data, or the fact that it possesses certain types of data, with a specific node or subset of nodes in the network. Thus, a query has only to locate one of these informed nodes to determine the data is available and where it can be found. A network of this type is referred to as a *hybrid push-pull* network because nodes send their information to a subset of the network's nodes without a specific request (i.e., push), but this information is not forwarded outside this subset of nodes unless a request is received (i.e., pull).

A straightforward, although somewhat naïve, approach to locating informed nodes is to flood the network with the query. In this manner, the querier can be assured every node in the network is examined for information related to the query; if the information exists, it will be found. However, flooding requires $O(N)$ node transmissions (where N is the number of network nodes) [BE02]. Alternatives to flooding seek to maximize the probability of finding information within the network (assuming the information exists) yet minimize the total amount of energy expended by the network for transmissions. One of the most successful hybrid push-pull query strategies, called *rumor routing*, was proposed in [BE02] and is discussed in detail in Section 2.1.3.1.

2.1.3 Hybrid "Push-Pull" Networks

Depending on the physical characteristics and data requirements of the network, information collected by nodes in hybrid push-pull networks is forwarded to a subset of the network's nodes based on either the network topology or the characteristics of the data itself; these approaches are categorized as *geo-centric* and *data-centric*, respectively. The remainder of this section discusses the rumor routing search algorithm, as well as several rumor routing variants. The section concludes by presenting a survey of several geo-centric and data-centric search protocols and discussing of the advantages and disadvantages of each approach.

2.1.3.1 Rumor Routing

The majority of routing algorithms use the physical locations of the nodes to determine a suitable route from the sender to the destination. This approach to routing strategy is logical when a node is designed to detect specific phenomena and then send a report of the event to a central location for further analysis. However, in contrast to this type of *event-based* approach, future applications of WSNs may be more likely to be *query-based* due to the distributed nature of information within the network. If nodes are unable to determine the utility of the data they gather in advance, using energy to transmit every event across the network is inefficient. Thus, the job of the query is to search the network for information it can use to answer a specific question.

The problem in a query-based routing approach is determining the best route from the requestor to the event. Rumor routing is designed to solve the query-routing problem by having witness nodes (i.e., nodes which observe an event of possible interest) inform a portion of the network about an observed event and the availability of data regarding that event [BE02]. As queries are subsequently propagated through the network, they are likely to encounter nodes aware of specific events. These nodes then direct the query toward the location of the event of interest. This scheme creates a hybrid push-pull network in which information concerning witnessed events is pushed to a subset of the network, and queries pull this information from the informed nodes.

Rumor routing is fundamentally based on the probability of random lines intersecting within a bounded rectangular region [BE02]. According to simulation experiments in [BE02], the probability of two random lines crossing in a rectangular plane is 69%. If five random lines are drawn in the same space, the probability of another line crossing at least one of them increases to 99.7%. Correspondingly, if there are five paths to a known event within a network, it can be inferred there is a high probability of a query encountering at least one of the known paths to that event.

To create paths to an event, witness nodes must keep the network informed. This information could be spread through broadcast or flooding techniques, but these have already been shown to be inefficient for most applications. Additionally, the example of intersecting lines demonstrates that only a small percentage of the network needs to be informed of an event for a query to locate it. For this reason, rumor routing proposes that witness nodes create *agents*, i.e., packets created for the purpose of "wandering" the network to keep distant nodes informed about local events. Agents travel from node to node by choosing a random receiving node at each hop. Upon arrival at a node, an agent synchronizes its information with the node's on-board *event table*. The event table stores information related to particular events and may include specific data and/or a path back to the witness node. If a node subsequently receives a query and it has a corresponding entry in its event table, the node will send the query on a path to the witness node to collect the information or will answer the query with the desired information if available. If a node has no information related to a received query, it forwards the query to a randomly-chosen neighboring node. This process continues until the query either finds a path to the event or expires.

Simulations of rumor routing indicate 98.1% of queries find the desired event path and are delivered successfully to the corresponding witness node [BE02]. Although average hop count per query and setup transmission costs are somewhat high (an average of 92 hops per query and 31,031 transmissions for setup were reported by [BE02]), overall energy costs are still only a fraction of the cost of flooding.

The distributed nature of data within a WSN makes it impractical for individual nodes to report every event across the network. As an alternative, rumor routing requires the query to find a path to the data of interest. Although rumor routing may not be the best choice for applications where low latencies are important or reportable events are well-defined in advance, it shows promise for networks where the number of queries related to an event is fairly low or the costs of creating a geographic routing system are high.

2.1.3.2 Rumor Routing Variants

The primary criticism of rumor routing is its reliance on the random walk used by both the agent to inform the network and the query to locate the information of interest. Although inadvertent backtracking by an agent or query can be eliminated by including a table of visited nodes in the agent/query packet, the size of this table grows at each hop, forcing nodes to expend more energy for transmission and jeopardizing the scalability of the protocol. Additionally, this strategy cannot eliminate the possibility of the agent/query visiting nodes in a spiral path [CSC05]. Spiral paths, when traveled, result in little spatial diversity; thus, agents may not travel very far from the witness node, and queries may never reach distant informed nodes. In addition to the difficulties imposed by the random walk routing method, rumor routing is also susceptible to *query slipping*, a phenomenon that results when a query fails to locate an informed node despite intersection of the agent and query trajectories [PTL+05].

To combat these problems, several variants of rumor routing have been proposed. Some of these variants are *geo-centric* [BTJ05, CSC05, SKH03] while others are *data-centric* [IGE00, RKY+02, RKS+03]. Also, the related field of unstructured peer-to-peer file sharing networks provides useful insight into the challenges posed by the search problem in WSNs.

2.1.4 Geo-centric Search Algorithms

Geo-centric variants of rumor routing frequently attempt to eliminate the problems associated with the random walk by imposing order or direction to the path traveled by the agent and query. For example, rumor routing's dual problems of spiraling agent/query routes and ever-increasing packet size (due to the need to record previously-visited nodes to prevent backward paths) can be solved by forwarding agents and queries using straight-line routing (SLR) [CSC05]. Routing agents and queries along curves was proposed in [IB05]. REDMAN [BCM05] is similar to SLR in that agents and queries are forwarded along straight-line trajectories. However, resource replicas are stored only at every *k*th node along the agent's path; the remaining intermediate nodes store a pointer to the nearest available replica. Zonal Rumor Routing [BTJ05] is an extension of rumor routing that partitions the network into artificial zones for the purpose of choosing intermediate nodes for agent/query routing. Neighboring nodes assigned to unvisited zones are favored when choosing an agent or query's next hop, thus improving the probability of a successful query.

The advantage of the geo-centric approach is that these rumor routing networks achieve a relatively high degree of data redundancy by using agents to propagate data. In the event the witness node and/or one or more informed nodes fails, the data collected by the witness node has a high probability of being preserved within the network. To obtain this level of redundancy, the network pays an energy cost of $O(\sqrt{N})$ point-to-point message transmissions [SRK+03]. The primary disadvantage of the geo-centric approach is the query must locate the desired data within the network; this search for data typically results in greater latency.

In a manner similar to rumor routing, *quorum-based* search protocols [LHJ06, MKB05, Sto99] facilitate intersection between queries and their corresponding agent trajectories by forwarding along straight-line paths in each of the four cardinal directions. For example, GCLP [TV04] propagates agents (called *content advertisements*) and queries along straight-line trajectories in the north-south and east-west directions, respectively. This method guarantees intersection of a query with at least one Content Location Server (i.e., a node aware of the location of a specific resource). Quorum-based schemes can also achieve a measure of energy efficiency by aggregating advertisements at each node prior to transmission. However, most quorum-based schemes require nodes to maintain sizeable stores of information regarding the location of distant nodes; in mobile networks, this information must be frequently updated or the node risks returning stale information in response to a query. Also, to ensure agent-query intersection, quorum-based search protocols must treat all resources with equivalent importance. Both popular and unpopular items consume the same amount of network storage capacity, and the mean energy and latency required to locate both popular and unpopular items are the same. As will be shown in Chapter 4, this paradigm forces over-representation of unpopular items within the network's aggregate storage capacity and increases the total energy expended for popular item queries.

2.1.5 Data-centric Search Algorithms

Rumor routing, its variants, and quorum-based approaches can be described as geo-centric because the dispersal of resource advertisements and/or replicates is based on network topology or direction. Such approaches differ from data-centric search algorithms in that the requesting node has no knowledge of the location of the desired resource when it issues the query. As an alternative, resources in data-centric networks are self-organized to facilitate answering queries. For example, all nodes sensing temperature readings between 55 and 60 degrees might forward their observations to a specific node or group of nodes. Therefore, the location of data can be determined based solely on the information required by the query, thus obviating a search of the entire network.

The Geographic Hash Table (GHT) is one such data-centric storage protocol that assigns each event to a particular geographic location within the network [RKY+02, RKS+03, SRK+03]. As nodes gather data related to specific events, they determine which node the data should be sent to by hashing the event key using a hash table. Thus, similar events will be forwarded to the same location. The query node also has access to the hash table, so it can independently determine the location of the desired data. Queries are forwarded directly to the location that holds the desired information, thereby decreasing latency as well as energy expenditure due to transmissions.

The data-centric approach is not without its own unique set of challenges and limitations. First, the hash space of the hash table includes the entire deployment region of the network, but it is unlikely that a node is located in the exact position specified by the hash function. In this case, the information is stored in the node closest to the hashed location [SRK+03]. Unless the hash table is carefully constructed, it is conceivable that a single node will become the repository for a large amount of information and exceed its limited storage capacity. While central storage of information is advantageous for locating data via a query, the energy expenditure of the affected nodes is much higher than the rest of the network. These "hotspots" inevitably lead to congestion of the transmission medium, premature energy depletion, and failure of the affected portions of the network.

Second, because the hash table must be developed carefully to prevent clustering the network's data in a small number of nodes, the network loses a certain degree of flexibility. In the event the data collected by the network is not as diverse as expected (or if the collected data is beyond the limits of the hash table's capabilities), the hash table will need to be updated to balance the distribution of data stored within the network. Additionally, if the end-user's data requirements change, the hash table needs to be modified accordingly. These hash table updates must be flooded throughout the network to every node, requiring $O(N)$ transmissions. If such updates are frequent, they will quickly erode the efficiencies gained by using a data-centric paradigm.

Third, as the number of events covered by the hash table increases, the size of the hash table must increase as well, thus creating problems of complexity and scalability in dense networks of resource-limited nodes. To combat this lack of scalability, several variants of a *distributed hash table* have been devised [MNR02, RFH+01, RD01, SMK+01, ZKJ01]. Unfortunately, implementing a distributed hash table destroys key ordering; consequently, queries designed to search for near-matches to the desired data cannot be supported [AS03].

Fourth, data-centric networks store related information at common nodes, thus making the network vulnerable to the unrecoverable loss of information in the event of a single node failure. GHT purports to overcome this limitation through the use of a *perimeter refresh protocol* that replicates data at *k* nodes located near the hashed location [SRK+03]. However, the perimeter refresh protocol cannot protect against losses of entire portions of the network caused by enemy action or the environment; such events tend to affect entire regions of co-located nodes versus individual nodes. One solution to this type of failure is to disperse the information throughout the network among non-co-located nodes in a geo-centric-type approach. Another solution implements a balanced-tree approach using skip graphs, such as that proposed in [AS03].

Finally, the data-centric approach is difficult to implement in mobile networks. The introduction of mobility to a sensor network complicates the data-centric requirement to store data at specific network locations. As nodes migrate, they must impart their data to neighboring nodes if the location-data pairing of the hash table is to remain intact; otherwise, queries forwarded to the hashed location will fail to locate the desired information. Depending on the rate of node movement, this data exchange will be costly in terms of total network energy expenditure.

The geo-centric and data-centric approaches are somewhat analogous to Redundant Array of Independent Disks (RAID) modes 0 and 1 in a computer system. The data-centric approach resembles RAID 0 because the storage capacity of the entire sensor network is available for use, and data retrieval latency is decreased. However, there is no inherent protection against data loss in the event of a single disk failure. The geo-centric approach resembles RAID 1 because data is replicated throughout the network, thus providing data redundancy. However, due to data replication at several nodes, the overall storage capacity of the network is decreased.

This is not to say that one approach or the other is superior. The common goal of both the geo-centric and data-centric approaches is to make the query's job of finding the desired information easier, faster, and more energy-efficient. The best approach for a particular wireless sensor network necessarily depends on network characteristics and the specific application(s), as well as the information and latency requirements of the end-user.

2.1.6 Unstructured peer-to-peer networks

Unstructured peer-to-peer networks (UP2P), such as Napster, Gnutella, and KaZaA encompass the general class of Internet file sharing applications in which there is no centralized directory nor is there any attempt to control the placement of data or the topology of the network [LCC+02]. Due to the similarities between UP2P networks and wireless sensor networks employing geo-centric search protocols, they deserve mention here.

Ongoing and relevant efforts to develop efficient replication and search strategies in UP2P networks include [BA05, CS02, GBB+05, GMS05, MNW04]. In contrast to WSN search algorithms, however, the primary focus of these efforts is to reduce query latency versus increasing energy efficiency as the computers in UP2P networks are less constrained by available energy, local storage, and computational capability. However, a key discovery of UP2P research is that the *expected search size* (i.e., the average number of nodes that must be visited to answer a query, averaged over all queries) is minimized when each resource is replicated based on the square-root of its query rates [CS02]. The importance of resource popularity to determine the appropriate number of resource replicates is an underappreciated factor in the WSN search algorithm literature and has the greatest relevance to this research.

2.2 Analytical Approaches to Modeling Search Algorithm Performance

The primary analytical approach used to evaluate the performance of WSN search protocols is a cost-based analysis. A cost-based analysis measures the total number of transmissions made, the total number of useful bits sent, or the total energy expended by the network as a direct consequence of the search algorithm. This approach is favored because it yields useful insight into search algorithm design, yet avoids high degrees of complexity and possible intractability of a mathematical model.

The cost-based approach, though, has several limitations. First, while it provides a means to determine the expense associated with propagating a query or agent through the network, it does not address certain quality of service (QoS) issues, such as any latency requirements of the end-user. Second, a cost-based approach does not ascertain how much traffic the network can support while simultaneously meeting the end-users' quality of service requirements. Finally, determining the design tradeoffs needed to balance the latency and energy expenditure requirements of the network is difficult when using a cost-based analysis. Even so, the cost-based approach has proven to be a useful tool for evaluating the energy efficiency and performance of a search protocol.

The remainder of this section is organized as follows: in the first subsection, a survey of the cost-based approaches in the literature is discussed. The second subsection introduces two node models based on the temporal relationship between agents and queries.

2.2.1 The Cost-based Approach

In their original rumor routing paper, Braginsky and Estrin used a cost-based analysis to demonstrate the energy savings of rumor routing [BE02]. Specifically, their analysis predicted the number of transmissions required to answer a query using rumor routing would be smaller than that required for flooding. Subsequent simulations demonstrated that rumor routing achieved a 98.1% query success rate, yet required only 1/40th of the transmissions required by flooding. They concluded the small increase in unsuccessful queries was acceptable given the substantial reduction in energy expended for transmissions.

Subsequent analyses of various search protocols strayed little from this approach. In 2004, Krishnamachari and Heidemann developed a cost-based analysis of push, pull, and hybrid push-pull networks [KH04] and later derived a closed-form expression for the cost of an optimal expanding-ring search using a modified dynamic programming algorithm [KA05]. A similar method was used to compare two hybrid push-pull query approaches: a structured data-centric storage technique, and an unstructured comb-needle query strategy [KaK06]. (A comb-needle search is accomplished by pushing data to a neighborhood of nodes; these nodes are called the *needles*. Each query is duplicated and subsequently propagated along several simultaneous, parallel trajectories to create a routing structure that resembles a comb. The query is successful when one of the comb's teeth encounters a node with the desired information.) A mathematical model of the energy cost associated with an optimal look-ahead query approach has been developed as well in [SKH03]. The costs associated with pure push and pull query strategies and an optimal hybrid push-pull query strategy have been determined [TYD+04], as well as the costs of the comb-needle query strategy [LHZ04].

The cost-based approach is a popular and effective means for analyzing search algorithm performance. However, it is difficult—if not impossible—to extend the cost-based approach to measure time-based metrics such as end-user quality of service and query latency. This is because cost-based models rely on probabilistic techniques that are not easily manipulated to incorporate time-dependent state information for each node in the network. To achieve this, a more sophisticated node model is required. Section 2.2.2 explores the temporal relationship between agents and queries and describes two models: the *subscription model* and the *non-subscription model*.

2.2.2 The Subscription-based and Non-subscription-based Models

To answer a query successfully in a geo-centric rumor routing network, a node must be the recipient of the query as well as an agent that contains the information sought by the query. Thus, there is a temporal aspect to the agent-query relationship, as wireless sensor networks contain no centralized means to control the arrival order of a query and its corresponding agent at a particular node. It is this temporal relationship between the agent and query that necessitates the definition of two separate models: the subscription model and the non-subscription model.

The non-subscription model assumes the individual network nodes do not retain any information regarding the queries they have processed. When a query is received, the node checks its local event table for applicable information previously received by a corresponding agent. If the information is available, the node answers the query with a *response*. If the information is not immediately available, the node forwards the query to a neighboring node. Therefore, if a query arrives prior to receipt of the corresponding agent, the node will not "hold" the query. While the non-subscription model reduces the storage requirements of the nodes, the probability of a node answering a particular query is reduced.

In contrast, nodes in the subscription model store local copies of queries prior to forwarding the query to a neighboring node. If an agent matching a stored query is subsequently received, the node can send a response immediately. Although this model places a larger storage requirement on the nodes, the probability of a successful query is increased. However, it also increases the likelihood that the sink node will receive several identical responses to its query, causing unnecessary additional energy expenditure by the network.

Regardless of the model used, the storage capacity of each wireless sensor node is limited. Hence, nodes require a policy for managing available resources. The simplest policy to implement is "first in, first out," whereby the oldest agents and queries are removed from memory to make room for newer queries and agents. This policy works well when all events are considered equally important. However, if events have tiered levels of importance, each witness node and querier should assign an *expiration time* to their respective agents and queries. In this case, nodes can assess the utility of stored agents and queries, and those having the least time remaining until expiration can be deleted if necessary to make room for agents/queries with more distant expiration times.

2.3 Design Considerations

Implementing a geo-centric search protocol in a wireless sensor network cannot be accomplished without several supporting algorithms and protocols. Most importantly, nodes must have some means for determining their location within the network. Location information is necessary to enable the geographic addressing structure used to determine the next intermediate hop in the agent/query route. Second, nodes must have an efficient, fair, and effective means to access the transmission medium. This capability is provided by the medium access control (MAC) protocol. Finally, it is advantageous to have an understanding of sensor network routing algorithms. Although certain search protocols, such as rumor routing, have self-contained routing algorithms, several improvements to existing search protocols are based on insight gleaned from these alternative routing protocols.

Although localization, medium access control, and routing are often treated as separate topics, the interactions among these elements of wireless sensor network design are significant. To consider one facet without evaluating its impact on the remaining elements leads to inefficient design. Therefore, Section 2.3.1 proposes five general guidelines for effective wireless sensor network design. Sections 2.3.2, 2.3.3, and 2.3.4 discuss several routing algorithms, medium access control protocols, and routing schemes, respectively; useful performance metrics are also proposed. Although this survey is certainly not exhaustive, the algorithms and protocols highlighted in these sections possess design elements that are commonly found in the literature and have relevance to this research.

2.3.1 Guidelines for Wireless Sensor Network Development

It is difficult to generalize WSN design without first considering the network's intended purpose. Wireless sensor networks must often trade computing power, transmitting range, and power reserves for smaller size, energy efficiency, and lower cost. The purpose of a particular WSN guides the tradeoffs made during the design phase, often leaving little additional capability beyond that needed to carry out the purpose of the network. (Of course, additional capability can be designed into a WSN, but it often requires a commensurate trade in rate of power consumption, node complexity, reliability, and cost.) Despite these limitations, there are several desirable characteristics for WSN design. Although it may not be possible to implement each simultaneously, they provide a basis for analyzing the particular choices and tradeoffs made during the design phase. The remainder of this section proposes five guidelines for design and evaluation of a WSN. Subsequent sections review localization, medium access control, and routing protocols in wireless sensor networks.

2.3.1.1 Energy Efficiency

Energy efficiency is normally the most important factor in the design of a WSN since, in most cases, the useful life of the network is limited by the expected lifetime of the available energy source. Even when sensor nodes have the capability to obtain additional power from renewable sources, the energy available at any given time is still limited and, thus, must be managed with care.

Three activities consume the majority of available power in a WSN: transmitting, receiving, and computing. Transmitting and receiving require the greatest expenditure of energy, with transmission being almost twice as costly as receiving in present-day devices [ROG06]. Computation is relatively cheap by comparison—3,000 instructions can be performed for the same energy cost as transmitting a single bit a distance of 100 meters [TAH02].

In the ideal WSN, nodes consume power for transmitting, receiving, or computing only when necessary to accomplish network functions. If not otherwise required to perform a network function, nodes prefer to enter a low-power state, or sleep mode, to conserve energy. Because computing consumes the least energy of all node tasks, computation at the individual node level should be used whenever possible, especially if such computation can prevent the expenditure of the network's energy resources on more costly activities.

> ***Guideline 1: The ideal WSN conserves energy to the maximum extent possible by ensuring every node is in the lowest possible power state compatible with the requirements of the network's purpose.***

2.3.1.2 Adaptability

Changes in the topology of a WSN are likely to occur even if the network topology is intended to be static. For example, as new requirements arise, additional nodes may be added. Nodes may be redeployed to new locations (or perhaps move autonomously) if the phenomenon of interest is mobile or exceeds the current sensor reach of the WSN. Nodes may also fail unexpectedly due to energy depletion, hardware failure, or harsh environmental conditions. Regardless of the circumstances, a WSN must have the capability to integrate new nodes seamlessly (i.e., it must be scalable), adapt to the challenges presented by node mobility, and recover from node failure when it occurs.

> ***Guideline 2: The ideal WSN is capable of adapting to changes in the network to prevent disruption of the network's service(s).***

2.3.1.3 Localization and Network Topology

If nodes can be added, moved, or deleted from a WSN, it is conceivable that sensor node density will change during the network's lifetime. Additionally, depending on the method used to deploy the nodes, the density distribution of the network will be non-uniform. In most cases, individual sensor nodes can make no assumptions about their own location or the overall network topology immediately after initial deployment.

Awareness of position and network topology provides several advantages for a WSN: first, the location of observed phenomena can be passed to the user to provide a useful context to sensor readings [SRB01]. Second, nodes which have knowledge of the network topology can often optimize the routing of that information, preventing excessive use of energy for transmission. Finally, changes in the topology of a network are often easier to discern and overcome when a point of reference is available.

Unfortunately, individual node knowledge of network topology involves an energy cost. A node must expend energy to determine its initial position and the positions of its neighbors—a process known as *localization*—as well as to conduct periodic updates of this information as nodes are added, deleted, or moved. When employed appropriately, localization and topology discovery ensure the invested energy cost to the network for learning and maintaining this information results in greater energy savings obtained through better management of the network's resources.

Guideline 3: The ideal WSN uses its knowledge of network organization and node location to serve the purpose(s) of the network and to derive greater efficiency in operation.

2.3.1.4 Medium Access Control

The purpose of the MAC in a network is to coordinate access to the transmission medium as well as to prevent and recover from collisions when necessary. MAC protocols perform the same duties in a WSN, but the functions of the MAC are complicated by four factors. First, due to power constraints, transmitters and receivers are not always "awake." In addition to ensuring access to the transmission medium, the MAC protocol in a WSN must also guarantee transmitters are ready and receivers are available at the appropriate times to prevent wasted transmissions. Second, collisions cost energy, both in the colliding transmissions as well as the energy expended for retransmissions. Collisions must be prevented to the maximum extent possible to avoid excessive drain on the network's energy resources [ROG06]. Third, priority may need to be given to certain information depending on the requirements of the network. The MAC must be able to distinguish between priority and normal transmissions and provide appropriate precedence. Finally, the deployed span of a WSN typically exceeds the limited transmission range of its sensor nodes. Hence, several nodes may be able to communicate simultaneously within the network without interference. It is advantageous to permit multiple non-colliding transmissions, so the MAC must manage these multiple transmissions effectively.

> ***Guideline 4: The ideal WSN MAC protocol ensures maximum, timely, and (when necessary) prioritized access to the transmission medium and prevents transmission collisions, thereby reducing unnecessary energy expenditure* [GZR01].**

2.3.1.5 Routing Algorithms

Once the MAC protocol provides a node with access to the transmission medium, the network's routing algorithm ensures delivery of the data to the intended destination. Routing algorithms in a WSN must balance two competing goals: first, they must minimize the total network energy needed to transmit the data to its destination and, second, meet any deadline requirements that may be imposed on the delivery time. When the most energy-efficient route through the network does not meet the network's time requirements, the routing algorithm must adapt to ensure timely delivery.

Because every node in a WSN is a potential router, WSNs are also susceptible to a phenomenon known as *looping*. Looping occurs when a node receives the same packet more than once, fails to detect the duplication, and forwards the packet along the same path as the original packet. If allowed to persist, this behavior creates a never-ending cycle of useless transmissions, a waste of energy resources, and failure of the data to reach its destination.

Guideline 5: The ideal WSN routing algorithm guarantees timely delivery of network data along the most energy-efficient route possible.

2.3.2 Localization and Topology Discovery

"Sensor data without complete coordinates…is next to useless" [SRB01]. This claim is powerful, as it is difficult to devise a WSN application that cannot benefit from location information. In addition to its usefulness to the end user, location information can also doubly benefit the network by simplifying and optimizing routing decisions.

In the following sections, various sources of information useful in localization are discussed, types of coordinate systems used as well as the advantages and disadvantages of each are reviewed, and several localization methods are evaluated based on the guidelines presented in Section 2.3.1.

2.3.2.1 Sources of Location Information

The majority of techniques available to determine a node's location rely on variations of a standard triangulation calculation performed using range measurements from a number of sources located either inside or outside the network. Several sources of range and location information have been explored, including the Global Positioning System (GPS), Time Difference of Arrival (TDOA), Angle of Arrival (AOA), and Received Signal Strength Indication (RSSI).

GPS signals have proven to be a convenient and reliable method for determining location worldwide. Unfortunately, several properties of GPS make its widespread use in WSNs unlikely in the near future. First, GPS signals are low power and do not penetrate solid structures well. WSNs deployed in buildings or environments which do not have unfettered access to the open sky may have difficulty obtaining accurate GPS measurements. Second, the additional hardware needed to receive and process GPS signals is relatively expensive. Since WSNs may have hundreds or thousands of individual nodes, the cost of equipping each node with a GPS device is prohibitive. Finally, the additional hardware complexity added by a GPS receiver also tends to make it an unsuitable choice for reliability reasons.

Although GPS may not be suitable for every WSN, techniques similar to those used to determine position in GPS might be useful to WSNs at the node level. Using the TDOA technique, several "location-aware" nodes in the network can broadcast a time-stamped signal and their location information to the network. If a node receives a number of these signals, it can triangulate its position. However, the relatively short transmission ranges in a WSN would require "synchronization demands of 3 psec per cm of resolution" [SRB01]. Even if such accuracy could be attained across thousands of nodes, the added cost, increased complexity, and high energy expenditure make this an unattractive choice.

AOA techniques, which determine position by using the arrival direction of received signals, suffer from many of the same limitations as GPS and TDOA. Implementation of AOA requires arrays of antennas on each node—an expensive proposition—and additional node complexity.

RSSI techniques determine range information by making use of the principle that transmitted energy levels decrease as a signal travels away from its source. Consequently, if a signal is transmitted at a known power level, the strength of the received signal provides an estimate of the distance between the transmitter and the receiver. If a small number of nodes in the network know their position, range information obtained using RSSI can enable subsequent nodes to determine their own positions. The RSSI approach is appealing because it requires little additional node complexity, uses minimal amounts of computation, capitalizes on normal network traffic, and the additional energy cost to the network is minimized.

Unfortunately, RSSI has several limitations. RSSI measurements have been shown to be far from uniform over time [WTC03], susceptible to fading effects [BM02], and prone to range errors exceeding 50% [MSK+01]. Some of these effects can be mitigated through the use of spread-spectrum technologies [PAK+05]. However, many factors, such as interfering obstructions or irregular terrain within the deployment environment, are typically beyond the control of the network designer. Despite these drawbacks, most proposed localization techniques use some form of RSSI information as the primary means of determining node location and, of all the techniques mentioned, RSSI is currently the method most easily adapted to a general WSN. Localization via RSSI has also been incorporated into the ZigBee specification for wireless networks [Zig06].

2.3.2.2 Coordinate Systems

Three types of coordinate systems are commonly used in WSNs: absolute coordinates, relative coordinates, and virtual coordinates. The choice of a coordinate system is linked to the network's purpose, and this choice also frequently influences the routing strategy.

Absolute coordinates determine a node's location within a defined coordinate system that has meaning outside the network itself (e.g., latitude/longitude). Once nodes determine their absolute coordinates, not only can they determine their location within the network, but also they know their location within the larger system. Absolute coordinates are useful when the user wants specific location information in the context of the environment associated with the collected data. Routing algorithms using absolute coordinates take advantage of the known positions of neighboring nodes to find shortest-distance paths through the network.

Relative coordinates are similar to absolute coordinates except that each node's coordinates only have meaning within the network itself. The axes used in a relative coordinate system are normally defined during the network's startup phase, and the ensuing localization solution results in discovery of the topology of the network. Relative coordinate systems are useful when the location of sensor data inside the network is the only context required. While routing strategies using relative coordinate information are similar to those used with absolute coordinates, the primary advantage of relative coordinates is that there is no need for location information outside the network (e.g., GPS).

When precise location information is unnecessary or cannot be obtained, virtual coordinate systems may be used. Virtual coordinates "locate" nodes using parameters other than physical location or distance information. For this reason, a node's virtual coordinates may change during its lifetime even if the node itself is immobile. Although virtual coordinates cannot be relied upon to provide accurate locations of nodes or observed phenomena, they can be valuable for developing efficient routing algorithms based on parameters such as link quality or packet delivery success ratio.

2.3.2.3 Localization Methods

Most localization methods use some form of RSSI as a means of providing distance information to individual nodes within the network. Due to the inherent problems associated with RSSI, proper evaluation of these localization techniques must answer the following questions: how does the algorithm overcome the range error of RSSI to determine an accurate location, how does node mobility affect the solution, and what is the network energy cost in terms of startup and maintenance?

2.3.2.3.1 Overcoming RSSI Errors in a Mobile Network

RSSI range errors due to fading effects can be reduced by taking a large number of signal strength measurements and averaging the samples over a large time window [BM02]. However, finding accurate positions of mobile sensor nodes is best accomplished using a small time window to reduce errors introduced by the node's movement (i.e., older measurements are less likely to indicate the node's present position). The difficulty lies in finding a sampling window which effectively reduces the location error due to fading while still providing an accurate position under mobility. Analytical solutions to this problem would be exceptionally difficult to solve, but simulation can provide insight into the optimum window size.

The network simulation consisted of 20 uniformly distributed nodes placed on a 100m by 100m square with two beacons positioned at opposite ends of one side [BM02]. Beacons transmit signals at a known power level, and each node uses a triangulation calculation to determine its location based on the received signal strength of the beacons. Under the best circumstances in a static network, location can be determined within 2.5m of the actual node position using a window size of 50 samples. Although larger window sizes yield marginally better accuracy, the error in the position calculation cannot be eliminated completely.

Once mobility is introduced into the simulation, the outcome is predictable: larger window sizes and higher node velocities result in larger position errors. Interestingly, the best results in this mobile network are also obtained using a window size of 50 samples; however, the position error at even the smallest node velocities is always at least twice as great as that of the stationary network. Higher rates of mobility yield even larger errors. Based on this analysis, there is a "window-size tradeoff when both fading and mobility are considered" [BM02].

The results of this simulation provide useful insight into locating mobile nodes using RSSI techniques, but there are additional obstacles in real-world WSNs. First, two beacons are sufficient in this simulation because the nodes are restricted to a well-defined two-dimensional area. In actual deployment, nodes may not be aware of the network's span and will likely be deployed in three dimensions. Consequently, optimum placement of beacons is not guaranteed, and additional beacons would be required for nodes to determine their location. Second, unless the network's requirement is limited to a determination of the network topology (e.g., using relative coordinates based on beacon positions), each beacon must have some method of determining its true location. The exact method must be chosen prior to network deployment. Finally, once each node in the network calculates its position, future updates should be performed only if the network's requirements or operation will be adversely affected by subsequent topology changes; updating more frequently uses energy resources unnecessarily. While several solutions to the first two issues come to mind (e.g., deploy additional beacons, use relative coordinates or GPS, etc.), the third problem requires some manner of alerting the network to topology changes. One such method is proposed in Section 2.3.2.3.3.

2.3.2.3.2 Determination of Relative Coordinates

If GPS or other external localization solution is unavailable to the network but some method for identifying relative node position is required, local topology can be determined using the Assumption Based Coordinates (ABC) method [SRB01]. In the startup phase of ABC, one node defines its position as the origin of the network. This origin node broadcasts a message, and the straight-line path between the origin node and the first node to respond is defined as the network's positive x-axis. The second and third nodes to respond define the positive y-axis and z-axis, respectively, in the same manner. All remaining nodes then determine their location using the coordinate system defined by these four nodes.

RSSI is the most commonly used method for determining distance information in ABC applications. However, if RSSI is used for determining distance between nodes, any error in measurements made by the first four nodes will affect the entire coordinate system, and position errors will multiply rapidly throughout the network. One proposal for improving ABC is Triangulation via Extended Range and Redundant Association of Intermediate Nodes (TERRAIN). TERRAIN implementations of ABC require no less than four independent anchor nodes in the network, and each node uses at least four anchor node transmissions to determine its position. After several iterations of TERRAIN, node positions have been found to be accurate within 5% [SRB01].

2.3.2.3.3 Node Awareness of Mobility

In mobile environments, a significant portion of a node's energy is spent monitoring the network for topology changes. It has been noted that "more than 90 percent of energy is spent on channel monitoring when nothing is happening," and "nodes' mobility can be a big sink of energy" [GZR01]. For example, in one particular channel-oriented MAC protocol, node knowledge of the local network topology is critical to network operation. The protocol requires each node be assigned a different transmission channel than any of its two-hop neighbors. If outdated neighbor information is used, overlapping channel assignments could be made, and collisions would result. Although energy efficiency suffers if nodes constantly monitor the network for updates, the protocol fails if nodes possess inaccurate neighbor tables.

The solution to the problem is to ensure each node is aware of its own mobility and to require mobile nodes alert neighboring nodes when changing position. Using "either an embedded processor or input from upper layer applications," nodes which detect their own movement transmit an alert signal over a "wake-up" channel, causing all nodes within range to wake up and update their neighbor table information accordingly [GZR01].

2.3.2.3.4 Localization without RSSI

Although taking RSSI measurements from several different sources can reduce position error to as little as 5% [MSK+01], it may be impractical to make a large number of RSSI measurements, or nodes in a particular network may not be equipped to make such measurements at all. In either case, a node can still estimate its location using other means as long as exact precision of node location is not required.

One of the simplest methods for estimating position is for each node to assume it is located somewhere between all nodes within its reception range. For example, a network could be deployed with several position-aware reference nodes which periodically transmit beacon signals to the network. Once a node receives a sufficient number of these beacon signals, it calculates its position as the centroid of the received reference positions. Although this method is not meant to provide precision coordinates, experimental results indicate over 90% of nodes randomly placed on a 10m by 10m square could be located within 3.0m of their actual position [BHE00].

A variation of the centroid localization method uses a link estimation technique to determine virtual coordinates for nodes [WTC03]. In this case, nodes monitor network transmissions to determine the probability of successful communication with neighboring nodes and then calculate a value representing the quality of each link. These values are based on a windowed average, so older, less frequent transmissions—indicating a node has failed or moved out of range—result in lower link quality estimations and are eventually dropped from the node's location calculations. The final result is a coordinate system in which nodes with the highest probability of successful communication are "closer" in virtual proximity.

2.3.2.4 Metrics for Evaluation of Localization Algorithms

Evaluation of the suitability of a localization algorithm for a particular network is application-dependent, but the following metrics will help the network designer make a comprehensive analysis:

Position Error. Position error is the most commonly used metric of performance for localization algorithms. It is calculated by finding the difference between a node's actual and calculated locations.

Time Required to Achieve Desired Position Accuracy. Most localization methods achieve greater accuracy if nodes are allowed to perform multiple iterations of the algorithm. If the network has a specific requirement for location accuracy, this metric can be used to determine how much time and/or number of iterations needed for each node's position to achieve the desired level of accuracy.

Total Network Energy Required for Localization. Localization processes require network energy resources both for initial location discovery and for location maintenance. Additionally, node triangulation calculations use energy for computation. Total Network Energy Required for Localization is calculated by determining the amount of network energy required to calculate each node's initial position to the desired level of accuracy as well as the energy expenditure necessary to update that information throughout the network's lifetime. Unfortunately, with the exception of [JBR+07], little of the literature addresses the energy requirements for localization, possibly indicating an area of future study.

2.3.3 Medium Access Control

A common sense approach to MAC design for a WSN would ostensibly begin with the successful IEEE 802.11 protocol for wireless ad hoc networks. It seems plausible that 802.11 could be adapted to a general WSN since the networks appear, at least on the surface, to be similar. However, there are several reasons why this protocol is unsuitable for sensor networks including: the number of nodes in a sensor network can be orders of magnitude greater; denser deployment of nodes; occurrence of node failure; frequent topology changes; broadcast versus point-to-point nature of transmissions; and limited power, computational ability, and memory capacity of individual nodes [ASC02].

In addition to the stated differences between WSNs and their wireless network counterparts, much of networking literature discusses medium access control mechanisms and routing algorithms as if they are inseparable. In the case of wired networks and networks based on 802.11, the reason is apparent: once access to the transmission medium is obtained, packets are normally transmitted along the same route or to a common access point for routing and delivery. Wireless sensor networks defy this traditional approach because they operate in an uncertain environment. Due to short transmission ranges and power concerns, neighboring nodes must often be used to route data to its destination, and the operational status of a neighbor can change from one moment to the next. This distinction permits a clear separation of the duties of the MAC protocol and routing algorithm in WSNs. Whereas the MAC guarantees access to the transmission medium, the routing protocol is responsible for ensuring accurate and timely delivery of the information. With this characteristic of WSNs in mind, the following section provides a discussion of various methods for ensuring node access to the transmission medium.

2.3.3.1 Comparative Analysis of Selected MAC Protocols

The challenge facing the MAC is to ensure each node has the opportunity to access the transmission medium even as several other nodes may simultaneously compete for the same privilege. Additionally, the MAC protocol must be aware of the amount of energy expended by the network and minimize energy consumption whenever possible while still meeting the requirements of the network's purpose. The nature of WSN transmissions might lead one to assume that nodes should simply transmit their data in broadcast fashion (e.g., as used in an ALOHA network) with the hope that the packet will be successfully received and subsequently retransmitted by neighboring nodes until it ultimately reaches its destination. Unfortunately, the simplicity of this approach is overcome by the fact that dense networks of nodes quickly overwhelm the network (much as occurs in ALOHA with a large number of transmitters), resulting in a waste of network energy and high probability of delivery failure. WSNs therefore require a more sophisticated approach.

One such approach is a multi-channel MAC optimized for low-power, distributed operation in WSNs [GZR01]. Implementation of this multi-channel MAC requires each node to select a communication channel that differs from those chosen by its one- and two-hop neighbors. A node announces its choice of channel by transmitting a Channel Assignment Packet (CAP) as well as the contents of its own Channel Assignment Table (CAT) on a common channel to all of its one-hop neighbors. The CAT contains a record of each node's one-hop neighbors' communication channels. Receiving nodes add the CAP and CAT information to their own tables, eventually resulting in complete knowledge of channel assignments for each node's two-hop neighbors. Based on this information, a node can ensure its choice of communication channel is unique.

The advantage of the multi-channel MAC is nodes may transmit freely over their chosen channel without the threat of collision. Collisions are prevented since hidden and exposed nodes are prevented through unique channel assignments. However, unless the network density is carefully managed or the number of available channels is large, dense networks can quickly exceed the channel capability of the sensor node hardware.* Also, although the protocol uses less energy per bit transmitted than "traditional radio protocols," there is no indication the transmission requirements for transmitting and maintaining the CAP and CAT information between nodes is taken into account. Finally, if nodes are mobile, they need to exchange CAP and CAT information more often or risk conflicting channel assignments. The required frequency of these updates, as well as the energy expended maintaining an accurate CAT under mobility, is still undetermined but certain to be significant.

If sufficient transmission channels are not available to a WSN, multi-channel MACs are impractical, and other means of accessing the medium and preventing collisions are required. Since random access to the transmission medium is prone to collision, efficiencies might be obtained by having nodes exchange their transmission schedules in advance. Such schedule-based protocols normally require far fewer channels than multi-channel MACs, and they prevent collisions through deconfliction of transmission schedules. One such schedule-based protocol is sensor-MAC (S-MAC) [YHE02].

S-MAC adopts 802.11's success in dealing with the hidden node problem, yet applies several WSN-specific optimizations to overcome the energy inefficiency of 802.11. Most of the energy inefficiency in an 802.11 network occurs because nodes continually monitor the channel for traffic; sensor nodes, however, do not have the

---

* The actual number of channels required is $\left[d\left(d-1\right)+1\right]$, where d is the maximum number of neighbors each node can have.

energy stores to do this. If these idle-listen periods could be eliminated, energy consumption can be reduced by 50% or more [YHE02].

S-MAC begins by having each node listen for sleep-wake scheduling information from its neighbors for a given period of time. If a node overhears a schedule from one of its neighbors, the node adopts the neighbor's schedule, rebroadcasts the schedule, and then enters sleep mode until the scheduled wake-up time. If a node does not overhear another schedule, it chooses its own schedule, broadcasts that schedule, and then enters sleep mode. Nodes which overhear another node's schedule after choosing their own schedule adopt both schedules.

The result of this exchange of sleep-wake schedules is clusters of nodes guaranteed to be awake and listening to the transmission medium at the same time. Consequently, S-MAC overcomes the problem of ensuring the intended receiver is awake and ready to receive messages from a neighboring node when needed. For node-to-node transmissions, the successful collision-avoidance Request-to-Send/Clear-to-Send (RTS/CTS) scheme of 802.11 is used.

S-MAC is a practical evolution of 802.11 adapted to WSNs, and the simplicity of the approach means it could be tailored to a wide array of applications. However, S-MAC suffers from latency issues as a result of random sleep scheduling, reducing its ability to guarantee delivery to the user within a specified period of time. Also, although S-MAC has provisions for nodes to re-enter sleep mode when they sense neighbor nodes are transmitting to other receivers, additional energy efficiency might be gained if nodes were to exchange their transmit-receive schedules (as opposed to the sleep-wake schedules used in S-MAC) in advance. The Traffic Adaptive Medium Access protocol (TRAMA) attempts to optimize S-MAC in exactly this manner [ROG06].

TRAMA claims significant energy savings over contention-based protocols such as Carrier Sense Multiple Access (CSMA) and 802.11. In deployment, TRAMA requires nodes to determine their desired transmission schedules in advance, exchange these requirements with each neighbor, and enter low-power sleep mode when not needed to transmit or receive. TRAMA claims superior energy savings by providing a deterministic method for permitting nodes to enter a low-power sleep mode. Additionally, nodes with scheduled transmissions are free to send their packets without collision, and the appropriate receiver node(s) will be awake and ready to receive the incoming data.

Implementation of TRAMA requires a time-slotted channel with two different types of slots: *signaling slots*, which are contention-based and random access; and *transmission slots*, which are guaranteed to be collision-free. Signaling slots are used for nodes to exchange one-hop neighbor information, as well as to add or delete nodes from the network. Because multiple nodes may try to access the channel simultaneously during a signaling slot, retransmission is used to overcome collisions between nodes. Transmission slots are used for previously-scheduled transmissions and for nodes to exchange their scheduling requests for the next transmission slot. If two or more nodes try to schedule the same time slot, the affected nodes will apply an *Adaptive Election Algorithm* to determine which node will be permitted to send its data. Since each node is aware of the Adaptive Election Algorithm, nodes can independently determine which node "wins" a particular slot; additional transmissions between nodes are unnecessary to resolve these conflicts.

As might be expected, TRAMA has a high delivery ratio due to its collision-free transmissions, but it experiences high queuing delays as a consequence of its scheduling requirements. Also, although the authors claim greater energy savings due to nodes being able to determine when they may enter sleep mode in advance, every node must be awake during each signaling slot (or risk out-of-date one-hop neighbor information) as well as during part of each transmission slot (to receive and/or exchange transmission schedules with other nodes). As a result, TRAMA has an average node sleep cycle of 87% (i.e., each node sleeps 87% of the time). This is in contrast to much of the literature which claims sleep cycles closer to 99% or higher are generally necessary for energy conservation and long network life [Cla04].

2.3.3.2 MAC Performance Metrics

Perhaps the most difficult part of assessing the utility of a specific MAC protocol is the absence of standardized network topologies and widely-accepted metrics. Each proposal tends to be evaluated using a diverse set of metrics and different network topologies for simulation and experimentation, making "apples-to-apples" comparisons between protocols nearly impossible unless each is examined independently. Additionally, many commonly-cited MAC performance measures are often affected by the performance of other aspects of the network outside the scope of the MAC, making it difficult to determine a MAC protocol's true efficiency. Ideally, metrics provide an accurate measure of MAC performance regardless of the network's choice of routing algorithm or localization method. With these issues in mind, the following metrics were deemed as most useful for evaluating MAC performance:

Network Energy Expended per Successful Packet Transmission. A measure of the energy efficiency of a particular protocol, this calculation includes not only the energy required for successful transmission of a single packet, but also the energy expended in retransmissions due to collisions, node listening/receiving (i.e., by all active nodes within range of the transmitter which could otherwise be in sleep mode), and node computations. By definition, MAC protocols which avoid/prevent collisions, ensure only the targeted receiver(s) are awake, and require the least computation are deemed the most efficient by this metric. This metric is a more comprehensive variation of the EPB (energy per useful bit) metric used in [GZR01].

"Goodput." Goodput is defined as "the ratio of the total number of packets received by the observer to the total number of packets sent by all receivers within the simulation time" [TAH02]. Goodput is a variation of the *Throughput* metric with the exception that only useful (i.e., no duplicate packets or retransmissions due to collisions) packets are counted.

Maximum Node Density Capability. A measure of a MAC protocol's ability to manage dense networks, Maximum Node Density Capability is determined by finding the maximum number of one-hop neighboring nodes which do not cause the MAC to exceed its management capabilities, node memory capacity, or network latency requirements. As an example, the density of nodes in a multi-channel MAC is limited by the total number of channels available to the network. Other MAC protocols might be limited by different factors, such as the amount of memory available to maintain neighbor tables. In

WSNs where latency is a concern, an increasing density of nodes may cause longer network delays (such as might be experienced in a schedule-oriented MAC when larger numbers of one-hop neighbor nodes require more transmission time to exchange schedules). In these cases, Maximum Node Density Capability would be limited by the maximum acceptable delay. The goal is to determine which factor places the most restrictive limit on network density and to find the upper bound of that limitation.

MAC Latency. A measure of the latency of a MAC protocol is the average time required for a node to gain access to the transmission medium once it has a packet to send. When calculating this value, the effect of transmission collisions should be included such that the metric accounts for the time needed for a node to gain uncontested access to the medium and transmit successfully. Hence, schedule-based MACs will usually have a deterministic latency, yet latency for collision-avoidance MACs (e.g., S-MAC) must include the probability of collision and retransmission in their calculations.

Scalability [TAH02]: A MAC's scalability determines an upper bound on the total number of nodes that can be managed by the MAC and still meet network performance requirements. This metric is similar to Maximum Node Density Capability, but Scalability determines the MAC's upper bound on the size of the network.

2.3.4 Routing Algorithms

After a node is granted access to the transmission medium, its transmission is limited to its neighboring nodes. A node's intended target will not always be within transmission range, so WSNs must have some means of relaying messages from node to node. Complicating this problem is the distributed nature of WSNs. Because there is no centralized router in a WSN (as would be found in most wired and 802.11 networks), nodes must decide independently how to forward a message to its destination. This section discusses various methods for routing a packet to its destination within a WSN.

2.3.4.1 Comparative Analysis of Routing Protocols

One of the simplest routing methods available requires a node to broadcast its message to all neighboring nodes, have each recipient rebroadcast the message to its neighboring nodes, and repeat the process until the entire network has heard the message. Known as *flooding*, the strongest advantage of this routing method is that it guarantees delivery of the message to the intended receiver with the shortest delay even in networks with rapidly-changing topologies. However, to be effective, it requires all nodes within transmission range to be on and listening prior to each transmission. Since transmitting and receiving use the greatest amount of energy in a WSN, the flooding technique expends a large percentage of network energy repeatedly transmitting messages to portions of the network that probably have no use for the information. While the ideal WSN routing algorithm delivers messages with the speed and robustness of flooding at a small fraction of the energy cost, alternatives to flooding generally require a trade in latency and reliability for energy efficiency.

2.3.4.1.1 Dynamic Source Routing

The most basic requirement of a routing algorithm is to determine a reliable path from the sender to the destination. Although intermediate receivers in the route might be determined dynamically at each node, Dynamic Source Routing (DSR) makes the sending node responsible for finding the entire network path in advance [JM96]. The sending node accomplishes this by inserting a complete route into each packet's header and then transmitting the packet to the first intermediate receiver. Intermediate receivers use this routing information to forward the packet until it finally reaches its destination.

Application of DSR requires each node to maintain a *route cache*—a table of working routes to various destinations in the network. In the event that a node does not have an entry in its route cache for a particular destination, it will search for one using a process called *route discovery*. Route discovery requires a node to broadcast a *route request* message to the network. As each node receives this route request, it appends its own address to the message and rebroadcasts the request. Once the request finally reaches the destination, the destination node forwards the resulting address list contained in the route request back to the original sender in a *route reply*. The sender now has a working route to the destination.

Since WSN topologies are dynamic, nodes may try to use a previously-successful route only to have that route fail. In this case, the intermediate node which discovers the transmission failure sends a *route error* message back to the sender. The sender modifies its routing cache with the updated information and initiates a new route request.

In the interest of energy efficiency, several optimizations can be made to the basic DSR algorithm [JM96]. First, by analyzing the information contained in route reply messages overheard from other nodes, intermediate nodes can discover new routes as well. Learning new routes in this manner prevents repetitive route request messages from flooding the network. Second, route replies may also indicate shorter paths to intermediate nodes that were previously unknown. When such routes are found, a node updates its route cache accordingly. Third, the probability of finding the shortest route to a destination is improved by introducing a small transmission delay prior to the transmission of a route discovery packet; the length of the delay at each node is based on the number of hops in the route (i.e., longer address lists will experience longer transmission delays). Shorter routes will, therefore, propagate faster through the network and back to the requester. Finally, data can be piggybacked on route requests to reduce the total number of packets transmitted throughout the network.

Overall, DSR uses less total network energy than flooding, especially when the network topology is fairly constant or changes slowly. It operates well under most conditions with a low packet overhead; however, appending the entire route to each message causes a high byte overhead [BMJ+98]. DSR also outperforms most ad hoc network routing algorithms in mobile networks. Simulation indicates DSR is capable of delivering more than 95% of packets successfully at average node speeds of up to 10 meters per second [BMJ+98]. Finally, if a node has a good route stored in its route cache, delivery latency is predictable, although not guaranteed to be minimized (because cached routes are not certain to be minimum routes). However, latency will be several times higher when a route fails and/or a node must initiate a route request.

2.3.4.1.2 Minimum Hop Routing

Determining the minimum-hop route from sender to receiver (which often corresponds to the minimum energy route) is important from a power management perspective in WSNs. However, if the minimum energy route is unreliable, energy savings can be eroded quickly by the necessity for retransmissions. If nodes could measure the quality of the links between themselves and their neighbors, greater energy savings might be obtained by favoring routes with better transmission characteristics.

One such technique for determining link quality between nodes is known as *link estimation* [WTC03]. Initially, each node "snoops" on its neighbor's transmissions and, based on the link sequence numbers observed in each packet, is able to determine the reliability of a particular link. Through the application of a new estimator, the Window Mean with Exponentially Weighted Moving Average (WMEWMA), each node computes an average transmission success rate over a given time period for each neighbor. The result is a neighborhood table populated with link quality estimations assigned to each neighboring node. However, node memory limitations make it unlikely that sensor nodes are capable of maintaining link quality information on every neighbor, especially in dense networks. For this reason, nodes use an adaptive down-sampling technique either to reinforce neighborhood table entries or to discard them for higher quality links (where the probability of a new link being inserted in the table is based on the ratio of the neighbor table size to the number of neighbors).

Before a node decides which neighbors are best suited for routing, one qualification about each node's neighborhood table must be made: the data gathered to build a neighborhood table is based solely on signals *received* by each node. Since links are not necessarily bidirectional, no assumptions can be made about the quality of the link in the other direction. For this reason, nodes are required to exchange their link estimates with neighboring nodes periodically so each node can determine the quality of its own outgoing transmissions across each link.

Once link estimates are made by each node, a variation of the distance-vector algorithm is used for routing. Distance-vector routing sends packets along routes with the "lowest cost." In this case, link quality estimations are used to determine the cost of each hop in the route, resulting in determination of the most reliable route. When link estimation is used to determine high quality transmission links in this manner, experiments indicate a high probability of successful end-to-end transmission at the expense of a slightly higher hop count (versus other minimum hop protocols).

Using link estimation for routing decisions makes sense from a reliability perspective, but routing techniques in WSNs must also be concerned with energy efficiency. Energy consumed during routing is more than just the energy used to transmit a packet from sender to receiver; it also includes the energy expended to maintain the data tables used for routing decisions. Link estimation requires each node to spend much of its time listening to the transmission medium, computing link estimates, and exchanging neighborhood tables with nearby nodes. Each of these activities has a significant energy requirement but, unfortunately, the cost of these route maintenance activities is not addressed.

A final unexplored aspect of link estimation is the performance of the algorithm under conditions of node mobility. Although performance under mobility has not been determined directly, use of the WMEWMA estimator results in increasingly lower link estimation values for links that experience a drop in quality (e.g., as nodes move apart). Thus, over a period of time, link estimation would probably adapt to a mobile topology, but the exact responsiveness of the algorithm has not been investigated.

2.3.4.1.3 Geographic Routing

Most routing algorithms in WSNs use some form of geographic information to determine the node-to-node transmission path from sender to destination. Since many WSN applications already require each node to determine its actual position, using this same location information for routing makes sense for energy efficiency. Taking this approach prevents the network from spending additional energy resources supporting a routing algorithm that depends on information other than location (e.g., link estimation).

At a minimum, for a node to forward a packet using geographic routing, it needs to know the locations of each of its neighbors as well as the destination. Once this information is known, intermediate nodes forward packets to the neighboring node closest to the final destination. However, depending on the topology of the network, a point may be reached in which a node has no neighbors closer to the destination than itself. In this case, the only option is to forward the packet to a node further away from the destination. Greedy Perimeter Stateless Routing (GPSR) defines how a node should choose the next hop when this situation occurs [KK00].

The first step in GPSR determines network connectivity in terms of a planar graph (i.e., a graph in which no two edges cross) yet maintains the connectedness of the network such that there is still a path from each node to all other nodes. Two types of planar graphs, the Relative Neighborhood Graph (RNG) and the Gabriel Graph (GG), meet these requirements.

Once the overall node-to-node connectedness is determined by finding the RNG or GG of the network, nodes transmit only to neighbor nodes with which they have a defined connection. Routing is accomplished as previously described; nodes choose the next transmission recipient as the neighboring node closest to the final destination.[†]

[†] The reader should note that the set of nodes available for reception in the RNG- or GG-connected network is probably smaller—and can never be more—than the total number of nodes actually within a given node's transmission range.

If a node is subsequently unable to forward a packet because none of its connected neighbors are closer to the destination, the packet enters *perimeter mode*. In perimeter mode, packets are forwarded around the face of the perimeter of the problem area by choosing the next available path using the right-hand rule (i.e., the next path located sequentially counterclockwise from the packet's arrival edge). After transmission, the receiving node checks the locations of its connected neighbors and determines whether the packet can be returned to normal routing or must remain in perimeter mode for the next hop. Since it is possible for a packet to enter a loop by being repeatedly forwarded around the same perimeter, nodes must have some means of recognizing this repetition. GPSR places a pointer in the packet identifying the first link traversed upon entering perimeter mode. When a node recognizes that a packet is attempting to traverse the same link twice, delivery is deemed impossible, and the packet is dropped.

Based on the results of network simulations with mobile nodes, GPSR successfully delivers nearly 97.5% of all packets at node speeds of up to 20 meters per second [KK00]. Of those packets successfully delivered, 97% are delivered along optimal-length paths. Comparing the performance of DSR and GPSR in this scenario, DSR's delivery success rate is nearly the same as GPSR. However, DSR delivers only 84.9% of packets along the optimal path; this is a result of DSR's use of cached routes which are not updated until a route terminates with a route error [KK00].

The primary disadvantage of GPSR is that each node's neighbor table must be updated on a periodic basis to maintain the overall network graph. Consequently, the level of maintenance-oriented traffic for GPSR routing is constant without regard for whether or not the network topology has changed. In immobile or nearly-immobile networks, GPSR's energy expenditure would be difficult to justify given that other routing algorithms perform comparably yet use much less energy. In contrast, DSR's level of traffic for routing maintenance is low unless the network topology changes significantly enough for a route to fail. As node mobility increases, DSR's maintenance overhead increases significantly as nodes attempt to recover broken routes.

2.3.4.1.4 Routing Algorithm Performance Measures

As stated previously, the purpose of a routing algorithm is to deliver network data to the intended destination in a timely, efficient, and reliable manner. Consequently, appropriate measures of routing algorithm performance must be capable of capturing these requirements. The following metrics provide appropriate means for measuring and comparing the performance of WSN routing algorithms.

Routing Energy Efficiency. The energy efficiency of a routing protocol is calculated by determining the total network energy expended using the optimum energy-efficient route and dividing by the energy expended using the chosen route. Energy calculations include the energy used for each transmission, energy expended for nodes to be awake and ready to receive transmissions, and node energy requirements for calculations. Energy expended due to collisions should not be included here as these effects are an indicator of the efficiency of the MAC protocol.

Routing Latency. Latency is normally calculated as the total delay from the moment a node has data to send until the data reaches the destination. Depending on the application, latency may also include the amount of time necessary for a network to answer a query (i.e., time between when the initial request is made and when the answer is delivered to the requester). If latency is calculated in this manner, the metric will include the effects of medium access delay due to the MAC protocol. For a true comparison of routing algorithms, any latency due to the MAC (as described in Section 2.3.2.2) should be subtracted from the total delay from sender to receiver.

Delivery Failure Ratio [KK00]. Delivery Failure Ratio is calculated by determining the number of deliverable packets either dropped or lost (due to looping, dead ends, or other routing failure) divided by the total number of deliverable packets sent. The Delivery Failure Ratio should be calculated under various rates of network mobility. Although higher losses are expected as networks become increasingly mobile, the Delivery Failure Ratio should ideally be zero for non-partitioned immobile networks [KK00]. This metric is also an implicit measure of the reliability of the routing algorithm.

Energy Required for Route Maintenance. This metric is calculated by determining the total amount of network energy expended to maintain the necessary state information for routing. For accurate comparison of routing algorithms between networks of varying sizes, it may be advantageous to determine this value over a period of time per node (e.g., joules per second per node).

2.4 Summary

This chapter provided an overview of several different types of wireless sensor network search algorithms, as well as an introduction to the principal analytical techniques used to study search algorithm performance. Additionally, five general guidelines for efficient wireless sensor network design were introduced. The importance of localization, medium access control, and routing to search algorithms was explained. Relevant details of several localization algorithms, medium access control protocols, and routing algorithms were also presented.

3. Methodology

The purpose of this chapter is to summarize the research goals of this dissertation, identify the scope of the research, provide justification for specific assumptions that are made, and offer a general outline of the tasks to be accomplished.

3.1 Problem Definition

Future wireless sensor networks are likely to be highly-dense networks composed of thousands, hundreds of thousands, or even millions of nodes. Additionally, to contain the costs associated with deploying these networks, they will continue to be populated by low-cost, unreliable, power-limited nodes. As a consequence of this unreliability and the requirement to deploy these networks in harsh environments where partial destruction of the network may occur with high probability, future search algorithms should be designed to enhance the survivability of data collected by the network. Consequently, there is a need for energy-efficient, reliable, and scalable search algorithms. Within the design space of high-density, large-population networks, current WSN search algorithms fail to meet this need.

Additionally, no research has been found that analytically determines the number of resource replicates that must be created per witnessed event to achieve energy-efficient search algorithm performance when both resources and queries have limited lifetimes. To fill this void, an analytical model of WSN nodes is developed and extensively analyzed via mathematical programming formulations. The results of these analyses are compared to observations obtained via simulation experiments.

3.1.1 Research Goals

General statements of the goals of this dissertation were summarized previously in Section 1.3. These goals are now restated with additional detail:

1. Develop an energy-efficient, reliable, scalable, small-footprint search protocol to promote the survivability of network data in the event of partial loss of the network. Determine the optimum parameters for this search protocol by deriving an analytical model of the expected total energy expended by the network to accomplish the following activities: advertising a resource's availability to a subset of the network's nodes, locating the resource via subsequent queries, and returning the response to the requesting node.
2. Develop an analytical model of a WSN node that determines the appropriate number of resource replicates to be created per witnessed event when resources are lifetime-limited and queries are time-constrained. The appropriate number of replicates created per event is determined by minimizing the total energy expended by the network while ensuring the total proportion of query failures does not exceed a specified threshold.
3. Determine the accuracy of the analytical node model developed in (2) to predict search algorithm performance in large-scale networks. Evaluate the effects of specific parameters, including transmission power/range and agent/query lifetimes, on system performance.

3.1.2 Approach

The first goal of this research requires the development of a new search protocol to overcome the deficiencies of current approaches. Most importantly, an analytical model of this search protocol is derived to permit the protocol parameters to be optimized via a mathematical programming formulation to achieve minimum expected total energy expenditure. The protocol should enhance the survivability of data within the network; hence, this research focuses on geo-centric search algorithms rather than data-centric approaches for the reasons stated in Sections 2.1.4 and 2.1.5. Additionally, it is assumed nodes requesting information have no prior knowledge of the location of a particular resource (i.e., nodes conduct a "blind" search). The intersections of resource advertisements and requests are, therefore, events that can be modeled probabilistically; hence, the development of the analytical model relies primarily on probability theory. This phase of the research assumes resources and queries are persistent, i.e., resources and queries do not expire.

The second goal extends the previous research by optimizing parameters for a random walk search protocol which incorporates expiration times for both resource advertisements and requests. Due to the introduction of expiration times, the state of each node is now time-dependent, and probability theory no longer adequately models the temporal behavior of the search protocol. However, queueing theory and Markov chains provide relatively straightforward means to model the arrival of resources/requests to each node, as well as the loss of resources/requests via transmission or expiration. The state of each node can be sufficiently captured by tracking the total number of agents stored in each node's event table in addition to the total number of agents and queries in each node's transmission queue. Once the analytical node model is derived, it is optimized to achieve energy efficiency and to ensure the total proportion of query failures does not exceed a specified threshold.

The third and final goal of this research validates the analytical node model's ability to predict search algorithm performance in networks with large node populations. This is important for two reasons. First, analyzing state information for every node in a large-population network is computationally demanding and therefore unsuitable for direct implementation in wireless sensor networks. However, the analytical node model may provide the capability to determine the mean performance of the network and, consequently, the potential to optimize the network's parameters without the need for extensive computation. Second, in large networks, the actual distribution of interarrival times of agents and requests may differ from those assumed by the analytical model. The degree and magnitude of the resulting performance differential, if any, between the analytical node model and the network must be determined. Since the purpose of this phase of the research is to investigate the actual performance of large-population wireless sensor networks, simulation is the appropriate means to obtain the necessary data.

3.2 System Boundaries

The system under test (SUT) consists of the nodes populating the wireless sensor network in which the search protocol is implemented; the component under test (CUT) is the search protocol. There are several sources of energy expenditure in a wireless sensor network, including the energy expended to initialize and maintain localization information, routing tables, and sensor data; transmission/timing synchronization; and computation. However, the energy expenditure associated with these activities is highly dependent on the selected protocols and the hardware characteristics of the nodes. Necessarily, analysis of the SUT will be limited to the energy expended by the network as a direct consequence of the search protocol itself, namely the total energy expended to advertise resources, answer queries, and return responses.

3.3 System Services

Wireless sensor networks are capable of providing a wide variety of services. In general, however, these services can be broadly characterized into one or more of the following categories:

- Monitor environmental phenomena and provide reports upon the detection of specific events or, alternatively, provide sensor readings at predetermined time intervals.
- Store data related to specific events.
- Use distributed computation to solve problems that are beyond the limited capabilities of a single node.
- Execute specific applications in support of the network's objectives.
- Answer queries related to information stored by the network.

Search protocols in wireless sensor networks support these network services by facilitating the answering of queries. To perform this function in an energy-efficient, scalable, and reliable manner, search protocols must execute specific tasks. These search protocol-specific tasks, as well as possible outcomes and results, are summarized in Table 1.

Table 1: Search protocol tasks, possible outcomes, and results.

Task	*Possible Outcomes*	*Result(s)*
Ensure each resource is advertised to an appropriately-sized subset of the network's nodes	Network is informed at the appropriate level	Protocol is energy efficient
	Network is under-informed	Increased energy expenditure and time required to locate the resource
	Network is over-informed	Increased energy expenditure required to advertise the resource; network's aggregate storage capacity is unnecessarily consumed
If an uninformed node receives a query, forward the query to a neighboring node (or a subset of the neighboring nodes)	Query is correctly forwarded	Protocol is energy efficient
	Query is incorrectly forwarded	Increased energy expenditure and time required to locate the resource
	Query is not forwarded	Query fails; increased energy expenditure and time required to reissue the query and locate the resource
If an informed node receives a query, generate the appropriate response and forward the response to the originating node	Response is correctly forwarded	Protocol is energy efficient
	Response is incorrectly forwarded	Increased energy expenditure and time required to answer the query
	Response is not forwarded	Query fails; increased energy expenditure and time required to reissue the query and locate the resource
If resources/queries have finite lifetimes, remove the corresponding agent/query from a node's event table and/or transmission queue upon expiration	Resource/query correctly removed upon expiration	Protocol is energy efficient
	Resource/query is not removed upon expiration	A query may be answered using stale information; also, increased energy expenditure and latency due to the need to reissue the query

3.4 Workload

In the context of energy efficiency, the total workload imposed on the network is a function of the total amount of time each node in the network spends in the transmitting, receiving, sensing, computing, and sleep states. To ensure long network life, the amount of time a node is permitted to remain in a particular state is normally inversely proportional to the energy expended in that state. These states, from least to most energy intensive, are: sleeping, computing, sensing, receiving, and transmitting. Since transmission and reception require the greatest expenditure of energy, low network traffic levels are the norm in wireless sensor networks. Thus, even in dense networks, the probability of transmission collision is low when compared to other types of wireless networks.

The amount of energy expended in the data collection/sensing function affects the frequency at which the search algorithm must generate resource advertisements. However, the frequency and duration of data collection is mandated by the network's requirements and is not controlled by the search protocol; therefore, its effects are not considered when setting the workload of the search protocol. Additionally, the amount of energy expended by computation in support of the search protocol is insignificant relative to the energy expended by transmission and reception [TAH02]. Hence, this research defines a search protocol's workload by the number of transmissions required and, in the case of multiple receivers per transmission, the total number of designated receivers.

The majority of the search protocol's work is generated under three conditions: by a node's detection of a reportable event, by a node's generation of a request for information not available in its local cache, and by the process of forwarding a response to the requesting node. Therefore, five factors affect the total workload generated by a search algorithm in a wireless sensor network:

- The frequency of reportable events
- The total number of resource replicates created per reportable event

- The frequency of resource requests
- The total number of nodes polled before an informed node is located or the query expires, whichever comes first
- The number of hops required to forward the response from an informed node to the originating node

The frequency of a reportable event can be either deterministic (e.g., hourly temperature reports) or probabilistic (e.g., the detection of a particular radioactive isotope). However, to prevent congestion of the transmission medium and ensure long network lifetime, the total rate of traffic generation within the network must remain relatively low. For example, if each node in a WSN has an event detection rate of 0.001 events per second, then a 10000-node network will generate 10 reportable events per second. If each node informs 100 other nodes of the event, then as many as 1000 transmissions per second are required. Despite the fact that WSNs can support simultaneous non-colliding transmissions due to the limited transmission range of the nodes, this transmission requirement would likely exceed the network's available bandwidth; it is improbable a WSN with limited energy stores could support or sustain this workload for any significant length of time. In contrast, if each node informs only five other nodes of an event, the network need only support 50 transmissions per second. The latter scenario is more likely to be within the capabilities of the network.

A consequence of the previous scenario is that a query is likely to require fewer transmissions to locate an informed node in the former network than the latter. The question, then, becomes determining the appropriate number of informed nodes required to minimize the total workload (i.e., transmissions and receptions) imposed on the network by the search algorithm. Since the rate at which events are detected and reported by individual nodes is typically beyond the control of the designer once the network is deployed, the primary means to affect the workload imposed on the network is to manage the total number of resource replicates created by each event. Therefore, to ensure the total workload created by the search algorithm is within the capacity of the network, the rates of generation of events and requests in the large-population networks examined in this research are assumed to be relatively small, and the total number of nodes informed per event will comprise only a small percentage of the total nodes in the network. Furthermore, by ensuring the search algorithm parameters are optimized for energy efficiency, the total workload generated is minimized—an important goal of this research. In subsequent chapters, additional workload details on are provided for each phase of the research.

3.5 Performance Metrics

Two metrics will form the principal means for evaluating the performance of search protocols in this research. These metrics are:

1. Mean total network energy consumed to transmit/receive agents, queries, and responses in support of the search protocol.
2. Mean total proportion of queries that fail to locate an informed node.

Due to the energy-limited characteristics of the nodes and the difficulty associated with replenishing the energy reserves of large-population sensor networks, measuring the energy efficiency of a particular algorithm or protocol is of utmost concern. As discussed in Chapter 2, transmission and reception typically consume the largest portion of a node's energy reserves in today's wireless sensor devices [ROG06, TAH02]. Therefore, the total energy consumed by the network to transmit and receive packets in support of a particular protocol determines its energy efficiency. Also, if the nodes are assumed to communicate in a unicast manner, i.e., one designated receiver per transmission, the energy consumed can be measured by counting the total number of transmissions made, bits/packets sent, or bits/packets received per unit time in a manner similar to the works cited in Section 2.2.1.

In agreement with the majority of research in the field, this research evaluates the energy efficiency of a search protocol by measuring the total energy expended by the network to transmit and receive agents, queries, and responses. Two variants of this metric are employed. In the case of multiple receivers per transmission, the total energy consumed by the search protocol consists of (1) the energy consumed by the transmitter to transmit search-related packets and (2) the sum total energy consumed by the receivers to receive these packets. If there is only one designated receiver per transmission, an indicator of the total energy consumed by the protocol is obtained by counting the total number of transmissions received by each node. When required, the actual energy consumed by a unicast search protocol is obtained by multiplying the total number of transmissions by $(E_{xmt} + E_{rcv})$, where E_{xmt} is the mean energy expended by each node per transmission, and E_{rcv} is the mean energy expended by each node to receive a transmission.

Although energy efficiency is a key metric, it provides no information on the ability of the search protocol to meet the data requirements of the network's application(s). If a particular search protocol cannot answer a sufficient fraction of the total queries generated by the network, the network's application(s) is (are) likely to fail; the energy efficiency of the protocol is of little consequence. Therefore, it is important to determine the total proportion of queries generated by the network that fail to locate the desired information. Surprisingly, there is little attention given to this metric in the current literature, and none have attempted to determine the expected proportion of query failures analytically.

3.6 Parameters

Parameters affect the performance of the system and/or the system workload [Jai91]. Although search protocols support the network by providing the capability for nodes to locate information necessary to complete assigned tasks, the discussion of parameters in the following subsections is limited to those parameters directly affecting the performance of the search protocol (i.e., system parameters) and those that affect the search protocol's workload.

3.6.1 System parameters

System parameters affect the performance of the search protocol. These parameters are:

- The number of nodes in the network
- Physical dimensions of the network deployment area
- Maximum effective node transmission range
- The length of time a resource is made available for access by the network
- The length of time nodes are able to wait for a response to a query before application failure occurs

- The amount of energy required for nodes to transmit packets, receive packets, carry out computation, collect data, and sleep
- The time required for a node to successfully transmit a packet to a neighboring node once access to the transmission medium has been granted
- The amount of time and energy expended by the medium access control protocol to gain access to the transmission medium
- The time and energy expended by the network to provide node localization (for search protocols requiring this information)
- The time and energy expended by each node to perform computations in support of the search protocol
- The probability of transmission collisions
- Retransmissions required due to transmission/reception errors or collision
- Individual node failure rates
- Node mobility

3.6.2 Workload parameters

Workload parameters affect the search protocol's intensity of service requests. The workload parameters are:

- The rate of occurrence of reportable (i.e., agent-generating) events and/or the rate at which individual nodes offer specific services to the system
- The rate at which applications generate requests at each node (i.e., resource popularity)

- The proportion of nodes informed by each resource advertisement (set via a *time-to-live*, or TTL, counter)
- The rate of expiration of requests
- The rate of expiration of resources/resource availability
- The rate at which agents and/or queries are successfully forwarded from node to node

3.7 Factors

To obtain an accurate measure of the performance of a search protocol via modeling or simulation, it is advantageous to isolate the performance of the search protocol from any effects attributable to other aspects of WSN design (e.g., delays in transmission as a consequence of the choice of MAC protocol). As discussed in Section 2.3, the interdependence of the many facets of WSN design complicates this goal. Additionally, by including a large number of factors in the analytical model of a search protocol, the model has a greater probability of correctly modeling performance in real-world networks; however, analysis of such models may be difficult, computationally intensive, or even intractable. By limiting the number of factors, the resulting models are easier to analyze, but this approach carries the risk of removing the model further from reality to the point that it no longer provides useful insight. Regardless, this research takes the approach that a particular factor should not be excluded from an analytical model or simulation unless its inclusion (1) unnecessarily complicates subsequent analysis or results in an intractable model and (2) provides little additional insight into the performance of the search protocol. The factors and anticipated performance effects used in this research are summarized in Table 2. The applicable levels chosen for each factor are discussed in detail in later chapters of this dissertation.

Table 2: Selected factors and anticipated performance effects.

Factor	*Anticipated effect on performance*
Number of nodes in the network	Increasing the number of nodes in the network should increase the total energy expended by the search protocol as a consequence of the need to inform/query additional nodes
Physical dimensions of the deployment area	Increasing the dimensions of the network decreases node density and reduces the number of neighbors that can be polled by a single query transmission. Consequently, overall energy expenditure of a search protocol is expected to increase.
Transmission range	Increased transmission range requires greater transmission power but also increases each node's one-hop neighborhood (thereby improving network connectivity) and reduces the number of hops required to answer a query. In general, though, the reduction in the number of hops required is outweighed by the increased transmission power consumed.
Resource lifetime	Longer resource lifetimes result in decreased total energy expenditure because each resource need only be advertised to smaller subset of the network's nodes.
Query lifetime	Longer query lifetimes are expected to slightly increase the total energy expended by the network as a consequence of lower query expiration rates. However, a smaller proportion of queries will fail to locate an informed node.
Transmission energy	Increasing the energy required for transmission will increase the total energy consumed by the search protocol and will increase the node density that corresponds to the minimum total expected energy expenditure.
Reception energy	Increasing the energy required to receive a packet will increase the total energy consumed by the search protocol and will decrease the node density that corresponds to the minimum expected total energy expenditure.
Transmission time/rate	Increasing the time required for transmission will increase the proportion of query failures (when deadlines are imposed).
Rate of query generation (resource popularity)	Increasing the popularity of a particular resource will require a larger subset of the network to be informed but will reduce the total number of transmissions per query. Overall energy expenditure per query will be reduced as the cost of resource advertisements is amortized over a larger number of queries.
Rate of resource generation	Higher rates of resource generation will decrease the number of resource replicates required for each instance of the resource, i.e., each agent will need only inform a smaller number of nodes.
Time-to-live (TTL)	Sets the maximum number of nodes that may be informed by a resource advertisement. Higher TTL values require more energy to be expended for forwarding agents but also reduce the expected number of query transmissions required to locate an informed node.

Although the energy expenditure and latency associated with a network's MAC protocol can affect the performance of a search protocol, it is not explicitly included in Table 2. This is because modeling a search algorithm in the context of a specific MAC protocol unnecessarily limits the generality of the results. There are a large number of MAC protocols available to WSNs; the effort required to assess every existing protocol is prohibitive. Instead, the temporal and energy expenditure characteristics associated with a network's MAC protocol are modeled indirectly via two parameters: the total time expired per successful transmission (i.e., transmission time/rate), and the total energy expended to transmit and receive a packet. These factors can be easily modified to reflect the actual performance of a particular MAC protocol. Moreover, despite the assumptions of low traffic intensity and limited node transmission range, the possibility of transmission collision still exists if a collision-avoidance MAC protocol is used. However, any increases in energy expenditure and latency associated with transmission collisions can be incorporated into these factors as well. When necessary, detailed discussion of any limitations imposed by this approach to modeling the MAC protocol is provided in the applicable chapter.

Performance effects due to network services such as localization, synchronization, and neighbor discovery are not modeled for several reasons. First, these services are not generally offered by the network for the exclusive support of the search protocol. Other network functions, such as data collection, are also dependent on the proper operation of such services. Hence, it is difficult, if not impossible, to differentiate the proportion of energy expended by these activities in direct support the search protocol and that expended for other purposes. Second, due to node mobility and node addition, deletion, and failure, the amount of energy expended for these services may vary greatly between networks. Since the occurrence of these events is beyond the control of the search protocol, the performance effects attributable to these services are not considered. Instead, it is assumed the network provides the necessary supporting services to enable the search protocol to operate properly.

3.8 Evaluation Technique

At the present time, actual WSNs composed of hundreds of thousands of nodes are unavailable, and the costs associated with deploying smaller networks with hundreds or thousands of nodes for testing are prohibitively expensive. Consequently, analytical modeling and simulation are the only viable alternatives for evaluation available and, in fact, comprise the majority of the performance evaluation methods employed in the current body of WSN search protocol literature.

Unfortunately, reliance on analytical modeling and simulation for evaluating the performance of search protocols in large networks for which no previous performance data exists begs the question: How does one validate the results? Answering this question requires examination of the three key facets of model design: assumptions, input parameter values and distributions, and output values and conclusions [Jai91]. Since this research is composed of three phases, each of these facets of design is discussed in further detail in the relevant chapter. On the whole, however, this research takes the approach that an analytical model must minimize the number of assumptions made and/or justify each assumption, provide the capability to optimize the search protocol's parameters, and generate results that are intuitively correct (referred to as "expert's intuition" in [Jai91]). Additionally, the results obtained via simulation should be similar to those predicted by the analytical model. Nevertheless, some differences between the analytical and simulation results are expected because simulation models generally require fewer simplifying assumptions than analytical models. However, any performance differences between the two should be readily explicable.

3.9 Experimental Design

For brevity, specifics regarding the experimental design for each phase of research are described in the appropriate chapter.

3.10 Summary

This chapter described the research goals of this dissertation, identified the scope of the research, provided justification for specific assumptions, and offered a general outline of the tasks to be accomplished. Additionally, system services, performance metrics, parameters, and factors were identified. The choice of evaluation techniques—analytical models and simulation—was justified, and the means to validate the results described. The next chapter focuses on the first goal of this research: the development of an energy-efficient, scalable, small-footprint search protocol for large, dense wireless sensor networks.

4. A Trajectory-based Selective Broadcast Query Protocol

4.1 Introduction

This chapter presents an energy-efficient, scalable, small-footprint search protocol that facilitates *any*-type queries for data content and services in large population, high-density wireless sensor networks. This protocol, named Trajectory-based Selective Broadcast Query (TSBQ), works in conjunction with time division multiple access- or schedule-based MAC protocols to reduce per-query energy expenditure. The performance of TSBQ is compared to unicast- and local broadcast-based search algorithms, and a critical node density based on the energy expended by nodes to transmit and receive is determined. As will be demonstrated, the minimum energy expenditure is achieved by determining the optimal number of data/service replicates and the number of nodes designated to receive each query transmission. The numerical results obtained from the analytical model indicate TSBQ significantly reduces the total energy expenditure of a network as compared to unicast and local broadcast-based search protocols.

The work in this chapter makes several unique contributions. First, an analytical model for the expected total energy expended by TSBQ is provided. Using this analytical model, the means to minimize the expected total energy expended is demonstrated via simultaneous determination of the optimal number of agent replicas and the number of nodes that should be designated as receivers for each query transmission. Using this model, the performance variance of rumor routing-based search protocols is predicted, and a means to minimize this variance is proposed. Third, by means of a simulation model, the performance of TSBQ is evaluated and consequently, further refinements to the protocol are suggested. Fourth, the effects of network boundaries on search algorithm performance are elucidated, and these effects are incorporated into the mathematical model. Finally, the means to evaluate tradeoffs between important network parameters—including the number of agent replicas stored in the network, total network storage capacity, hardware power requirements, and node density—has received little attention in the open literature. Portions of this research close that gap by providing a means to evaluate the effects of these parameters on overall energy savings, effective total network storage capacity, query response variance, and query latency.

The remainder of this chapter is organized as follows. Section 4.2 provides a brief discussion of the aspects of the TSBQ protocol that make it unique compared to existing search protocols. In Section 4.3, a mathematical model for the expected total energy expenditure of the TSBQ protocol is developed and analyzed. The results of simulation experiments with large, high-density networks are presented in Section 4.4. Based on the results of these experiments, improvements to the protocol and mathematical model are proposed.

4.2 Uniqueness of TSBQ

The original rumor routing protocol [BE02] discussed in Section 2.1.3.1, as well as several of its variants [BTJ05, BCM05, CSC05, TV04], are most closely related to the TSBQ search protocol. With respect to this research, however, it has been noted that there are currently no analytical models of rumor routing-based search protocols that determine the optimum resource replication levels based on node hardware characteristics and resource popularity. Moreover, no protocols currently take advantage of the power of broadcast transmissions, nor do they incorporate a feedback-driven caching mechanism to improve latency and decrease the energy expended by subsequent queries.

Although TSBQ is inspired by traditional rumor routing, the following characteristics make it unique:

- TSBQ is the only WSN search protocol to minimize the total expected energy expenditure of the network by analytically determining the optimum number of resource replicates created by each agent. Additionally, TSBQ leverages the broadcast nature of wireless transmissions to query multiple nodes per transmission, thereby reducing total energy expenditure.
- TSBQ specifically accounts for resource popularity and the energy expended by nodes both to listen and to receive when determining the appropriate number of receivers and the number of nodes informed via agents. Additionally, TSBQ accounts not only for the energy expended to inform the network via an agent and locate the desired information via a query but also for the energy expended to return the response to the originating node. Achieving maximum energy savings requires optimizing each of these sources of energy expenditure simultaneously.
- Nodes need only maintain one-hop neighbor information to eliminate redundant node querying. Although a node may receive a *reissued* query more than once (see Section 4.4), this can be eliminated by permitting nodes to ignore a reissued query during the applicable transmission period.

- TSBQ reduces network congestion by limiting responsibility for transmission of the query to a single node, thus avoiding the inherent difficulties and inefficiencies associated with network flooding.
- TSBQ includes a feedback-driven caching mechanism to reduce search latency for popular data/services. This mechanism requires negligible additional energy expenditure by the network.

4.3 Protocol Description

It is well known that nodes can conserve energy resources by turning off transmitting and receiving hardware when not in use [LKR04, ROG06, VL03, YHE02]. Several MAC protocols such as S-MAC [YHE02], D-MAC [LKR04], T-MAC [VL03], and TRAMA [ROG06] achieve energy savings in this manner. TSBQ takes advantage of node hardware characteristics and the energy savings of TDMA-based MAC protocols to determine the appropriate advertising and query strategy for the network. Although all nodes must participate in the MAC's contention period to coordinate transmission and reception schedules, nodes not designated to transmit or receive during a given transmission period are permitted to enter a low-power sleep mode. The goal, then, is to minimize the total energy expended by simultaneously determining the appropriate number of receivers designated by the MAC during each transmission period and the optimum number of resource replicates.

4.3.1 TSBQ Overview

When discussing the means to propagate and locate information within a network, this dissertation adopts and expands much of the terminology of Braginsky and Estrin

[BE02]. *Agents* are packets transmitted by *witness nodes* to advertise the availability of specific services or data. *Informed nodes* have received an agent transmission and stored the agent's content in a local *event table*. A node seeking data or a particular service is the *origin query node* (OQN), and nodes that relay query packets on behalf of the OQN are *query nodes* (QN). OQNs and QNs transmit *queries*, packets that "roam" the network in search of specific services or data. *Receiving nodes* (RN) adjust their sleep cycles to accommodate the transmission schedules of neighboring OQN/QNs when designated by the OQN/QN to receive a query transmission. When a query is received by an informed node, the node generates a *response* that is returned to the OQN. The response may contain the specific data requested by the end-user or simply provide the location of the desired data or service.

Two basic principles motivate the development of TSBQ. First, it is necessary to strike a balance between the energy expended to inform the network of an event or service via an agent and the energy required to locate an informed node via a query. If too few nodes are informed, less energy is used to transmit agents and the network storage burden is decreased. However, a query will likely expend additional energy to locate an informed node thereby negating any potential energy savings. Conversely, if too many nodes are informed, the amount of energy expended for each query is reduced, but the energy required to propagate each agent is increased and a larger portion of the network's aggregate storage capacity is consumed. Second, when querying neighboring nodes, the number of nodes that receive each query transmission should be determined by the energy expended by these nodes to receive the query. If too few nodes receive the query, additional transmissions may be required to locate an informed node. By contrast, if too many nodes receive the query, an informed node may be located with lower latency, but the uninformed receiving nodes still pay a cost for receiving the query packet.

The TSBQ search protocol consists of the following steps:

1. A node witnesses an event and generates an agent to inform an additional $(\alpha N - 1)$ nodes, where N is the number of nodes in the network. To ensure the value $(\alpha N - 1)$ is integral, $\alpha \in \{1/N, 2/N, ..., (N-1)/N\}$.
2. An OQN generates a query and chooses a random direction (trajectory) for routing. Based on this trajectory, the OQN chooses the next *potential query node* (PQN) from among its one-hop neighbors using the Most Forward within Range (MFR) criterion (Figure 2) [SL01].

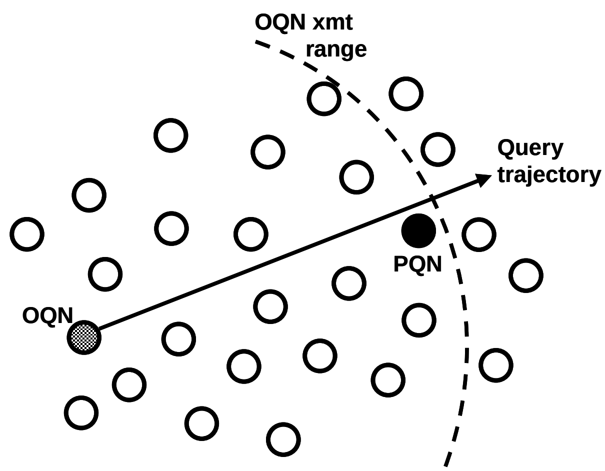


Figure 2. The OQN chooses the PQN using MFR.

3. The OQN/QN randomly selects $(\delta' - 1)$ RNs from among its neighbors that are closer to itself than the PQN (Figure 3), where δ' is a positive integer no

greater than the cardinality of the node's neighbor set, δ. (Determining the optimum value of δ' is discussed in Section 4.3.)

4. Transmission/reception coordination between the OQN/QN and RNs is achieved via a TDMA- or schedule-based MAC protocol during the contention period. The OQN/QN sets the transmission-reception schedule for its neighbors and designates the RNs. Nodes not designated as a QN, PQN, or RN enter sleep mode to conserve energy during the appropriate transmission period(s).
5. The OQN/QN broadcasts the query to the PQN and the designated RNs (a total of δ' receivers per query transmission).

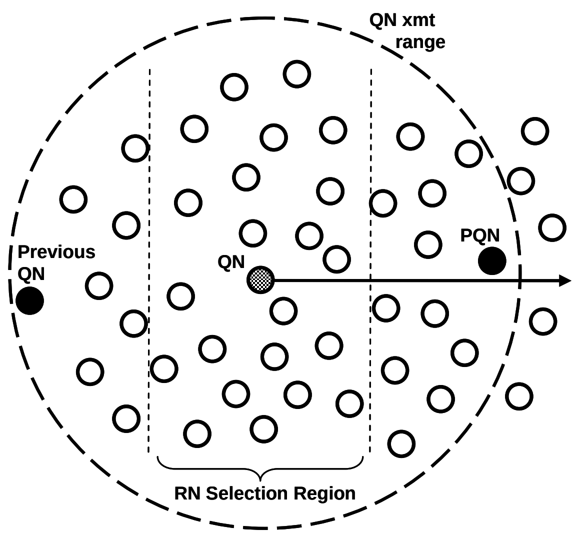


Figure 3. RN selection region (isotropic transmission model).

6. If no response is received from the PQN or RNs (i.e., the query fails to locate an informed node), then the PQN becomes the next QN. The new QN chooses a PQN using MFR along the designated trajectory. The process returns to Step 3 and repeats until the query is successful or terminated.

7. If at least one PQN or RN is informed, the node transmits the desired information to the QN. The response is then returned to the OQN via MFR routing along the trajectory defined by the positions of the QN and OQN. The query is terminated by the PQN once it overhears the response transmitted by the QN.
8. A feedback-driven caching mechanism may be incorporated to enable intermediate nodes along the route from the informed node to the OQN to add the information in the response to their own event tables. This mechanism is discussed in Section 4.4.

The partial network diagram in Figure 4 is a graphical depiction of the TSBQ protocol. The black arrow is the OQN's randomly-chosen query trajectory, the solid black circles are the PQN/QN sequence of nodes responsible for transmitting the query at each hop, and the gray circles designate the RNs randomly polled by a QN to determine if they have a corresponding agent. The dashed arrow is the trajectory of the desired agent, and an "X" within a node indicates it is informed. For example, nodes C4 and D3 in Figure 4 have received and stored a copy of the agent sought by the OQN. Each node has approximately $\delta = 18$ one-hop neighbors, and $\delta' = 8$. The means to analytically determine δ' is discussed in Section 4.3.3.

When a node needs a non-local resource yet has no knowledge of the resource's location, the node designates itself as the OQN and randomly picks a query trajectory. Based on this query trajectory, the OQN selects the PQN (node QN1 in Figure 4) and randomly chooses $(\delta' - 1) = 7$ neighbors (i.e., RNs) from among those nodes closer to itself than the PQN. After coordinating with its neighbors during the MAC contention

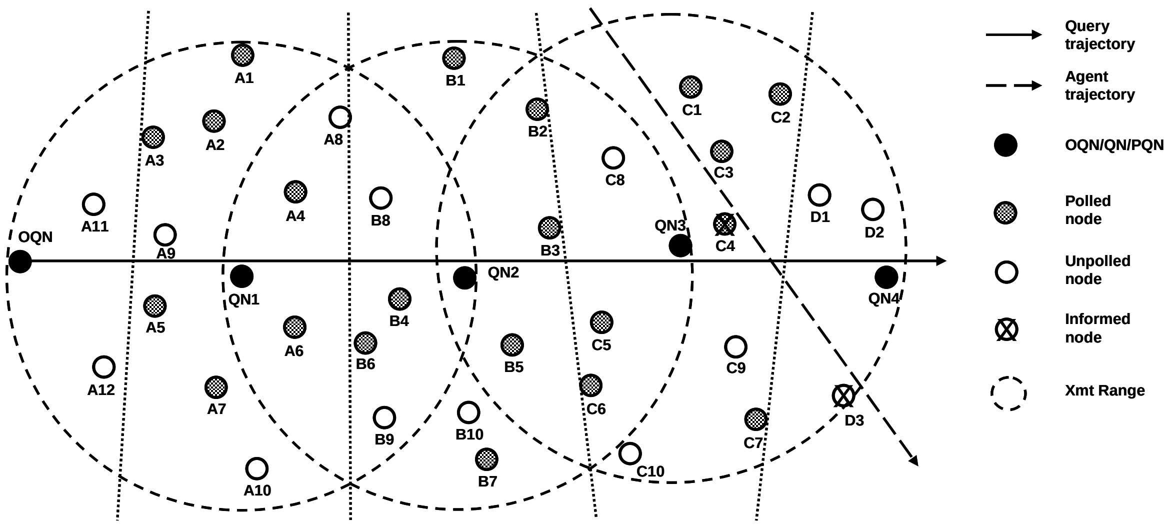


Figure 4: Graphical depiction of the TSBQ protocol.

period, the OQN transmits the query to the PQN and the RNs. The OQN's remaining neighbor nodes are permitted to sleep during this transmission period. If neither the PQN nor the seven RNs polled by the OQN can answer the query, the PQN will query a subset of its neighbors on behalf of the OQN. Although not shown in Figure 4, the OQN's query is unsuccessful; therefore, node QN1 must forward the query.

Based on the query trajectory chosen by the OQN, node QN1 identifies node QN2 as the PQN and randomly selects nodes A1 – A7 as RNs. Since neither QN2 nor A1 – A7 are informed, QN1's query fails, and QN2 assumes responsibility for the next query transmission. QN2 chooses a PQN (QN3) based on the specified query trajectory and selects seven RNs (B1 – B7). Since none of these nodes hold a copy of the desired agent, QN2's query also fails.

Once QN3 recognizes QN2's query has failed, it identifies the PQN (QN4) and chooses seven RNs (C1 – C7). Upon polling these nodes, node C4 responds with the desired information. QN3 uses this information to generate a response, determines the appropriate response trajectory, and returns the response to the OQN. When QN4 overhears the response transmitted by QN3, it terminates the query.

During each query transmission, it is possible that an informed node is a neighbor of the QN but is not located because the node was not chosen as a PQN or RN. This will delay a response to the OQN and require additional transmissions. Eliminating this possibility can only be achieved by transmitting the query to *all* neighboring nodes. However, Section 4.3.4 will show the expected total energy expended by the network to answer a query is minimized by choosing a subset of a node's neighbors as receivers when the node density exceeds a specific threshold.

4.3.2 Analytical Model of TSBQ Energy Expenditure

Three primary sources of network energy expenditure are required to generate a successful response to a query: agent transmission/reception, query transmission/ reception, and response transmission/reception. Achieving the minimum energy expenditure per successful query requires balancing these elements. Each source of energy expenditure is discussed individually in the following subsections.

4.3.2.1 Agent Transmission/Reception

Traditional rumor routing assumes each node within range of an agent transmission receives the agent and adds the event to its local event table. This results in a "thick line" of informed nodes in the network [BE02]. However, in high-density networks, this approach has two disadvantages. First, a large percentage of the total network storage capacity is consumed by these agents. Event tables of nodes located near active areas of the network will likely reach capacity quickly, requiring a replacement strategy for event table entries—an undesirable alternative. Second, unless the agent time-to-live (TTL) value is high, an agent may not be transmitted to distant regions of the network. This means large portions of the network have no informed nodes (i.e., a low spatial dispersion of informed nodes). As a consequence, networks using traditional rumor routing techniques may not locate an informed node without large energy expenditure.

To increase the spatial dispersion of informed nodes while simultaneously minimizing the number of transmissions, it is proposed that agents be forwarded along straight-line trajectories in a manner similar to [BCM05, NN03, TV04]. Additionally, to minimize local storage requirements, each agent transmission is unicast (i.e., intended for exactly one receiving node). Coordination between transmitting and receiving nodes is achieved via a TDMA- or schedule-based MAC protocol, such as T-MAC, during the MAC protocol's contention period. During the transmission period, all nodes within range of the agent transmission not designated as receivers deactivate their receiving hardware to conserve energy. The intended receiving node is chosen using MFR to eliminate routing loops [SL01]. In the event a node cannot forward an agent along the desired trajectory (e.g., due to encountering a network boundary), the node randomly chooses a new forwarding trajectory for the agent. Alternatively, if the agent cannot be forwarded due to a void or obstacle within the network, a face routing scheme such as Greedy Perimeter Stateless Routing [KK00] can be used to circumvent this region until the desired trajectory can be resumed. However, in the design space of large-scale, high-density networks using MFR, the probability of encountering a void is small [XK06]. Therefore, this occurrence is not included in the development of the mathematical model. Each agent is forwarded to exactly $(\alpha N - 1)$ unique nodes, thus ensuring there are αN informed nodes.

Once a node receives an agent, the node makes an entry in its event table that includes the type of service/data advertised, the location of the witness node, and a copy of the data (if available). Although any node that overhears an agent transmission may add the agent to its event table, this research advocates the unicast transmission of agents between nodes and the use of MFR to select receivers as a means to promote the maximum physical distance between identical event table entries. This reduces the probability that large numbers of informed nodes are found only within limited portions of the network.

If A denotes the total energy used to propagate each agent, then for large networks such that $\alpha << 1$, the expected total energy used to propagate each agent is

$$E[A] = (E_{xmt} + E_{rcv}) \cdot (\alpha N - 1), \tag{4.1}$$

where E_{xmt} is the energy expended by a node to transmit a packet, and E_{rcv} is the energy expended to receive a packet.

4.3.2.2 Query Transmission/Reception

When a node needs access to services or data but has no corresponding entry in its event table, the node generates a query. Because nodes may selectively activate and deactivate their receiving hardware, the node's query transmission may be received by one, some, or all of its one-hop neighbors simultaneously. Assuming informed nodes are uniformly distributed throughout the network and disregarding the effect of network boundaries (these assumptions will be revisited in Section 4.4), the number of informed nodes that are also neighbors of each QN is a binomial random variable.

Let Y be the number of informed nodes within one-hop distance of the QN. If a QN has δ neighbors and a corresponding query is transmitted to δ' of these neighbors, $0 < \delta' \le \delta$, the probability of failing to find an informed node is

$$\Pr\{Y=0\} = \binom{\delta'}{0}\alpha^0\left(1-\frac{\alpha N}{N-1}\right)^{\delta'} = \left(1-\frac{\alpha N}{N-1}\right)^{\delta'}, \tag{4.2}$$

and the probability of finding at least one informed node is

$$\Pr\{Y>0\} = 1-\left(1-\frac{\alpha N}{N-1}\right)^{\delta'}. \tag{4.3}$$

It is assumed a node does not generate a query for a particular service or data if it is already informed. As a consequence, the probability that an uninformed node's neighbor possesses the data of interest is slightly greater than α.

In TSBQ, queries are forwarded along straight-line trajectories in a manner similar to that used for agents. However, in contrast to agent transmissions, queries are broadcast to a subset of each node's neighbors. Nodes that have not been chosen to receive a particular query transmission turn off their receivers to conserve energy. The use of straight-line routing trajectories increases the probability that a subset of the QN's neighbors have not yet received the current query compared to random walk methods. Therefore, the probability of finding an informed node increases with each hop of the query along its assigned trajectory. Let Z_j be a Bernoulli random variable denoting success or failure of the jth query hop (transmission) such that $Z_j = 0$ when the jth query hop fails to locate an informed node and $Z_j = 1$ otherwise. If a query is broadcast to a unique set of δ' receivers at each hop in its path, the probability that the *j*th query transmission fails to locate an informed node is

$$\Pr\left\{Z_j = 0\right\} = \left(1 - \frac{\alpha N}{N-1-(j-1)\delta'}\right)^{\delta'}, \quad j \geq 1. \tag{4.4}$$

If an informed node is found on the *j*th hop, then an informed node was not located on the previous $(j-1)$ hops because a query is not propagated further once an informed node is found. Recall that TSBQ is designed for *any*-type searches; therefore, the search is concluded when at least one copy of the desired information is located. Consequently, the probability of locating an informed node on the *j*th hop is

$$\Pr\left\{Z_1 = Z_2 = \cdots = Z_{j-1} = 0, Z_j = 1\right\} = \begin{cases} 1-\left(1-\dfrac{\alpha N}{N-1}\right)^{\delta'} & j = 1 \\ \left[1-\left(1-\dfrac{\alpha N}{N-1-(j-1)\delta'}\right)^{\delta'}\right] \cdot \displaystyle\prod_{i=1}^{j-1}\left(1-\dfrac{\alpha N}{N-1-(i-1)\delta'}\right)^{\delta'} & j \geq 2 \end{cases}. \tag{4.5}$$

Clearly, sensor networks are comprised of a finite number of nodes. Assuming a query can be propagated without encountering a network boundary, the maximum number of query transmissions, *k*, that can be made to unique neighboring nodes before locating *at least one* informed node is

$$k := \left\lfloor \frac{N(1-\alpha)-1}{\delta'} \right\rfloor + 1, \quad \alpha \in \left\{1/N, 2/N, \ldots, (N-1)/N\right\}. \tag{4.6}$$

Equation (4.6) assumes that at least one node in the network has not received a copy of the agent; otherwise, there would be no need for a node to generate a query. Let $X_{\alpha,\delta'}$ denote the random number of transmissions required to find an informed node for fixed values of α and δ'. Then the probability of needing *j* query transmissions is

$$\Pr\left\{X_{\alpha,\delta'}=j\right\}=\begin{cases}1-\left(1-\frac{\alpha N}{N-1}\right)^{\delta'} & j=1\\ \left(1-\left(\max\left\{1-\frac{\alpha N}{N-1-(j-1)\delta'},0\right\}\right)^{\delta'}\right)\cdot\prod_{i=1}^{j-1}\left(1-\frac{\alpha N}{N-1-(i-1)\delta'}\right)^{\delta'} & 2\le j\le k\end{cases}, \tag{4.7}$$

and the expected value of $X_{\alpha,\delta'}$ is

$$E\left[X_{\alpha,\delta'}\right]=\sum_{j=1}^{k} j\cdot\Pr\left\{X_{\alpha,\delta'}=j\right\}. \tag{4.8}$$

Let Q be the energy expended by the network to locate an informed node. The use of straight-line trajectories for forwarding queries assuming no redundant polling of nodes means the expected energy to forward a query can be derived from (4.7) as

$$E\left[Q\right]=n\cdot\left(E_{xmt}+\delta'\cdot E_{rcv}\right)\cdot E\left[X_{\alpha,\delta'}\right], \tag{4.9}$$

where n is the total number of unique queries generated by n OQNs to locate a particular agent. Note that the number of informed nodes, αN, is assumed to be constant for all n queries. Although the number of informed nodes should increase as queries are answered, no temporal assumptions regarding the generation of queries or responses are made. Hence, (4.9) is an upper bound on the expected energy expended by the network to locate an informed node. Additionally, the value of n may be set prior to deployment based on analysis of the network's application(s), or it may be updated dynamically if, for example, one or more nodes recognize the number of unique requests for a particular resource exceeds a specified threshold. Alternatively, a feedback-driven caching mechanism can be used (cf., Section 4.4.3).

4.3.2.3 Response Transmission/Reception

Once the desired information is located, the response is returned to the OQN. Although it is assumed intermediate nodes in the response path are chosen using MFR along the straight-line trajectory defined by the current QN and OQN, there are several energy-efficient routing protocols that could perform this function. Most notably, Span [CJB+02] and GAF [XHE01) provide point-to-point routing services and are specifically designed to reduce energy expenditure by maximizing the number of nodes in the sleep state.

Let R be the energy used by the network to return a response to the OQN. Assuming the query does not encounter a network boundary prior to locating an informed node, the expected number of transmissions to return the response is identical to the expected number of query transmissions required to locate the informed node. Then the expected energy to return n responses to n OQNs is

$$E\left[R\right]=n\cdot\left(E_{xmt}+E_{rcv}\right)\cdot E\left[X_{\alpha,\delta'}\right]. \tag{4.10}$$

4.3.2.4 Expected Energy Requirement

The total energy T required to propagate an agent, its associated query(ies), and response(s) is the sum of (4.1), (4.9), and (4.10). An additional transmission and reception must be added for each query since an informed node, once located, must advise the current QN the desired information has been found. Therefore, the expected total energy expended by the network to generate n unique responses is

$$E\left[T\right]=\left(\alpha N-1+n\right)\left(E_{xmt}+E_{rcv}\right)+\left(2nE_{xmt}+n\left(\delta'+1\right)E_{rcv}\right)\cdot E\left[X_{\alpha,\delta'}\right]. \tag{4.11}$$

4.3.3 Minimizing Expected Total Energy Expended

The main objective of TSBQ is to minimize the expected total energy expended by the network to generate n successful responses to n queries for the desired data/service. Therefore, whenever E_{rcv}, E_{xmt}, N, and n are known, the objective is to select the optimal pair (α, δ') that minimizes (4.11).

The problem and its solution procedure are now formalized. To emphasize the explicit dependence of (4.11) on the decision variables α and δ', let $f(\alpha, \delta') \equiv E[T]$ denote the expected total energy expended by the network. The mathematical programming formulation is as follows:

$$\begin{aligned} &\min\ f(\alpha, \delta') \\ &\text{s.t.}\ \ \alpha \in \{1/N, 2/N, ..., (N-1)/N\} \\ &\qquad \delta' \in \{1, 2, ..., \delta\}. \end{aligned} \tag{4.12}$$

For a finite network, $f(\alpha, \delta')$ is a discrete function on a feasible region with $(N-1)\cdot\delta$ possible solutions. Therefore, the mathematical program is a straightforward discrete optimization problem in which the minimum energy expenditure may be obtained by enumerating all possible combinations of (α, δ'), and then choosing the (α, δ') pair that results in the least total energy expended. The pair of α and δ' values that result in the minimum expected energy expenditure is (α^*, δ'^*). A partial graph of the objective function for a 5000-node network is shown in Figure 5 where the expected total energy expended is normalized by the energy expended for node transmission and it is also assumed that $0 < E_{rcv} \leq E_{xmt}$. The E_{rcv}/E_{xmt} ratio is defined by the hardware characteristics of the nodes and sizes of the transmitted packets. It can also include the energy expended by the MAC layer for transmissions and retransmissions.

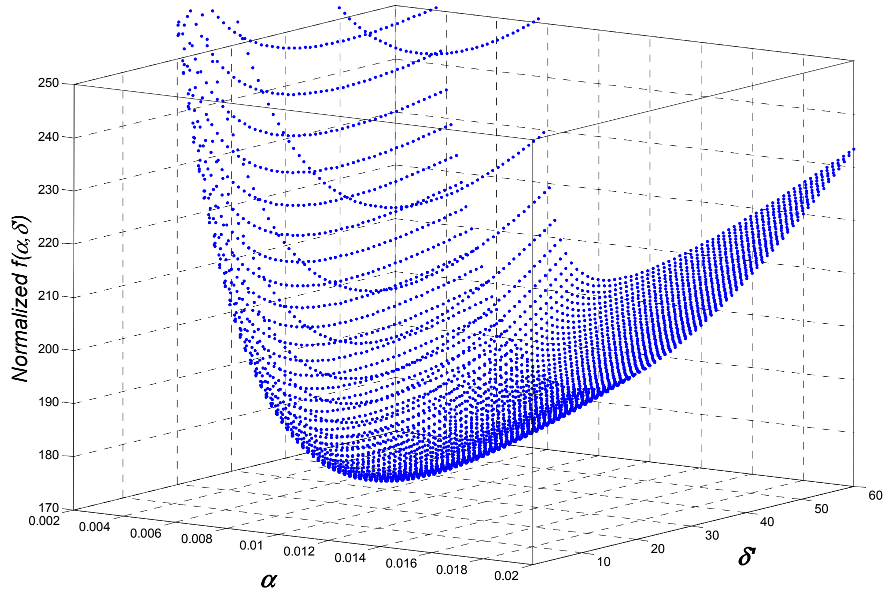


Figure 5. Plot of $f(\alpha,\delta')$, $N = 5000$, $n = 1$, $E_{rcv}/E_{xmt} = 0.7$.

The effect of increased network size and various E_{rcv}/E_{xmt} ratios on the optimal (α,δ') pair is now examined. The results of this analysis for a wide range of network sizes are shown in Figures 6 and 7 for the single-query case (i.e., $n = 1$), and the

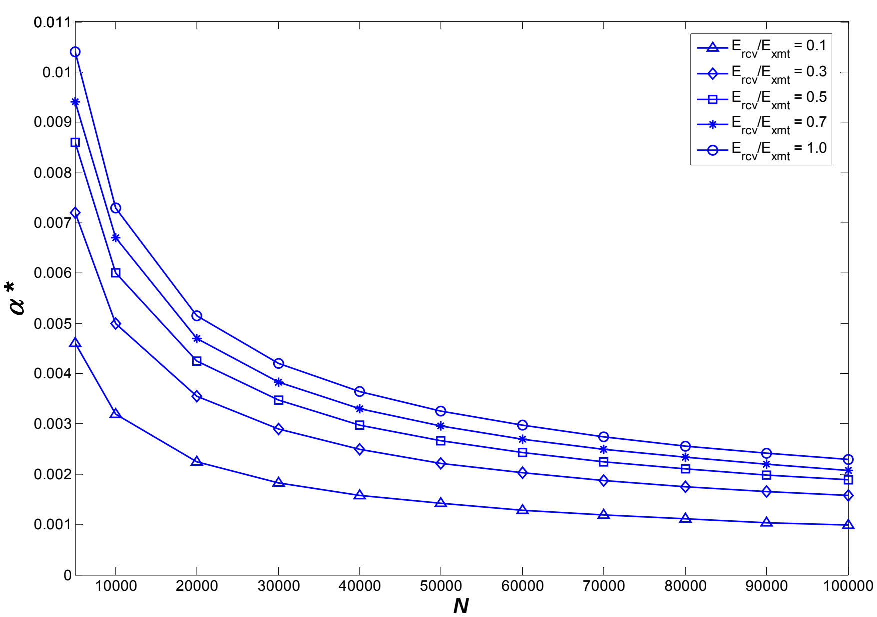


Figure 6. Effect of N and E_{rcv}/E_{xmt} on α^*, $n = 1$.

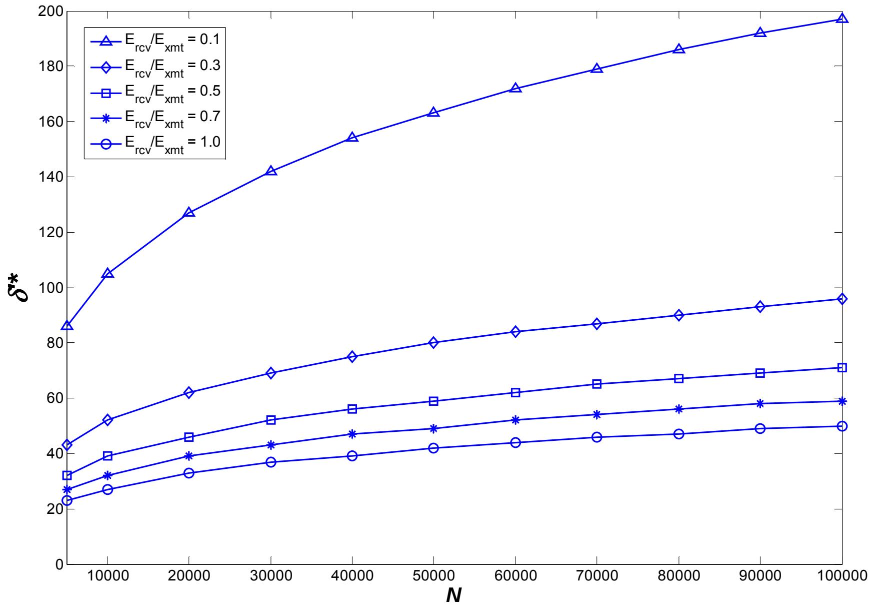


Figure 7. Effect of N and E_{rcv}/E_{xmt} on δ'^*, $n = 1$.

minimum expected total energy expended is shown in Figure 8. For example, a 50000-node network in which $E_{rcv}/E_{xmt} = 0.5$ has $(\alpha^*, \delta'^*) = (0.00266, 59)$, and expected total energy expended (normalized) is 419.6.

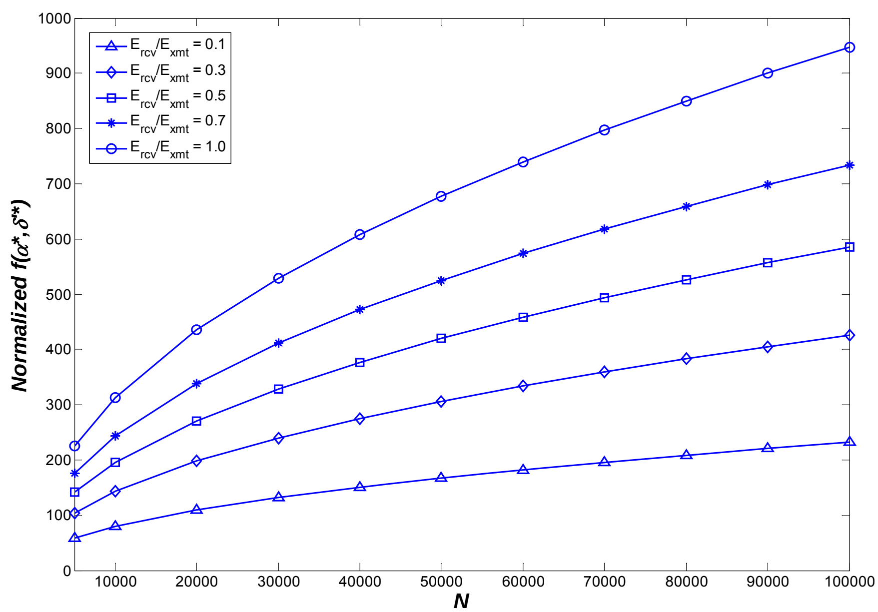


Figure 8. Expected minimum energy expended using (α^*, δ'^*), $n = 1$.

4.3.4 Approximating the Optimal Solution

Although (α^*, δ'^*) can be obtained for a network of fixed size, density, and E_{rcv}/E_{xmt} ratio via explicit enumerations, this method imposes a high computational requirement when N is very large. In the worst case, the optimization program requires $O(N)$ floating-point additions, $O(N^2)$ floating-point multiplications, and $O(N^2)$ floating-point exponential operations. For extremely large, dense, networks, it may not be feasible to carry out this analysis. Additionally, the parameters that characterize a newly deployed network will almost certainly change during the network's useful lifetime, requiring the optimal solution to be periodically updated. Thus, it is advantageous to express α^* and δ'^* as functions of N and E_{rcv}/E_{xmt}.

Regression analysis of the curves in Figures 6 and 7 reveals that the power model provides an excellent fit to the numerical results, yielding correlation coefficients greater than 0.999. The generalized power model is

$$A = B \cdot C(x)^p, \tag{4.13}$$

where A is the dependent variable, $C(x)$ is the independent variable, and B and p are constants. The following equations determine α^* and δ'^* as a function of the network size N

$$\begin{aligned} \alpha^* &= b_1 \cdot N^{p_1} \\ \delta'^* &= b_2 \cdot N^{p_2}, \end{aligned} \tag{4.14}$$

where b_1, b_2, p_1, p_2 are constants for a fixed E_{rcv}/E_{xmt} ratio.

The regression analysis reveals several key observations. First, the value of α resulting in the smallest total energy expenditure for a fixed E_{rcv}/E_{xmt} ratio is inversely proportional to the square root of N (i.e., $p_1 \approx -0.5$), and b_1 increases as the E_{rcv}/E_{xmt}

ratio increases. Hence, as network size increases, the minimum expected energy expenditure is obtained by using a smaller percentage of informed nodes. This property has the added benefit of reducing the percentage of total network storage capacity required by each unique agent, decreasing the probability that nodes will need to employ an event table entry replacement protocol. Second, the value of δ'^* for a fixed E_{rcv}/E_{xmt} ratio is approximately proportional to the fourth root of N (i.e., $p_2 \approx 0.265$), indicating that δ'^* increases at a much slower rate than the size of the network. As the E_{rcv}/E_{xmt} ratio increases, b_2 decreases, thus reflecting the increased cost of receiving a transmission. The value of δ'^* also defines the threshold one-hop neighbor density required to achieve the most energy-efficient search performance. As the average size of a node's neighborhood increases beyond the values indicated in Figure 7, TSBQ is more efficient than local broadcast (i.e., transmitting the query to all of a node's one-hop neighbors). However, when δ is less than $\delta'^*/(1-c_1)$, where c_1 is the average proportion of shared neighbors between each QN and PQN, the query should be broadcast to a node's closest neighbors to reduce total energy expenditure. That is, local flooding is simply a special case of TSBQ in which the computed value of δ'^* is greater than $\delta(1-c_1)$.

If δ' is decreased below the values in Figure 7, the expected total energy expenditure increases due to the larger number of query transmissions required to locate an informed node. The unicast query model, in which each query transmission is intended for a single receiver, defines the largest possible reduction in δ', i.e., $\delta' = 1$. The expected total energy expenditure for the unicast rumor routing model, similar to that used in SLR [CSC05], can be computed using (4.11) by substituting $\delta' = 1$. However, analysis of the unicast model indicates much larger values of α are required to achieve the minimum energy expenditure, and the minimum energy expenditure of the unicast model exceeds that of TSBQ. For example, in a 20000-node network with an E_{rcv}/E_{xmt} ratio of 0.7 and $n = 1$, the minimum $E[T]$ of TSBQ consumes 50.2% less energy than the unicast query strategy (338.7 versus 680.0 normalized energy units). Additionally, TSBQ requires only 94 informed nodes per agent to achieve minimum $E[T]$ versus 199 for the unicast protocol, a 52.8% reduction in total network storage capacity consumed per agent. For the 20000-node network, Figure 9 shows the minimum total energy expended by TSBQ ranges from 45.5% to 75.0% less than trajectory-based unicast search protocols, such as SLR.

Additional analysis of the model reveals the value of α^* increases by a factor of approximately 3.4 for each order of magnitude increase in n (Figure 10), and δ'^* decreases by a factor of approximately 2.0 for each order of magnitude increase in n

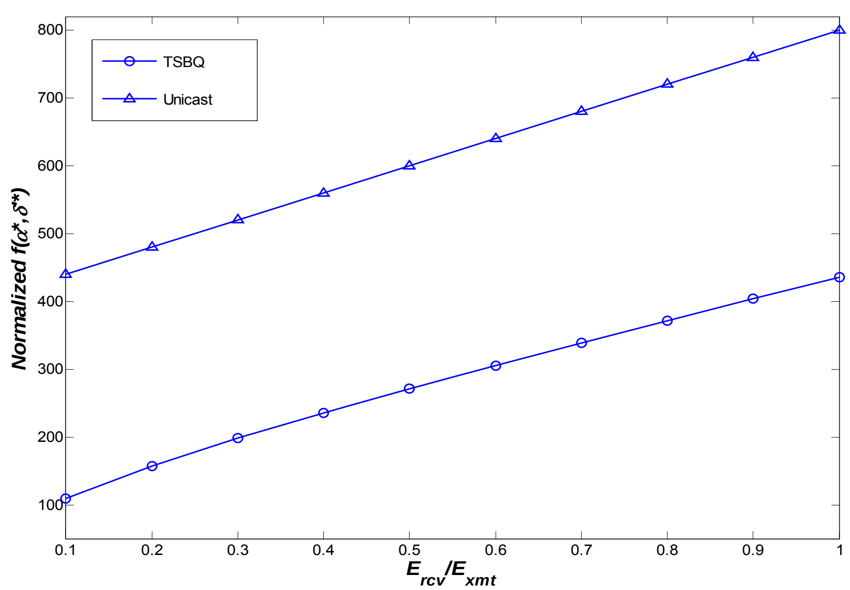


Figure 9. Minimum $E[T]$ of TSBQ versus unicast search, $N = 20000$, $n = 1$.

(Figure 11). This result is consistent with intuition: minimum $E[T]$ is achieved by advertising popular data/services to a larger portion of the network, thus permitting the energy costs related to advertising to be amortized over a larger number of queries.

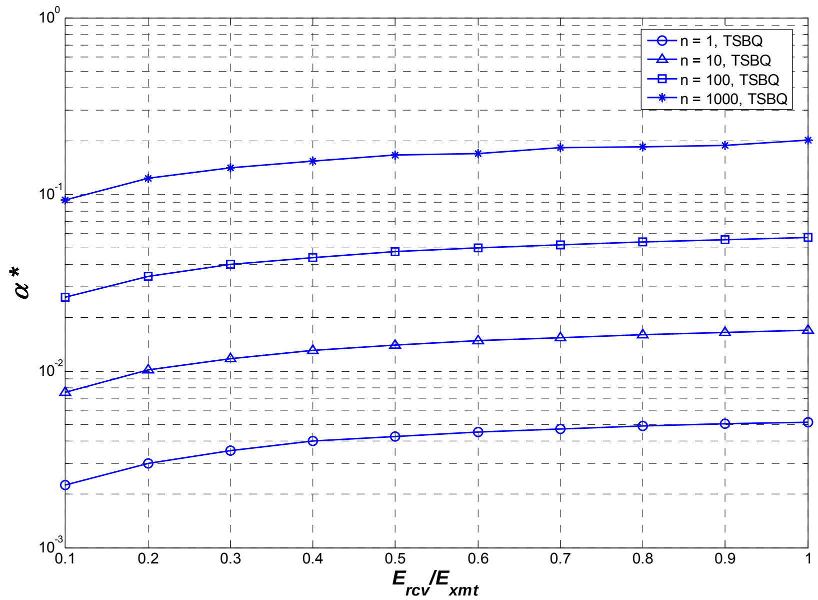


Figure 10. Effect of n on α^*, TSBQ protocol, $N = 20000$.

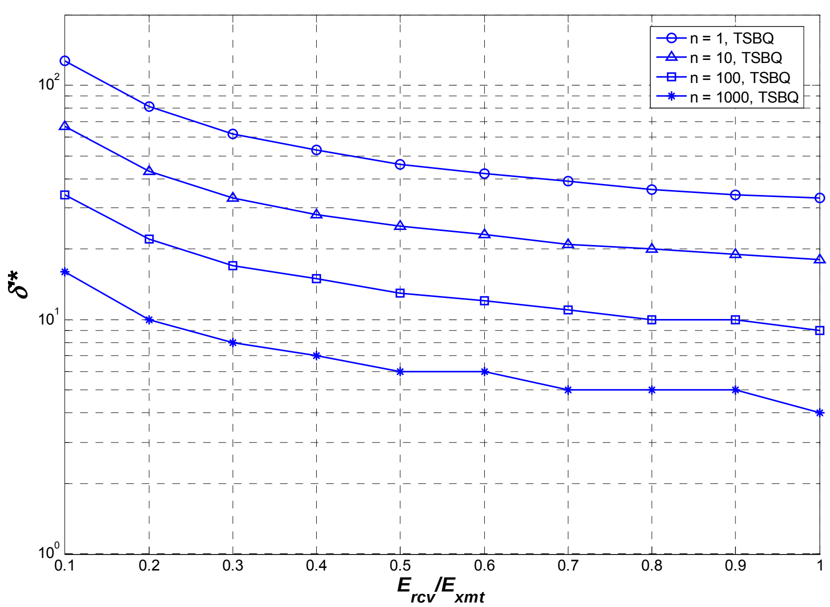


Figure 11. Effect of n on δ'^*, TSBQ protocol, $N = 20000$.

Additionally, when an item is heavily advertised, it is expected that the information will be located using fewer transmissions. Accordingly, δ' should be decreased to achieve the minimum total energy expenditure when an item is popular and heavily advertised, while δ' should be increased to locate less popular (and, hence, lightly advertised) items.

In contrast to TSBQ, unicast search algorithms require a higher proportion of informed nodes—regardless of the E_{rcv}/E_{xmt} ratio—to achieve minimum $E[T]$. As shown in Figure 12, the value of α^* for the unicast search protocol is unaffected by the E_{rcv}/E_{xmt} ratio, and this value always exceeds the corresponding α^* value for TSBQ since unicast protocols cannot take advantage of efficiencies gained by querying multiple nodes per transmission.

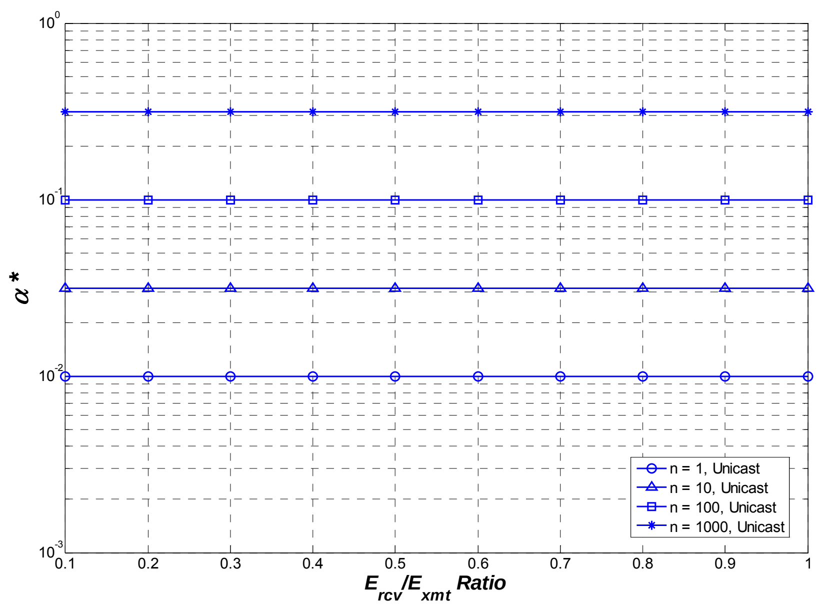


Figure 12. Effect of n on α^*, unicast search, $N = 20000$.

4.4 Simulation Results

Section 4.3.2 demonstrates how the TSBQ mathematical model can be usd to minimize the expected total energy expended to locate services and data within a WSN.

However, as noted in Section 4.3.2.2, the analytical model makes two simplifying assumptions. First, it assumes informed nodes are spatially uniformly distributed throughout the network. Second, the analytical model does not explicitly account for the probability of a query encountering a network boundary prior to locating an informed node. To examine the significance of these assumptions on the analytical model, the predicted performance of TSBQ is compared to the results of simulation.

Section 4.4.1 explains the construction of the network simulator. Section 4.4.2 examines the impact of network boundaries on the predictive value of the mathematical model, and Section 4.4.3 assesses the effects of trajectory-based forwarding—and the resulting non-uniform distribution of informed nodes—on performance. To improve performance, a simple feedback mechanism is proposed that imposes negligible additional energy cost. Section 4.4.4 evaluates the predicted and observed variance of energy expenditure per query. Finally, based on the simulation results, Section 4.4.5 proposes an improved mathematical model that incorporates network boundaries.

4.4.1 Simulation Construction

To accommodate the large, dense networks of nodes needed to evaluate the performance of the TSBQ protocol, a network simulator was implemented in MATLAB 7.0.0.19920 (R14). Since the analytical model assumes a reliable channel, no collisions, and retransmissions managed by the MAC layer (although these effects are indirectly included in the analytical model via the E_{xmt} and E_{rcv} parameters), a MATLAB-based simulation was well-suited for these purposes. Thus, it is possible to obtain in a reasonable time 1000 replicates per set of parameters—and ensure the stability of the simulation on a standard desktop PC.

The simulator generates networks of N randomly-placed nodes within the confines of a user-defined square deployment region. To simplify the process of determining the set of neighbors of each node, a circular (isotropic) radio propagation model was assumed, and the maximum transmission range that results in the minimum acceptable E_b/N_o for each node was specified. Although this transmission model is somewhat unrealistic for indoor environments, it has been found to be accurate for modeling outdoor WSNs [HBE+01]. Regardless, TSBQ does not require an isotropic transmission range for proper operation.

The simulation follows the steps of the TSBQ protocol outlined in Section 4.3. First, randomly-selected witness nodes forward an agent to $(\alpha N-1)$ unique nodes. Once the agents have informed the network, randomly-selected uninformed nodes generate queries. Prior to each query transmission, the transmitting node selects a PQN and also randomly chooses δ' of its closest one-hop neighbors as receiving nodes from among those nodes closer to the current QN than either the PQN or the previous QN. Although the node transmission model results in a well-defined region for choosing RNs (Figure 3), irregularly-shaped one-hop neighborhoods can be accommodated by permitting designated RNs to turn off their receivers if they determine they have already received a copy of a particular query. Once an informed node is found, the response is returned to the OQN. The mean total energy expended to inform the network, answer each query, and return the response is reported at the completion of 1000 independent trials for each (α,δ') pair. Simulations consisted of testing 5000-, 10000-, and 20000-node networks using the parameters summarized in Table 3.

Table 3. Simulation model parameters.

Network Size (N)	*Deployment Area*	*Effective Node Transmission Range*	*Average One-hop Neighborhood Size (δ)*
5000 nodes	30000 m^2	11 m	63
10000 nodes	59395 m^2	11 m	64
20000 nodes	97470 m^2	11 m	78

The average run-time for each simulation varies based on several user-defined parameters, including the number of nodes in the network and the number of replications of each experiment. However, using a 3.2 GHz Pentium IV computer with 1 GB of RAM and 1000 replicates per data point, the results presented in the next subsection required approximately 6 hours for the 5000-node network, 17 hours for the 10000-node network, and 56 hours for the 20000-node network.

4.4.2 Effect of Network Boundaries on Performance

The mathematical model of the expected energy requirement assumes a uniform distribution of informed nodes. Therefore, to study the effect of network boundaries on the performance of the protocol, the simulation was permitted to randomly choose αN informed nodes, thus permitting an assessment of the performance of TSBQ free of the effects of the agent routing method. The impact of trajectory routing on system performance is evaluated in Section 4.4.3.

The results of these simulations for 5000-, 10000-, and 20000-node networks are shown in Figures 13, 14, and 15, respectively. Each data point represents the average performance of 1000 independent simulation runs. With the exception of the smallest values of α (e.g., $\alpha < 0.004$ for the 5000-node case), the value of $E[T]$ predicted by (4.11) was within the 95% confidence interval of the simulation results. The observed results at lower values of α differ from the mathematical model due to a large number of queries dropped by the network at a boundary prior to discovering an informed node. When this event occurred in the simulations, the OQN was forced to reissue the query along another randomly-chosen trajectory after an appropriate timeout period. Since no limits were placed on the OQN's choice of trajectories for reissued queries in the

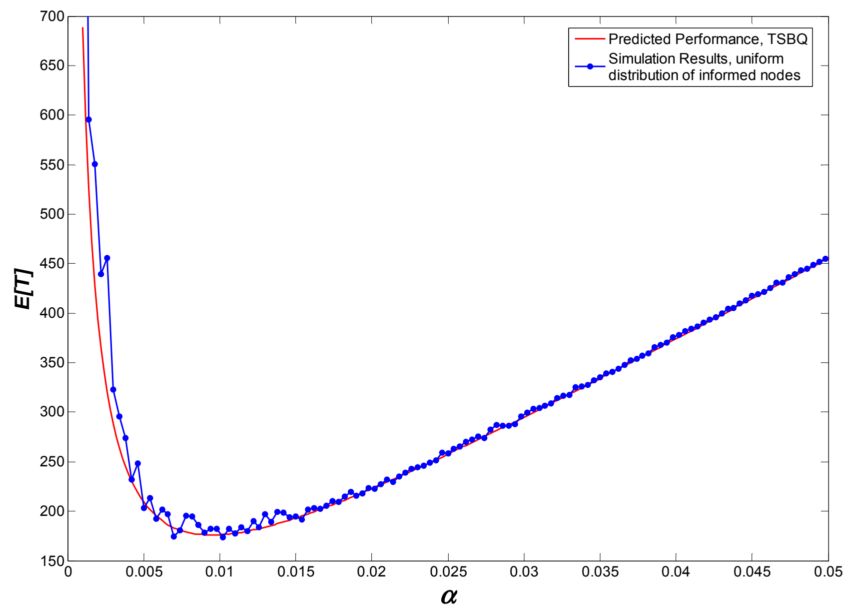


Figure 13. TSBQ performance, 5000-node network, $\delta' = 27$, $E_{rcv}/E_{xmt} = 0.7$.

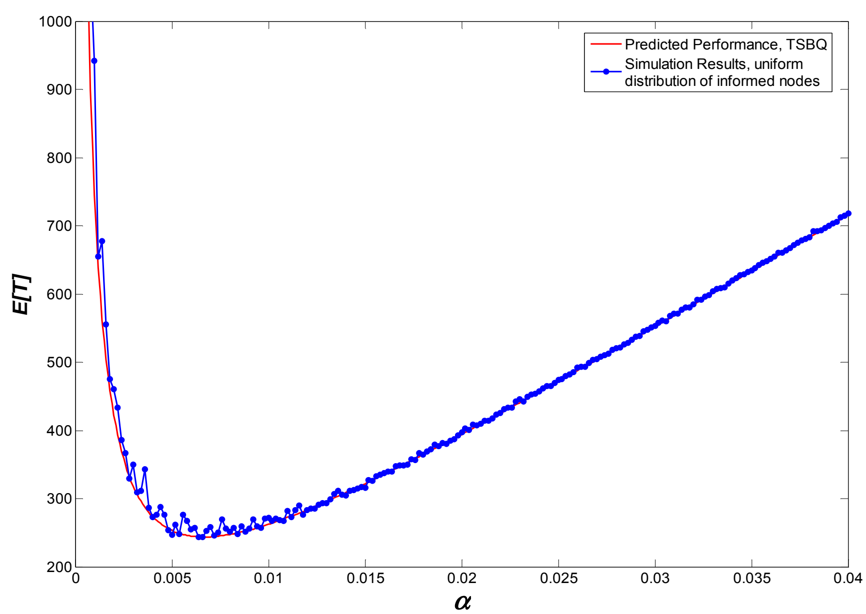


Figure 14. TSBQ performance, 10000-node network, $\delta' = 32$, $E_{rcv}/E_{xmt} = 0.7$.

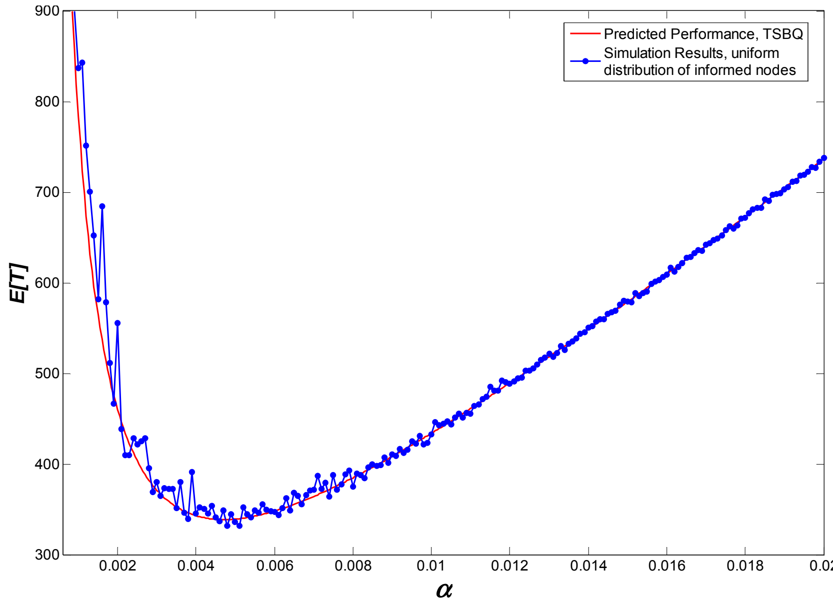


Figure 15. TSBQ performance, 20000-node network, $\delta' = 39$, $E_{rcv}/E_{xmt} = 0.7$.

simulation model, a node may receive the same query more than once if subsequent trajectories are similar to the original. As TSBQ is designed to prevent nodes from receiving transmissions of the *same* query on subsequent hops, it does not attempt to prevent nodes from being queried more than once by *reissued* queries. However, further energy savings can be obtained if nodes turn off their receivers once they determine a given query has already been received.

Based on these results, it is concluded that the mathematical model is useful for predicting the performance of the network if the actual proportion of informed nodes is not significantly smaller than α^*. However, the predictive capability of the model can be improved at small values of α by extending (4.11) to include parameters associated with the network deployment area and the transmission range of the nodes. Section 4.4.5 explains this extended mathematical model.

4.4.3 Effect of Trajectory-based Forwarding of Agents

Although the mathematical model assumes a spatially uniform distribution of informed nodes, such a distribution of informed nodes is difficult to achieve in real-world networks due to the limited transmission range of nodes. A uniform distribution of informed nodes might be attained by artificially partitioning the network into equal-size zones such as those used in Zonal Rumor Routing [BTJ05] or by guaranteeing at least k-hop distance between identical event table entries using a method such as k-DID [BCM05], but such schemes require additional energy expenditure and increase complexity. Also, algorithms such as k-DID have been found to scale poorly in dense networks [BCM05]. Instead, it is proposed to route agents along randomly-chosen straight-line trajectories and use MFR to choose intermediate receivers to achieve maximum initial spatial dispersion of informed nodes in the fewest possible transmissions. As a consequence, it is expected that mean per-query energy expenditure will differ from that predicted by the mathematical model, especially at lower values of α, due to a spatially non-uniform distribution of informed nodes and queries encountering a network boundary prior to locating an informed node.

To examine the effects of straight-line forwarding of agents on overall energy expenditure, additional simulation experiments were conducted using the parameters in Table 3. The results of these simulations are shown in Figures 16, 17, and 18. Each data point represents the average performance observed over 1000 independent simulation runs.

As expected, informing nodes via trajectory-based forwarding results in differences between the predicted and observed mean per-query energy expenditures;

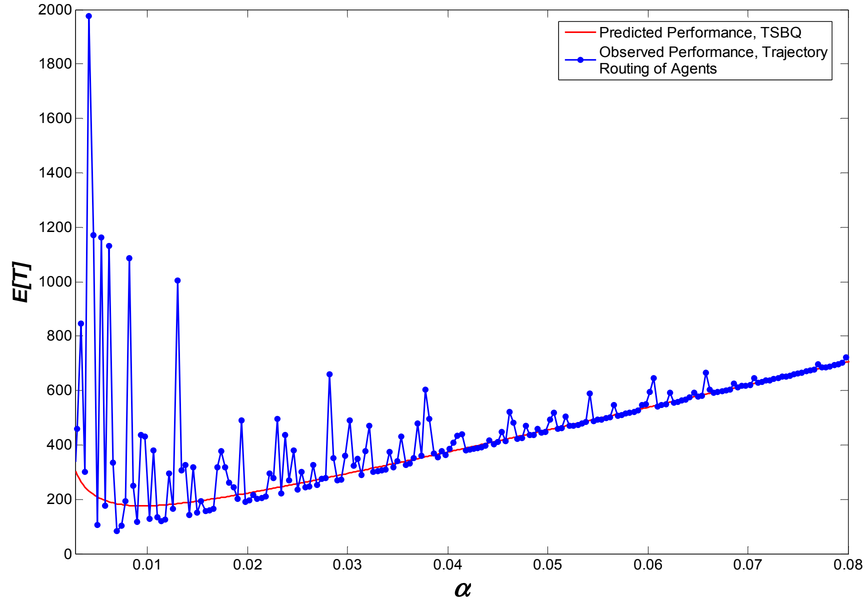


Figure 16. TSBQ with trajectory routing, 5000 nodes, $\delta' = 27$, $E_{rcv}/E_{xmt} = 0.7$.

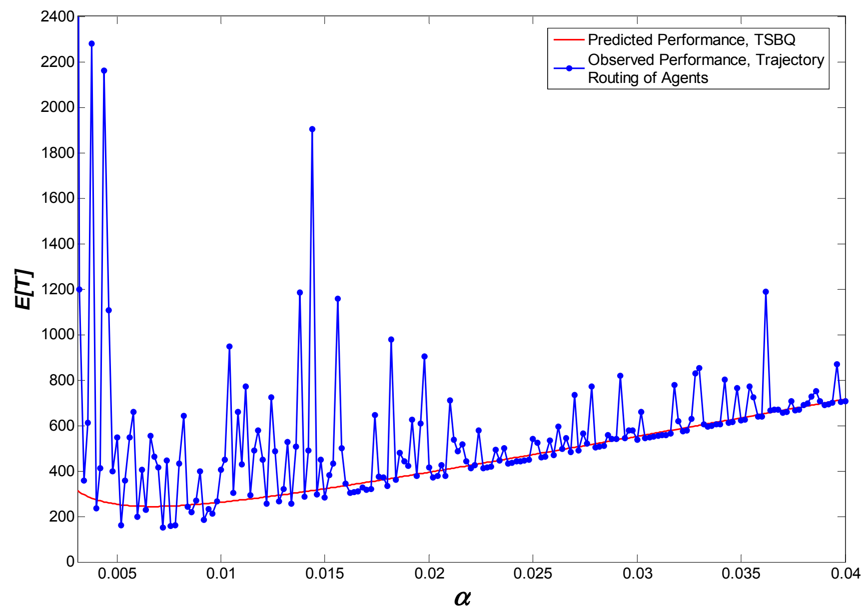


Figure 17. TSBQ with trajectory routing, 10000 nodes, $\delta' = 32$, $E_{rcv}/E_{xmt} = 0.7$.

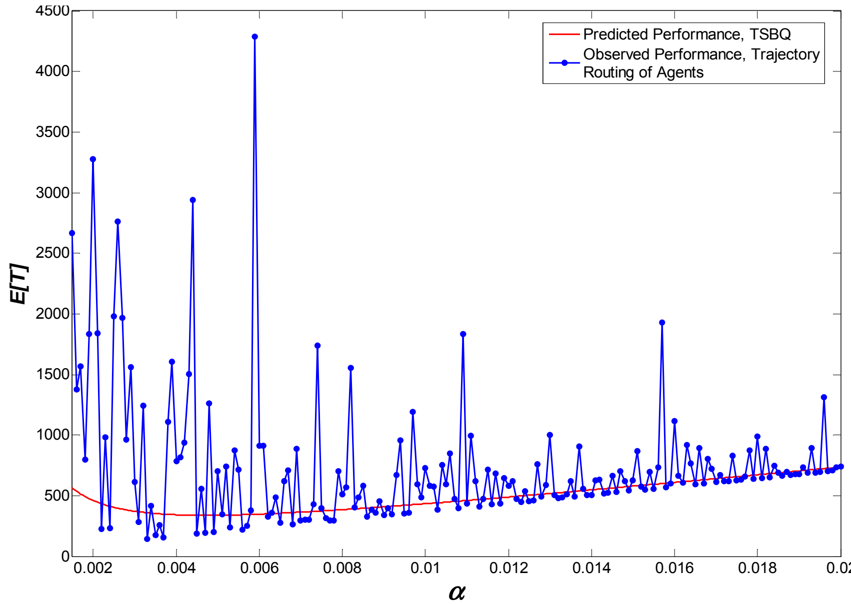


Figure 18. TSBQ with trajectory routing, 20000 nodes, $\delta' = 39$, $E_{rcv}/E_{xmt} = 0.7$.

however, the general trend of the results follows that predicted by (4.11) at higher values of α. For this reason, the use of a feedback-driven caching mechanism to increase the number of informed nodes at little or no energy cost to the network is advocated. The purpose of this mechanism is to decrease the energy expended by the network to answer future queries; it is also useful if the magnitude of n is unknown during the network design phase.

This feedback-driven caching mechanism operates as follows: once a QN locates an informed node, the actual total number of query transmissions required, $x_{\alpha,\delta'}$, is compared to the number of query transmissions expected, $E[X_{\alpha,\delta'}]$. Assuming the OQN becomes an informed node upon receiving the response, a value ρ, $0 \le \rho \le 1$, is computed by

$$\rho = \max\left\{\frac{x_{\alpha,\delta'} - 2E[X_{\alpha,\delta'}]}{x_{\alpha,\delta'} \cdot E[X_{\alpha,\delta'}]}, 0\right\}. \qquad (4.15)$$

Intermediate nodes at each hop in the response's path add the information contained in the response to their own event tables with probability ρ. Although not presented here, experiments indicate this feedback mechanism provides a significant decrease in total energy expenditure for subsequent queries at the expense of total available network storage capacity. Alternatively, nodes recognizing a higher-than-expected number of queries for a particular agent might also forward the high-demand agent autonomously to inform a larger portion of the network, thereby increasing the probability that additional nodes are capable of answering a query. Additional energy savings may also be realized by aggregating updates.

4.4.4 Performance Variance

The mathematical model and the simulation results indicate the variance in the total energy consumed to generate a response can be large, especially at smaller values of α and δ'. Although no mention of a variance analysis of total energy expenditure in the literature has been found, these results can be generalized to any rumor routing-based search algorithm. However, as shown in Figure 19, the variance of total energy expended

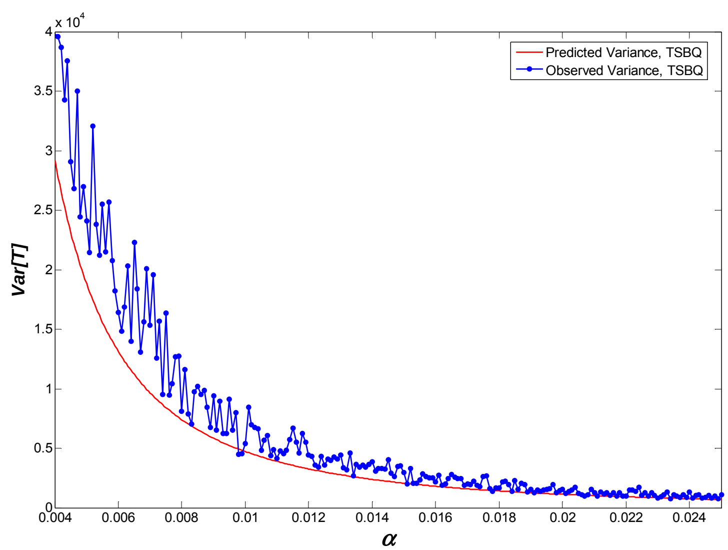


Figure 19. Predicted vs. observed variance of T, $N = 20000$, $n = 1$, $\delta' = 39$.

(and, hence, the number of transmissions and/or latency required to answer a query) is inversely proportional to α. Therefore, if an application requires a query to be answered within a specific number of transmissions (or, alternatively, specifies a maximum latency) with a given probability, the requirement can be met by adjusting α appropriately. The cost of increasing α, however, is an increase in mean per-query energy consumption and a decrease in the total effective storage capacity of the network. The predicted variance based on the choice of α is

$$Var[X_{\alpha,\delta'}] = \sum_{j=1}^{k} j^2 \cdot \Pr\left\{X_{\alpha,\delta'} = j\right\} - \left[\sum_{j=1}^{k} j \cdot \Pr\left\{X_{\alpha,\delta'} = j\right\}\right]^2 . \tag{4.16}$$

In Figure 19, the observed variance of T in the simulations is generally higher than predicted by (4.16) at lower α because a query is dropped if it attempts to travel beyond the defined network boundaries. When a response fails to arrive after the expiration of a timeout period, the OQN may reissue the query along new randomly-chosen trajectories until a response is received; this is the approach used in the simulations. However, if a node chooses random trajectories for reissued queries that result in similar paths through the network, redundant querying of nodes can result. Thus, it may be prudent to limit a node's range of available trajectories in the event that it must reissue a query. Additionally, the predictive value of the model could be improved by incorporating the probability of a query encountering a network boundary. This improvement is discussed in the next subsection.

4.4.5 Network Boundaries and the Analytical Model

The mathematical model (4.11) can be improved by accounting for the effect of a query encountering a network boundary prior to locating an informed node. This requires determining the mean hop-distance between a randomly-chosen node and a random point located on the network boundary. If d is the straight-line distance between a randomly-chosen node and a random point on the network boundary, the expected number of hops, β, before a query encounters a boundary is

$$\beta = \left\lceil \frac{d}{D'} \right\rceil \leq k, \tag{4.17}$$

where D' is the mean distance between transmitter-receiver pairs. Assuming a network of sufficient density, D' is approximately equal to the node transmission range D using MFR routing. The value of d can be determined mathematically or via Monte Carlo experiments. For example, in a square $w \times w$ deployment region such as those used in the simulations, d is approximately $0.65w$. A query that encounters a boundary is expected to have checked $\beta \cdot \delta'$ nodes unsuccessfully. Therefore, the probability of an OQN's original query encountering a network boundary prior to locating an informed node is

$$\Pr\left\{X_{\alpha,\delta'} > \beta\right\} = \left(1 - \frac{\alpha N}{N-1}\right)^{\beta \cdot \delta'}. \tag{4.18}$$

If the OQN is permitted to reissue failed queries using an unrestricted range of trajectories, the expected number of query attempts, n', to locate an informed node is

$$n' = \left(1 - \left(1 - \frac{\alpha N}{N-1}\right)^{\beta \cdot \delta'}\right)^{-1}. \tag{4.19}$$

Because the OQN's choice of trajectories is not restricted in these experiments, there is a non-zero probability of overlap in the regions of subsequent query transmissions. Therefore, a term, ζ, is introduced to account for the energy expended due to nodes being polled more than once in the event a query is reissued. The value of ζ is a function of both the density and transmission range of the nodes, and $\zeta \geq 1$. Using a least mean squares analysis, the value of ζ for the 20000-node network simulations is approximately 1.438, indicating 43.8% of the nodes polled by all reissued queries received the query transmission more than once. Fortunately, the additional energy expenditure due to repeated polling of nodes is only significant at small values of α. At higher α, $n' \approx 1$; hence ζ has little effect. For example, using the value of α^* shown in Figure 6 for the 20000-node network, $n' \approx 1.0314$; thus, only 3% of original queries fail to locate an informed node. The revised model for the expected total energy expenditure is

$$\begin{aligned} E\left[T\right] &= \left(\alpha N - 1 + n\right)\left(E_{xmt} + E_{rcv}\right) + \left(\zeta \cdot n \cdot (n'-1) \cdot \beta\right) E_{xmt} \\ &\quad + \left(\zeta \cdot n \cdot (n'-1) \cdot \beta \cdot \delta'\right) E_{rcv} + \left[2nE_{xmt} + \left(n \cdot (\delta'+1)\right) E_{rcv}\right] E\left[X_{\alpha,\delta'}^{\beta}\right], \end{aligned} \tag{4.20}$$

where $X_{\alpha,\delta'}^{\beta}$ is the expected number of hops required to locate an informed node when network boundaries limit the maximum distance each query may traverse, and

$$E\left[X_{\alpha,\delta'}^{\beta}\right] = \sum_{j=1}^{\beta} j \cdot \Pr\left\{X_{\alpha,\delta'} = j\right\}. \tag{4.21}$$

As seen in Figure 20, (4.20) provides a better prediction of the total energy expended by the network at small α than (4.11). However, (4.11) still provides an accurate means to estimate the values of α^* and δ'^* that result in the least total energy expended without the need to determine ζ.

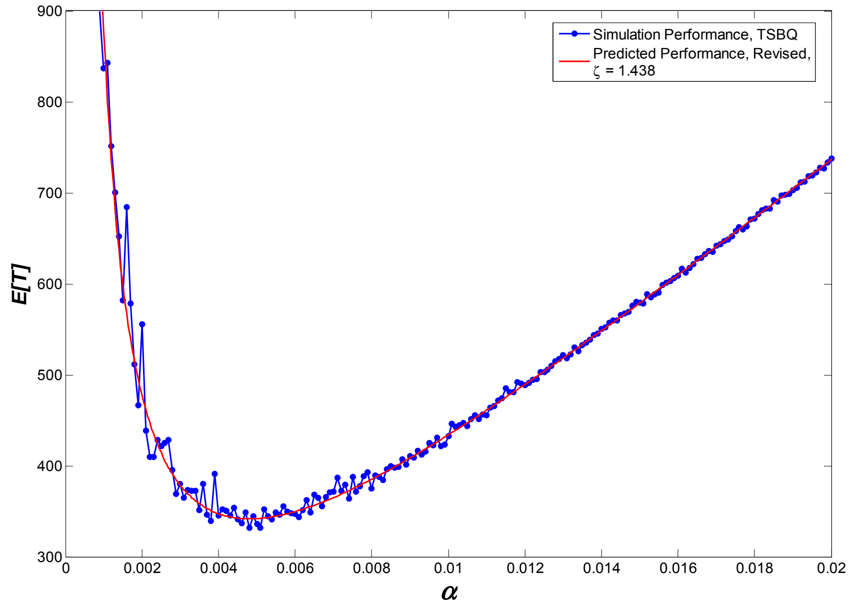


Figure 20. Revised TSBQ performance, 20000-node network, n = 1, δ' = 39.

4.5 Summary

This chapter describes a new search protocol, TSBQ, which minimizes the total energy expended to advertise services/data and respond to queries in large-scale, high-density WSNs. This search protocol is the first to take advantage of the energy efficiency of broadcast transmissions. A mathematical model that predicts the expected total energy expenditure of TSBQ is developed, and the model's parameters are optimized for minimum energy expenditure. This model enables a network designer to consider the effects of node density, memory capacity, data/service popularity, and latency on the total energy expended to answer a query. Finally, the performance variance of TSBQ is analyzed, and a feedback-driven caching mechanism that improves search performance at negligible additional energy cost to the network is provided.

The mathematical model of total energy expenditure can be extended to encompass more general search protocols and network application requirements. For example, if a node needs frequent access to a particular service, the most energy efficient strategy is to locate the service in close proximity to the node. The model can be modified accordingly, thereby increasing the probability of locating the service at a nearby node. Additionally, if improved agent dissemination algorithms are developed (i.e., methods that result in a more uniform initial distribution of informed nodes), these algorithms can be incorporated into the model. Finally, the mathematical model can be easily modified to evaluate the optimum transmission range for networks of nodes that have the capability to vary transmission power.

5. A Queueing Approach to Optimal Resource Replication

5.1 Overview

In the previous chapter, a unique search protocol, TSBQ, was developed. However, TSBQ is designed for networks in which both resources and requests are time-independent and do not expire (or, alternatively, have very long expiration times). In this chapter, a queueing model is developed for analyzing replication strategies for networks in which both resources and requests have limited lifetimes. The model can be used to minimize either the total transmission rate of the network (an *energy-centric* approach) or to ensure the proportion of query failures does not exceed a pre-determined threshold (a *failure-centric* approach). The model explicitly considers the limited availability of network resources, as well as the frequency of resource requests and query deadlines to determine the optimal replication strategy for a network resource. It will be demonstrated that insufficient resource replication increases query failures and transmission rates, and replication levels beyond the optimum result in only marginal decreases in the proportion of query failures at a cost of higher total energy expenditure and network traffic.

Although the mechanisms for advertising and locating resources are well-understood, none of the search protocols previously discussed consider quality of service (QoS) issues such as query deadlines, the proportion of query failures, or the effect of limited resource lifetimes. Additionally, no mention of the effect of resource advertising on the intensity of network query traffic has been found in the literature. Nodes aware of a particular resource have no need to transmit a query to locate this resource; hence, increased resource replication inherently decreases overall query traffic levels. This research considers these effects by providing a node model of search algorithm behavior that minimizes total network transmissions while meeting specified QoS constraints.

Four contributions to the query-based WSN domain are made. First, an analytical queueing model of WSN nodes is developed to assess the total arrival rate of traffic to a node as well as the total proportion of query failures in the network. This model captures much of the behavior of the original rumor routing algorithm [BE02] but extends that research by incorporating deadlines associated with the availability of resources, application timing requirements, and the effect of resource advertising on query traffic levels. Second, the resource replication level that minimizes the total traffic intensity while ensuring a specified upper bound on the proportion of query failures is not exceeded is determined. Third, the effects of various network parameters on search algorithm performance are explained, and it is shown that increasing the replication level of the network beyond a certain threshold is detrimental to network performance from both an energy-efficiency and query-failure perspective. Finally, simulation experiments examine the effects of alternative agent/query lead time distributions on the metrics.

The remainder of this chapter is organized as follows. In Section 5.2, mathematical models of a WSN node's event table and transmission queue are developed. The behavior of the system is characterized using a Markov chain, and the resulting balance equations are solved to determine the steady-state populations of the event table and transmission queue. In Section 5.3, it is shown how discrete optimization problems can be solved to determine the optimal resource replication level by minimizing the total node transmission rate while satisfying query failure constraints. In Section 5.4, the results of simulations are shown using alternative agent/query expiration time distributions.

5.2 Node Model

It is assumed that the wireless sensor network consists of N homogeneous nodes with similar resource requirements and limitations. Over the useful lifetime of the network, nodes are relatively indistinguishable in terms of time spent sensing, sleeping, transmitting, receiving, and computing. Nodes are also similar with respect to their information requirements and the rates at which they observe and report relevant phenomena.

During their lifetimes, nodes are both producers and consumers of network resources. A node produces a resource when it monitors the environment and gathers data on the occurrence of pertinent events or when it offers a particular service to the network. In addition to data gathering, nodes must also 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 poll the network to locate the necessary information and/or services.

The nomenclature adopted in this chapter is consistent with previous chapters. However, small variations in description are required due to the introduction of expiration times. For clarity of discussion, these descriptions are revisited.

When a node senses relevant phenomena or offers a particular service to the network, it advertises this information to a subset of the network by means of an *agent*, a packet that describes the resource available, the location of the resource (or, alternatively, the data itself), and the period of time the resource is available or valid. An agent increases the probability a resource can be located without flooding the entire network with the request. It is assumed agents are transmitted from node to node via a random walk until either the agent's time-to-live (TTL) counter is exhausted or the resource's availability deadline expires.

Upon receiving an agent, a node adds the agent's contents to its local *event table* and is thereby considered *informed* while the resource is available. Only informed nodes are capable of answering the *queries* of uninformed nodes. A query contains at least three pieces of information: the identifier and/or location of the node originating the request, the type of resource sought, and the maximum amount of time the query is permitted to roam the network for an informed node. In a manner similar to 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 originating node, typically via shortest-path routing. The response 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 prior to the expiration of its deadline, the query fails. The desired end state is to minimize the total transmission rate (and, hence, the total rate of energy consumption) required by the network to propagate agents and queries while simultaneously ensuring query failures do not exceed a predetermined limit.

In the remainder of this section, a queueing model that captures the behavior of a node's event table and transmission queue is developed. The model is analyzed to determine the agent replication level that minimizes the expected total rate of transmission arrivals while simultaneously ensuring query failures remain at or below a specific threshold. Finally, the effects of various network parameters on the optimal agent replication level are investigated.

5.2.1 Queueing Model Preliminaries

A typical wireless sensor node is capable of sensing, computing, transmitting, and receiving. Of these activities, transmitting requires the largest energy expenditure [ROG06]. For this reason, minimizing transmissions within the network reduces total energy expenditure and extends the useful lifetimes of the nodes. Additionally, minimizing the amount of traffic in a WSN reduces contention for the transmission medium and decreases the probability of collisions.

As discussed in Chapter 2, a cost-based analysis is frequently used to evaluate the efficiency of WSN search algorithms. Since transmitting a packet typically expends more energy than any other node activity, most search algorithm cost models use the number of transmissions, messages, bits, or hops as their primary performance metric (e.g., [AB04, AyS02, BK03, BA05, BE02, GMS05, JM96, KaK06, KA05, LHZ04, LB04, NSC03, OK04, Sha04, TYD+04]). However, it is difficult to incorporate agent and query deadlines into these cost-based models; hence, there is no opportunity to assess energy-efficient replication strategies that consider agents and queries with timing constraints. In contrast, queueing models provide a relatively straightforward means of associating timing constraints with arriving customers (i.e., agents and queries).

When an agent arrives at a node, the node stores a copy of the agent in its on-board event table. This copy remains in the event table until the agent's lead time (i.e., the difference between the current time and the resource's expiration time) expires. Assuming the agent's TTL counter has not been exhausted, the node also places a copy of the agent in its transmission queue to be forwarded to a neighboring node during a future transmission window. Agents remain in the transmission queue until they are successfully transmitted to a neighboring node or the agent's lead time expires, whichever occurs first.

When a node receives an agent and adds it to the event table, the expected number of hops an arbitrary query must make prior to locating an informed node is reduced. Additionally, a node has no need to transmit a query if the desired information is stored in its event table; as a result, informed nodes transmit less query traffic than uninformed nodes. Therefore, increasing the number of informed nodes decreases the expected number of query transmissions required to locate an informed node and simultaneously decreases the total amount of new query traffic generated by the network. Of course, this decrease in query transmissions comes at the cost of additional agent transmissions.

When a query arrives at a node, the node takes one of two actions. If the node's event table contains the information needed to answer the query, the node replaces the query with the appropriate response and places the response into the transmission buffer for later transmission. If, however, the node is uninformed, the node places the query directly into its transmission buffer. In either case, if the lead time of the query (or resulting response) expires prior to transmission, the query has failed. Otherwise, the query (response) is removed from a node's transmission buffer once it is successfully transmitted. All arrivals to a node's transmission queue, regardless of type, are assumed to be served using a *first-in, first-out* (FIFO) queueing discipline.

A node's transmission buffer can be modeled as a multi-class queue because there are multiple customer types (i.e., agents, queries, and responses) awaiting access to a single server (the transmission medium). Additionally, these customers leave the system (i.e., renege) if they are forced to wait beyond their expiration times. Furthermore, as will be shown below, a node's event table can be modeled as a queue in which customers arrive with specific service time requirements. By tracking the number of agents stored in a node's event table, the proportion of time the node is informed can be determined.

The energy expended to respond to a query is a function of the distance between the informed node and the originating node. Although returning a response to the originating node requires one or more transmissions, it is assumed the amount of response traffic in the network is small compared to the total number of agent and query transmissions. Hence, the node model focuses on optimizing the total number of agent and query transmissions. The problem, then, can be stated as follows: what level of agent traffic is required to minimize the total rate of agent and query transmissions while not exceeding a specified maximum level of query failures?

5.2.2 Agent/Query Transmission Traffic

Answering this question requires defining the parameters used in the node model. These parameters are also summarized in Table 4 at the end of this section. Let E be the total number of possible event types in the network. A single node witnesses a reportable type-i event (or, alternatively, offers a specific service) according to a Poisson process with rate parameter λ_i, where $i \in \{1, 2, \ldots, E\}$. Nodes advertise the availability of this resource by forwarding an agent to $(\alpha_i N - 1)$ nodes, $\alpha_i \in \{2/N, 3/N, \ldots, (N-1)/N\}$, via a random walk using a unicast (single transmitter, single receiver) transmission scheme.

When a type-i agent arrives at a node, its lead time is assumed to be an exponentially distributed random variable with mean $1/\delta_i$. The total expected arrival rate of agents to a node's event table includes its local rate of agent generation, λ_i, plus a proportion of the agents received from the remaining $(N-1)$ nodes. Let A_i be the rate of type-i agent arrivals to a single node. Then the total expected type-i agent arrival rate to a node's event table is

$$E[A_i] = \alpha_i N \lambda_i, \quad i \in \{1, 2, \ldots, E\}. \tag{5.1}$$

A node always attempts to transmit locally-generated agents to at least one neighboring node. Type-i agents received from the remaining $(N-1)$ nodes are also added to the node's transmission queue as long as the agent's TTL counter is not exhausted. Since each agent is initially assigned a TTL of $(\alpha_i N - 1)$, externally-generated agents are added to a receiving node's transmission queue with probability $(\alpha_i N - 2)/(\alpha_i N - 1)$. Therefore, the total expected arrival rate of agents to a node's transmission queue, A_i^{xmt}, is

$$E[A_i^{xmt}] = \left(\alpha_i N - 1\right) \lambda_i, \quad i \in \{1, 2, \ldots, E\}. \tag{5.2}$$

An agent is removed from a node's event table only when its expiration time is exceeded. In contrast, an agent awaiting transmission in the node's transmission queue is removed when the agent is successfully forwarded to a neighboring node or when the agent's expiration time passes, whichever occurs first. If an agent expires in the transmission queue, its copy contained in the event table is also removed since the expiration times for both are identical.

Nodes use type-i queries to locate type-i agents. Assume individual nodes generate type-i queries according to a Poisson process with rate parameter γ_i. If a node's event table contains no information related to its query, the node must transmit the query to the network. Let $\pi_{0,i}$, $0 < \pi_{0,i} < 1$, be the proportion of time that a node is i-uninformed, i.e., the node has no type-i agents in its event table. Then the node adds locally-generated type-i queries to its transmission queue according to a Poisson process with rate parameter $\pi_{0,i}\gamma_i$. Nodes cannot be informed with probability 1; otherwise, the node would never need to transmit a locally-generated query. Likewise, nodes cannot be informed with probability 0 since this means the node never provides a resource or observes the phenomenon of interest.

A node may receive queries originating from the remaining $(N-1)$ nodes. Assume the lead time of an arriving query of type-i is described by an exponentially distributed random variable with mean $1/\beta_i$. Nodes forward queries in the same manner as agents, i.e., a random walk and unicast transmissions. The expected number of times a query must be forwarded before an informed node is located is a function of $\pi_{0,i}$. Therefore, the expected arrival rate of externally-generated type-i queries to a node, τ_i, depends on the proportion of informed nodes in the network, and

$$\tau_i = \pi_{0,i}\gamma_i (N-1)\left[\frac{1}{(N-1)(1-\pi_{0,i})}\right] = \frac{\pi_{0,i}\gamma_i}{1-\pi_{0,i}}. \tag{5.3}$$

The total arrival rate of queries to an i-uninformed node's transmission queue is $\gamma_i + \tau_i$, and the total arrival rate of queries to an i-informed node's transmission queue is τ_i. It is important to note that increasing the number of informed nodes in the network not only reduces the expected number of times a query must be forwarded but also decreases the total number of nodes that may transmit new queries to the network. Combining the above expressions for the rates of type-*i* agent and query arrivals, one can determine the total expected arrival rate of type-*i* agents and queries, $f(\alpha_i)$, to each node, or

$$\begin{aligned} f(\alpha_i) &= \alpha_i N \lambda_i + (\gamma_i + \tau_i)\pi_{0,i} + \tau_i(1-\pi_{0,i}) \\ &= \alpha_i N \lambda_i + \gamma_i \pi_{0,i} + \frac{\gamma_i \pi_{0,i}}{1-\pi_{0,i}}. \end{aligned} \tag{5.4}$$

Now, $\pi_{0,i}$ is a function of α_i, while N, λ_i, and γ_i are parameters; therefore, the objective is to choose α_i such that (5.4) is minimized. The mathematical programming formulation is

$$\begin{aligned} &\text{Minimize} \qquad f(\alpha_i) = \alpha_i N \lambda_i + \gamma_i \pi_{0,i} + \frac{\gamma_i \pi_{0,i}}{1-\pi_{0,i}} \\ &\text{Subject to} \qquad \alpha \in \{2/N, 3/N, 4/N, \ldots, \alpha_{i,\max}\}, \end{aligned} \tag{5.5}$$

where $\alpha_{i,\max} \leq (N-1)/N$. For a finite network, $f(\alpha_i)$ is a discrete function on a feasible region with at most $(N-2)$ possible solutions, and $\alpha_{i,\max}$ is the largest value of α_i that can be supported by the transmission medium. Since flooding an agent to all network nodes has been shown to be an inefficient means for advertising a resource [BE02], it is assumed $\alpha_{i,\max} << 1$. Consequently, (5.5) is a discrete optimization problem which can be solved by enumerating all possible solutions and choosing the value of α_i, called α_i^*, that minimizes $f(\alpha_i)$. However, before this analysis can be completed, $\pi_{0,i}$ must be cast as a function of α_i. This is accomplished in the next subsection by modeling a node's event table as a M/M/∞ queue.

Table 4. Summary of node model parameters.

Parameter	*Description*
N	The total number of nodes in the network
α_i	The proportion of nodes informed by a type-*i* agent, $\alpha_i \in \{2/N, 3/N, \ldots, (N-1)/N\}$
λ_i	Type-*i* agent generation rate (single node)
δ_i	Type-*i* agent expiration rate
γ_i	Type-*i* query generation rate (single node)
β_i	Type-*i* query expiration rate
$\pi_{0,i}$	The proportion of time a node is *i*-uninformed

5.2.3 Event Table as an M/M/∞ Queue

Whether a node is informed of the availability of a specific network resource is determined solely by the presence (or absence) of corresponding agents in the node's event table. A copy of the information contained in each arriving agent is added to a node's event table according to the same process by which agents arrive to a node's transmission queue. Additionally, copies of agents are stored in the event table until their lead times expire. Therefore, for a single type-*i* event, the event table can be modeled as an M/M/∞ queue with arrival rate $\alpha_i N\lambda_i$ and state-dependent service rate $s_i\delta_i$, where s_i is the number of type-*i* agents present in the event table. The proportion of time the event table has no corresponding agents, $\pi_{0,i}$, must be determined. For the M/M/∞ queue, this is equivalent to the well-known result for p_0 [Kle75], or

$$\pi_{0,i} = e^{-\alpha_i N\lambda_i/\delta_i}. \tag{5.6}$$

Recognizing that the on-board storage capacity of a wireless sensor node is necessarily limited in size, it is likely that nodes will not be able to store local copies of every received agent. Therefore, nodes may implement a replacement strategy for event table entries. If a node receives more than one agent advertising equivalent resources, the node can eliminate duplicate entries to make room for other agent types. However, as long as a node always retains a copy of the received agent with the longest lead time (a sensible strategy since it is advantageous to the network for nodes to remain informed as long as possible), then (5.6) accurately reflects the proportion of time a node is uninformed. Consequently, (5.4) may be rewritten as

$$f(\alpha_i) = \alpha_i N \lambda_i + \gamma_i e^{-\alpha_i N \lambda_i / \delta_i} + \frac{\gamma_i e^{-\alpha_i N \lambda_i / \delta_i}}{1 - e^{-\alpha_i N \lambda_i / \delta_i}}, \; i \in \{1, 2, \ldots, E\}. \qquad (5.7)$$

The final step is to determine the value of α_i^*.

5.2.4 Proportion of Query Failures

Although the total arrival rate of agents and queries to a node's transmission queue can now be minimized, the proportion of queries that fail to locate an informed node must also be evaluated. This metric is critical to the network for two reasons. First, when a query fails to locate an informed node, all energy expended by the network to forward the query has served no purpose. Therefore, it is important not only to minimize the rate of transmissions within the network, but also to ensure the energy expended by the network is used effectively to achieve the network's objectives. Second, a node that fails to receive a response to its query may be unable to complete its assigned tasks. If a large number of nodes cannot complete their tasks, the likelihood that the network cannot complete its objectives increases. To simplify the development and analysis of the model and to maintain tractability, it is assumed that failed queries are not reissued by the originating node. Instead, nodes always assign the latest possible deadline to their queries as the data will not be useful after that point in time.

> **Definition**: A *query failure* occurs when a query (or, if the node is informed, the query's corresponding response) expires in the node's transmission queue before it can be transmitted.

The preceding definition accounts for the two possible modes of query failure. First, when a query arrives to an uninformed node, the node places the query into its transmission queue to be forwarded to a neighboring node. If the query's lead time expires before the query can be forwarded, the query has failed. If, however, the query can be transmitted to a neighboring node prior to the expiration of its lead time, the query has not yet failed nor succeeded. Second, if a query arrives to an informed node, the node will generate a response, and the response will be placed into the node's transmission queue. If, however, the response is not transmitted before the expiration time of the original query, the response cannot be returned to the originating node prior to the deadline. In this case, the query has failed even though an informed node has been located.

No service preference is given to either agents or queries in a node's transmission queue; therefore, the long-run rate at which a node transmits either an agent or a query is dependent upon the proportion of agents and queries in its transmission queue. Assume the amount of time required for a node to successfully transmit a single agent or query to a neighboring node is an exponentially distributed random variable with mean $1/\mu$, independent of agent/query type. At this point, only one type of agent and its corresponding query(ies) is considered. Later, the model is expanded to account for the remaining traffic, including multiple agent and query types.

The proportion of query failures at a node depends on the state of the node's event table as well as the number and proportion of agents and queries in the node's transmission queue. The state of the event table determines the arrival rate of queries, and the number and proportion of agents and queries in the transmission queue determines the queries' access to the transmission medium. Therefore, the state of a node is defined by the triplet (l, m, q), where l is the number of agents in the node's event table, m is the number of agents awaiting transmission in the node's transmission queue, and q is the number of queries awaiting transmission in the node's transmission queue. Let $p_{l,m,q}$ denote the steady-state proportion of time the node spends in state (l, m, q). This system can be characterized by the set of balance equations listed in Table 5.

The final row in Table 5 indicates a node can never have more agents in its transmission queue awaiting transmission than agents stored in its event table, i.e., $0 \le m \le l$. For purposes of modeling the desired system, this condition is necessary even if nodes retain only the received agent(s) with the longest remaining lead time(s). Further, 1_x is an indicator function, where

$$1_x = \begin{cases} 1, & \text{if condition } x \text{ is true} \\ 0, & \text{otherwise} \end{cases}. \tag{5.8}$$

Due to the presence of three infinite state variables, the system characterized by the balance equations in Table 5 does not lend itself to a closed form solution. However, the system can be approximated by a set of $(L+1)(L+2)(Q+1)/2$ balance equations,

Table 5. Node model balance equations.

State	*Condition(s)*	*Balance Equation*
(0,0,0)	none	$\left[\alpha_i N\lambda_i + \gamma_i + \tau_i\right] p_{0,0,0} = \delta_i p_{1,1,0} + (\beta_i + \mu) p_{0,0,1} + \delta_i p_{1,0,0}$
(0,0,*q*)	$q \geq 1$	$\left[\alpha_i N\lambda_i + \gamma_i + \tau_i + \mu + k\beta_i\right] p_{0,0,q} = (\gamma_i + \tau_i) p_{0,0,q-1} + \left[\mu + (n+1)\beta_i\right] p_{0,0,q+1}$ $+\delta_i p_{1,1,q} + \delta_i p_{1,0,q}$
(*l*,0,0)	$l \geq 1$	$(\alpha_i N\lambda_i + \tau_i + i\delta_i) p_{l,0,0} = \delta_i p_{l+1,1,0} + \left[(l+1)\delta_i\right] p_{l+1,0,0} + (\beta_i + \mu) p_{l,0,1} + \mu p_{l,1,0} + \lambda_i p_{l-1,0,0}$
(*l*,*m*,0)	$l, m \geq 1, l \geq m$	$\left[(l-m)\delta_i + \alpha_i N\lambda_i + \tau_i + m\delta_i + \mu\right] p_{l,m,0} = (m+1)\delta_i p_{l+1,m+1,0} + \left[\beta_i + \mu/(m+1)\right] p_{l,m,1}$ $+\mu p_{l,m+1,0} 1_{l>m} + (\alpha_i N - 1)\lambda_i p_{l-1,m-1,0} + \left[(l+1-m)\delta_i\right] p_{l+1,m,0} + \lambda_i p_{l-1,m,0} 1_{l>m}$
(*l*,0,*q*)	$l \geq 1, q \geq 1$	$(l\delta_i + \alpha_i N\lambda_i + \tau_i + q\beta_i + \mu) p_{l,0,q} = \left[(q+1)\beta_i + \mu\right] p_{l,0,q+1} + (l+1)\delta_i p_{l+1,0,q}$ $+\tau_i p_{l,0,q-1} + \delta_i p_{l+1,1,q} + \left[\mu/(q+1)\right] p_{l,1,q} + \lambda_i p_{l-1,0,q}$
(*l*,*m*,*q*)	$l, m \geq 1, l \geq m, q \geq 1$	$\left[(l-m)\delta_i + \alpha_i N\lambda_i + \tau_i + m\delta_i + q\beta_i + \mu\right] p_{l,m,q} = (m+1)\delta_i p_{l+1,m+1,q}$ $+\left[(q+1)\beta_i + (q+1)\mu/(m+q+1)\right] p_{l,m,q+1} + \left[(m+1)\mu/(m+1+q)\right] p_{l,m+1,q} 1_{l>m}$ $+(\alpha_i N - 1)\lambda_i p_{l-1,m-1,q} + \tau_i p_{l,m,q-1} + \left[(l+1-m)\delta_i\right] p_{l+1,m,q} + \lambda_i p_{l-1,m,q} 1_{l>m}$
(*l*,*m*,*q*)	$l < m, q \geq 0$	Infeasible state since the number of agents in the transmission queue cannot exceed the number of agents in the event table.

where *L* and *Q* denote the maximum number of agents in the event table/transmission queue and queries in the transmission queue, respectively. Although this introduces blocking probabilities into the model, this effect can be reduced by choosing large *L* and *Q*. The complete set of state diagrams for this variation of the model is provided in the appendix.

The complete set of $(L+1)(L+2)(Q+1)/2$ balance equations has $(L+1)(L+2)(Q+1)/2$ unknowns. However, the sum of the steady-state proportion of time in each possible state must be 1, so the normalization condition is

$$\sum_{l=0}^{L}\sum_{m=0}^{l}\sum_{q=0}^{Q} p_{l,m,q} = 1. \tag{5.9}$$

To determine the steady-state proportion of time in each state, the linear system $AX = B$ is solved for *X*, where *A* is a $((L+1)(L+2)(Q+1)/2) \times ((L+1)(L+2)(Q+1)/2)$ matrix containing the balance equation coefficients of Table 5 and the normalization condition, X is the column vector containing the limiting state probabilities, $p_{l,m,q}$, and B is a column vector of zeros with the exception of the normalization condition represented in the appropriate position by an element of 1. Assuming the existence of A^{-1}, one may obtain X by

$$X = A^{-1}B. \tag{5.10}$$

To compute the proportion of query failures observed by a node, one need only compare the rate of query failures, $q\beta_i$, in each possible state to the local rate of query arrivals. The total proportion of type-*i* query failures, denoted ε_i, is

$$\varepsilon_i = \sum_{l=0}^{L}\sum_{m=0}^{l}\sum_{q=1}^{Q}\left[\frac{q\beta_i}{\gamma_i} p_{l,m,q}\right]. \tag{5.11}$$

5.2.5 The Effect of Other Network Traffic

In general, the level of traffic in a wireless sensor network should remain relatively low to maximize network lifetime. However, depending on the transmission requirements of the network's localization algorithm, medium access control protocol, routing mechanism, and applications, agent/query access to the transmission medium can be somewhat less than that captured by the balance equations in Table 5. Additionally, agents and queries related to other types of resources (i.e., other than the particular resource of interest) compete for access to the transmission medium. Therefore, it is advantageous to examine the effect of worst-case traffic levels on search algorithm performance.

The effect of network traffic unrelated to the agents and queries of interest can be captured by modeling the number of "other" packets in a node's transmission queue as a Poisson random variable with mean θ. The effect of this additional traffic on the agents/queries of interest is an increase in the amount of time spent in the queue. The resulting revised balance equations are contained in Table 6.

Table 6. Balance equations revised to include other network traffic.

State	*Condition(s)*	*Balance Equation*
(0,0,0)	none	$\left[\alpha_i N\lambda_i + \gamma_i + \tau_i\right] p_{0,0,0} = \delta_i p_{1,1,0} + \left[\beta_i + \mu/(1+\theta)\right] p_{0,0,1} + \delta_i p_{1,0,0}$
(0,0,q)	$q \geq 1$	$\left[\alpha_i N\lambda_i + \gamma_i + \tau_i + q\mu/(q+\theta) + q\beta_i\right] p_{0,0,q} = (\gamma_i + \tau_i)\, p_{0,0,q-1}$ $+\left[(q+1)\,\mu/(q+1+\theta) + (q+1)\,\beta_i\right] p_{0,0,q+1} + \delta_i p_{1,1,q} + \delta_i p_{1,0,q}$
(l,0,0)	$l \geq 1$	$(\alpha_i N\lambda_i + \tau_i + l\delta_i)\, p_{i,0,0} = \delta_i p_{l+1,1,0} + \left[(l+1)\delta_i\right] p_{l+1,0,0} + \left[\beta_i + \mu/(1+\theta)\right] p_{l,0,1}$ $+\mu/(1+\theta)\, p_{l,1,0} + \lambda_i p_{l-1,0,0}$
(l,m,0)	$l,m \geq 1, l \geq m$	$\left[(l-m)\,\delta_i + \alpha_i N\lambda_i + \tau_i + m\delta_i + m\mu/(m+\theta)\right] p_{l,m,0} = (m+1)\,\delta_i p_{l+1,m+1,0}$ $+\left[\beta_i + \mu/(m+1+\theta)\right] p_{l,m,1} + (m+1)\,\mu/(m+1+\theta)\, p_{l,m+1,0} 1_{l>m}$ $+(\alpha_i N - 1)\,\lambda_i p_{l-1,m-1,0} + \left[(l+1-m)\,\delta_i\right] p_{l+1,m,0} + \lambda_i p_{l-1,m,0} 1_{l>m}$
(l,0,q)	$l \geq 1, q \geq 1$	$\left[l\delta + \alpha_i N\lambda_i + \tau_i + q\beta_i + q\mu/(q+\theta)\right] p_{l,0,q} = \left[(q+1)\beta_i + (q+1)\,\mu/(q+1+\theta)\right] p_{l,0,q+1}$ $+(l+1)\delta_i p_{l+1,0,q} + \tau_i p_{l,0,q-1} + \delta_i p_{l+1,1,q} + \left[\mu/(q+1+\theta)\right] p_{l,1,q} + \lambda_i p_{l-1,0,q}$
(l,m,q)	$l,m \geq 1, l \geq m, q \geq 1$	$\left[(l-m)\,\delta_i + \alpha_i N\lambda_i + \tau_i + m\delta_i + q\beta_i + (m+q)/(m+q+\theta)\,\mu\right] p_{l,m,q} = (m+1)\delta_i p_{l+1,m+1,q}$ $+\left[(q+1)\beta_i + (q+1)\,\mu/(m+q+1+\theta)\right] p_{l,m,q+1} + \left[(m+1)\,\mu/(m+1+q+\theta)\right] p_{l,m+1,q} 1_{l>m}$ $+(\alpha_i N - 1)\lambda_i p_{l-1,m-1,q} + \tau_i p_{l,m,q-1} + \left[(l+1-m)\delta\right] p_{l+1,m,q} + \lambda_i p_{l-1,m,q} 1_{l>m}$
(l,m,q)	$l < m, q \geq 0$	Infeasible state.

5.3 Numerical Results

In this section, a numerical example illustrates the determination of the optimal replication level for a specific resource based on the results of Section 5.2. Also, the tradeoffs associated with the minimum transmission strategy (the energy-centric approach) and the minimum query-failure strategy (the failure-centric approach) are discussed. Finally, the effect of various parameters on replication levels is explored.

5.3.1 Example: 5000-node Network

For the purpose of analyzing the performance of a 5000-node network, a variation of the optimum energy-centric replication level, α_i^*, is first defined. Let κ_i denote the maximum acceptable proportion of type-*i* query failures as defined by the network application. Then this variation, $\alpha_{\kappa_i}^*$, is the minimum resource replication level capable of meeting the network's highest tolerable bound for the proportion of query failures while simultaneously minimizing the rate of received transmissions. Consequently, $\alpha_{\kappa_i}^*$ is equivalent to the smallest possible value of α_i, $2/N \le \alpha_i \le \alpha_{\max}$, such that $g(\alpha_i) \le \kappa_i$ where

$$g(\alpha_i) \equiv \sum_{l=0}^{L} \sum_{m=0}^{l} \sum_{q=1}^{Q} \left[\frac{q\beta_i}{\gamma_i} p_{l,m,q} \right]. \tag{5.12}$$

Suppose the time to successfully transmit an agent or query at a single node is an exponentially distributed random variable with mean $1/\mu = 0.2$. The goal of this example is to optimize the replication level for a specific resource with agent and query parameters defined by Table 7. For this particular example, the effect of traffic other than that related to the agents and queries of interest is ignored (i.e., $\theta = 0$), and $L = Q = 9$. These values of L and Q are sufficiently large to minimize the effect of blocking probabilities on the solution.

Table 7. Parameters for the 5000-node network example.

Parameter	*Value*
Agent generation rate	0.005 agents/sec/node
Agent expiration rate	0.300 agents/sec
Query generation rate	0.050 queries/sec/node
Query expiration rate	0.500 queries/sec

Following the solution procedure described in Section 2, the mathematical program of (5.5) is solved. The objective function and corresponding optimal solution are shown in Figure 21. Based on the results of this energy-centric analysis, the total number of transmissions is minimized when $\alpha_i = 0.0052$; thus, $f(0.0052) \approx 0.2546$ which corresponds to an agent TTL of $(\alpha_i^* N - 1) = 25$.

The next step is to determine if the proportion of query failures obtained at the computed value of α_i^* is acceptable, i.e., $\varepsilon_i \le \kappa_i$. Using (5.12) yields the results shown in Figure 22. Based on these results, the proportion of query failures at $\alpha_i^* = 0.0052$ is

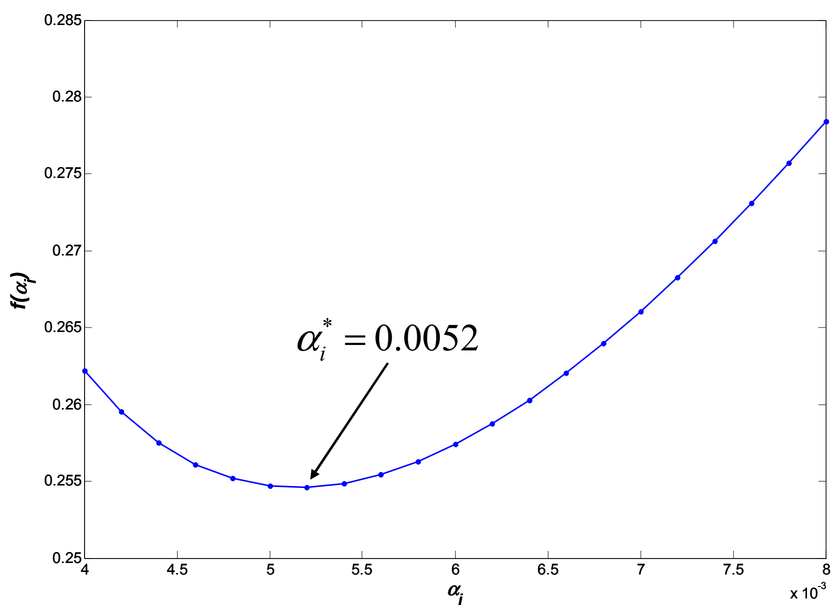


Figure 21. Total rate of arrivals to a node's transmission queue as a function of α_i.

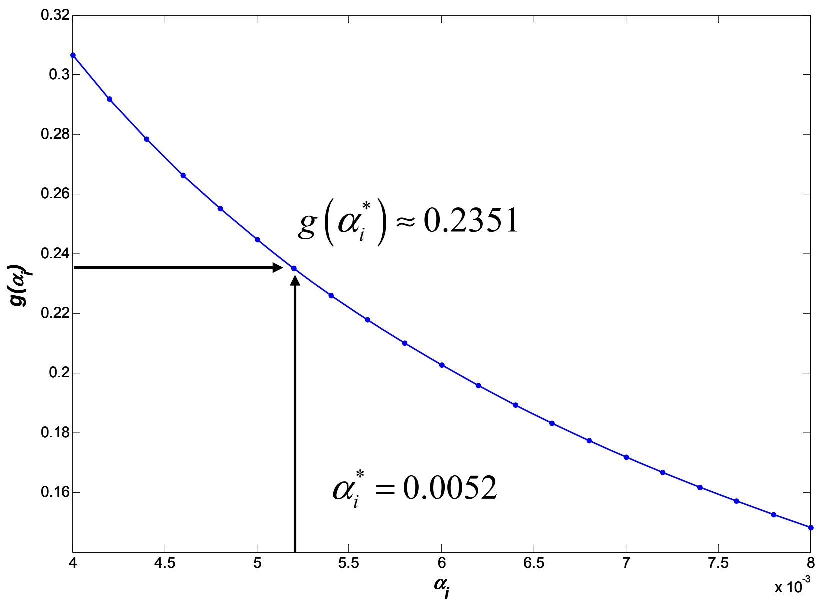


Figure 22. Proportion of query failures as a function of α_i .

$g(\alpha_i^*) \approx 0.2351$. Consequently, it is concluded that approximately 23.51% of all queries received and generated by nodes in this particular network will fail if an energy-centric approach is adopted; this is acceptable only if the application can tolerate this level of query failure.

If, however, the application can tolerate a query failure rate no greater than $\kappa_i = 0.01$, the value of α_i must be increased. The results achieved by examining a wider range of α_i values are presented in Figure 23. Based on this analysis, a value of $\alpha_{\kappa_i}^* = 0.0366$ (i.e., an agent TTL of 182) is necessary to achieve $\varepsilon_i \le 0.01$, and the corresponding rate of received transmissions is $f(\alpha_{\kappa_i}^*) \approx 0.9199$. Therefore, meeting the failure rate requirements of the application necessitates increasing the number of informed nodes per witnessed event by a factor of 7.28. This increases the total rate of transmissions received at each node by a factor of approximately 3.6 and, as a

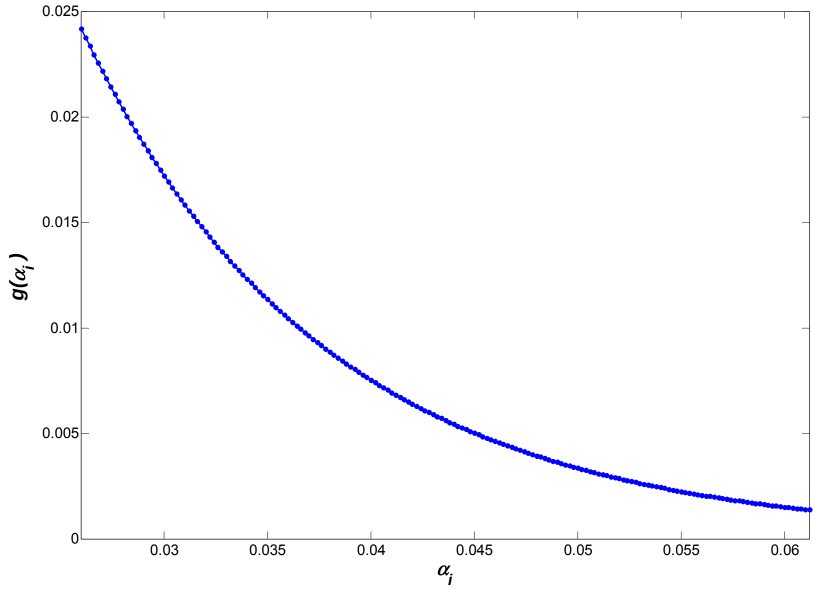


Figure 23. Effect of increasing α_i on query failure rates.

consequence, requires additional energy expenditure to support. Furthermore, practical values of α_i are limited by the network's node density, the intensity of network traffic, node sleep schedules, and the medium access control protocol. Under certain circumstances, namely high node density and heavy traffic, it may not be possible to achieve the desired minimum proportion of query failures. That is, the required replication level necessary to meet the maximum tolerable query failure requirement is greater than $\alpha_{i,\max}$. Hence, in the presence of agent/query timing constraints, the proportion of query failures cannot be reduced indefinitely by increasing the number of resource replicates without bound. On the contrary, the value of α_i must be chosen carefully to prevent excessive query failures due to either insufficient replication or excessive traffic levels.

The effect of α_i on search algorithm development is clear: effective, energy-efficient search algorithms must be capable of managing the number of informed nodes in the network. Failing this, the total proportion of query failures observed at each node cannot be predicted or controlled. Consequently, the stability and reliability of the network's application(s) cannot be assured.

5.3.2 The Effect of Network Parameters on Optimal Replication Levels

During the course of its useful lifetime, a wireless sensor network is subject to several factors that affect optimal resource replication levels. These factors include but are not limited to topology changes due to changing environmental conditions; node addition, deletion, and failure; node mobility; changes in the frequency of sensed events and/or changes in the availability of network resources; and updates to network applications resulting in revised information requirements and deadline constraints. To maintain the desired level of performance, it is important to understand the effects of network parameters on the energy-centric and failure-centric replication strategies. By adjusting various parameters in the analytical model, the resulting effects on the corresponding values of α_i^*, $f(\alpha_i^*)$, and $\alpha_{\kappa_i}^*$ can be observed. The effects of various network parameters are summarized in Table 8.

5.4 Simulation Results

In Sections 5.2 and 5.3, a Markovian model of a WSN random walk search algorithm was developed, and the replication level that minimizes a node's total expected arrival rate of traffic while simultaneously ensuring the proportion of query failures does

Table 8. Effects of parameter changes.

Parameter	α_i^*	$f(\alpha_i^*)$	$\alpha_{\kappa_i}^*$
$\lambda \uparrow$	$\downarrow$	$\uparrow$	$\downarrow$
$\gamma \uparrow$	$\uparrow$	$\uparrow$	$\uparrow$
$\beta \uparrow$ (decreased query lifetime)	unchanged	unchanged	$\uparrow$
$\delta \uparrow$ (decreased agent lifetime)	$\uparrow$	$\uparrow$	$\uparrow$
$\mu \uparrow$	unchanged	unchanged	$\downarrow$
$N \uparrow$	$\downarrow$	unchanged	$\downarrow$

not exceed a predetermined maximum was determined. This model predicts the behavior of networks where the interarrival and lead times of witnessed events and query requests at a node are described by exponentially distributed random variables. However, depending on the characteristics of the network and its associated applications, the lead time of arriving agents and queries may have a different distribution. In this case, it cannot be assumed the Markovian model will correctly describe the system at hand. To examine the effect of different arrival distributions on the node model, a node simulator was constructed in OPNET 10.5, a discrete-time network simulator.

Prior to examining the effects of alternate agent/query arrival distributions, the operation of the OPNET model was compared with the results predicted by the Markovian model. Each data point in Figures 24 and 25 represents the average of three independent replications using different random seeds; the corresponding 95% confidence intervals are also shown. The simulation parameters are identical to those listed in Table 7. As can be seen, the results obtained from the OPNET simulator conform well to those predicted by (5.4) and (5.11).

The effect of continuous uniformly distributed lead times for arriving agents and queries is now examined. As in the previous examples, the mean values of all parameters

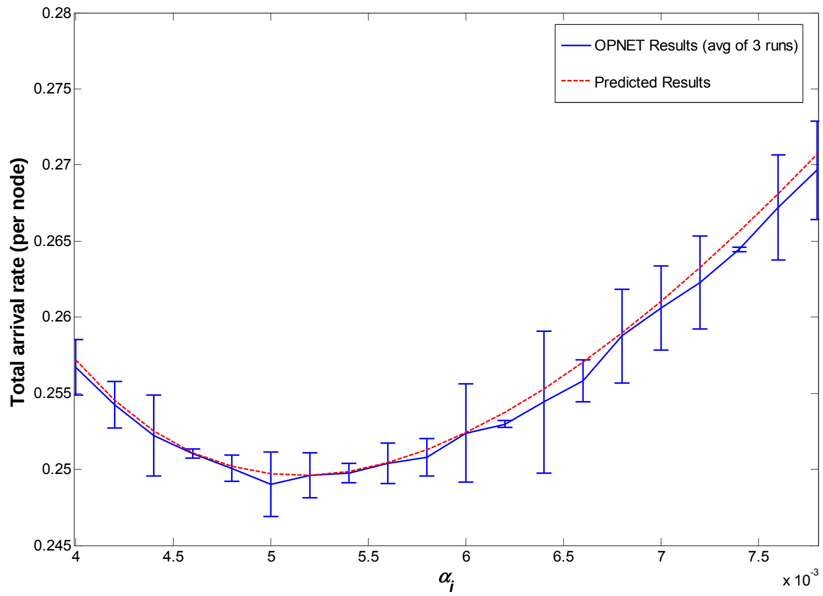


Figure 24. Total arrival rate, predicted versus observed results (Markovian model).

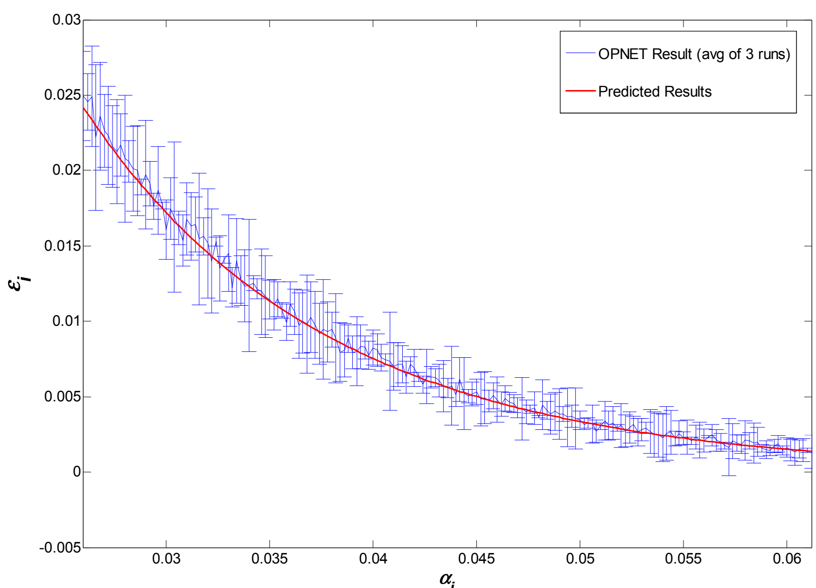


Figure 25. Predicted versus observed results, ε_i (Markovian model).

remain as shown in Table 7, and the mean service time is 0.2. However, the mean lead times of arriving agents and queries are uniformly distributed random variables within the intervals (0,6.6666] and (0,4], respectively.

Since the lead times of arriving agents and queries are no longer exponentially distributed, the behavior of the event table is described by a M/G/∞ queue. Despite the change in the distribution of the service rate, however, (5.6) still characterizes the probability a node's event table contains no applicable agents [Kle75]. Since the assumption of Poisson agent and query arrivals is unchanged, Figure 24 depicts the total rate of arrivals at a node in this system. As a final step, the proportion of query failures of this system is compared to that predicted by the Markovian model. Figure 26 shows the proportion of query failures is lower than that predicted by the Markovian model when the distribution of lead times is uniform. Thus, the Markovian model provides a reasonable upper bound on the corresponding value of ε_i in the event of uniformly distributed expiration times but would tend to overestimate the optimum replication level, α_i^*.

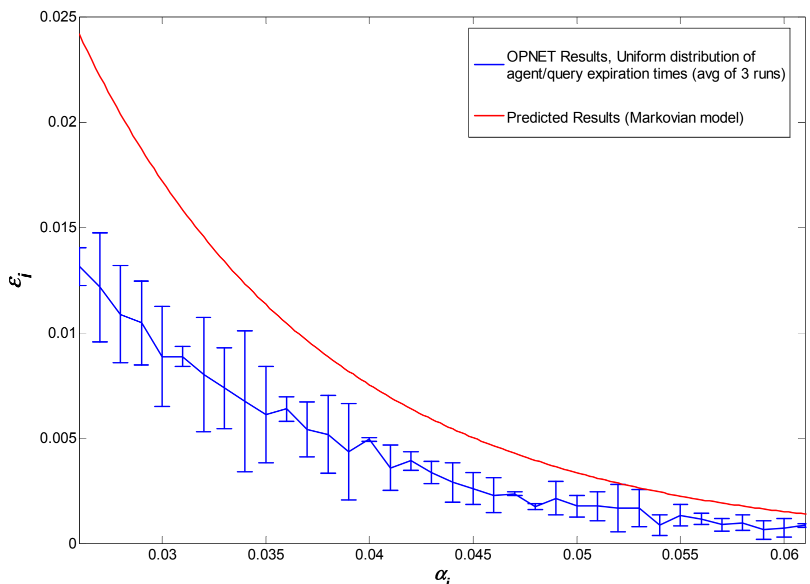


Figure 26. Uniformly distributed agent and query lead times.

5.5 Summary

This chapter characterizes the performance of random walk WSN search algorithms when both agents and queries are assigned expiration times. Using a queueing approach, the appropriate number of resource replicates per observed event required to minimize the total agent/query arrival rate while simultaneously meeting the time-constrained information requirements of the requesting application is analytically determined. Based on the results of analysis and simulation, it is concluded WSN resource replication levels must be carefully managed to achieve efficiency with respect to total energy expenditure and query failures, and this research provides a means to determine the appropriate level. As shown, insufficient resource replication increases energy expenditure (due to excessive query transmissions) and leads to possible application failure. In contrast, excessive replication reduces query failures but needlessly consumes the network's aggregate storage capacity and consumes excessive energy to propagate agents. Excessive replication also increases traffic levels and congestion, thus resulting in a higher proportion of query failures.

It is recognized that the Markovian model developed here is computationally intensive; hence, it is likely better suited for use during the development phase of wireless sensor network design rather than the deployment phase (although approximations can be used to simplify calculations at each node). Therefore, there is merit in deriving a closed form expression for the node model. Unfortunately, due to complicating factors—including the presence of two customer types with dissimilar lead time distributions and state-dependent arrival rates—such an expression may not be tractable.

6. Large Networks with Finite-lifetime Resources and Queries

6.1 Overview

In this chapter, a simulation model is used to examine the performance of a random walk search algorithm for large-population wireless sensor networks in which resources are subject to limited lifetimes and queries are constrained by application-specific deadlines. Specifically, via the TTL parameter, the appropriate number of resource copies that must be created per observed event to minimize the total node arrival rate (the energy-centric approach) is estimated, and the total proportion of queries failures is examined to ensure a specified maximum is not exceeded (the failure-centric approach). Also analyzed is the effect of node transmission range on network performance. The results of the simulation experiments are compared to the queueing-based analytical node model of Chapter 5.

In the previous chapter, a queueing node model was developed to analyze the performance of a random walk search algorithm. To ensure the tractability of the Markovian model, certain simplifying assumptions were required. Most importantly, both requests and advertisements for a particular resource had lead times (i.e., the time remaining until expiration) that, upon arrival at a node, were exponentially distributed with (possibly) dissimilar means. It is more likely, however, for expiration times to be assigned to requests and advertisements by the originating node at the time of generation. When a request/advertisement arrives at a node, the lead time is a consequence of the originally assigned expiration time less any processing, queueing, and transmission delays experienced at previously-visited nodes. Therefore, the actual distribution of lead times of arriving requests and advertisements may not resemble the original distribution. Moreover, the model presumes the expiration time assigned to each agent permits the desired number of agent copies to be stored by the network. That is, the agents' TTL counters are always exhausted before their expiration times occur. Additionally, the distribution of nodes possessing a local copy of a particular agent type is assumed to be uniform throughout the network. As node transmission range is reduced, however, each node's one-hop neighborhood necessarily decreases, thus decreasing both the uniformity of agent distribution and the probability of locating an agent far from its point of origin. Finally, the Markov chain node model assumes the interarrival times of both agents and queries, whether generated locally by the node itself or received from a neighboring node, are exponentially distributed. Whether or not this assumption will hold in a network composed of thousands of nodes is unclear.

While the Markov chain model is useful for predicting the mean performance of individual nodes within the scope of the original assumptions, accurate analytical modeling of the effects of various lead time distributions, agent deployment methods, and transmission range on overall network performance is difficult; studies of such parameters are currently limited to simulation models. The purpose of this chapter is to determine how effects that are difficult or impossible to capture in the analytical model affect the performance of a random walk search algorithm in a network.

The remainder of this chapter is organized as follows. In Section 6.2, a stochastic simulation model of a wireless sensor node that incorporates each node's event table, transmission queue, transceiver, sensors, and applications is developed. Two important indicators of network performance—the total arrival rate and the total proportion of query failures—are discussed in Section 6.3. The results of simulations of networks with large node populations are analyzed in Section 6.4. Section 6.5 provides a summary of this chapter.

6.2 Node Model

To examine the effects of various parameters on the performance of random walk search algorithms, each node is modeled in OPNET as a wireless transceiver with a fixed maximum transmission/reception range, an event table, and a transmission queue (Figure 27). The activity of an *on-board sensor* is represented by a processor which creates new agents in response to external stimuli, and the *application* creates queries for information needed to complete node tasks. The purpose of the *splitter* is to ensure copies of agents received from neighboring nodes are forwarded to the event table and— if the agent's TTL counter has not been exhausted—also to the transmission queue to be scheduled for

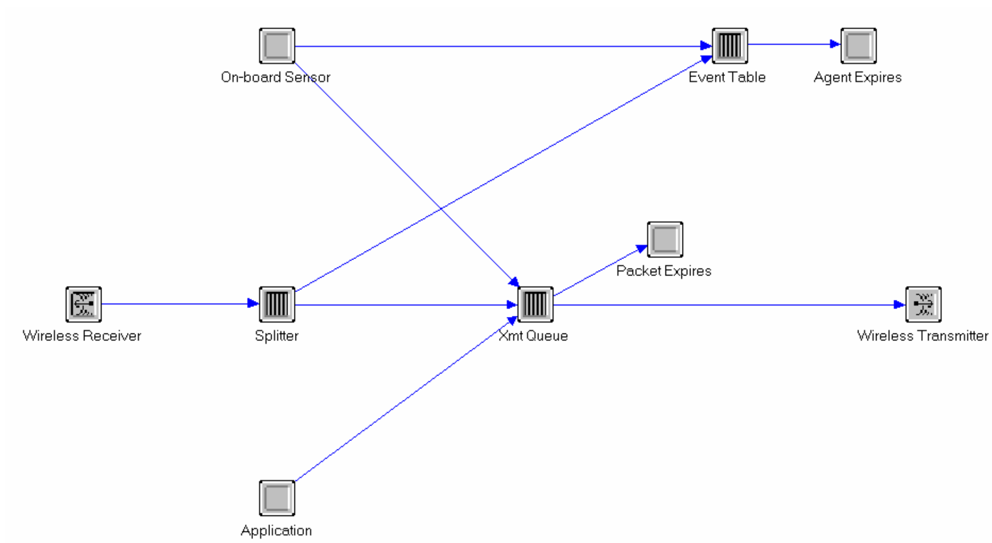


Figure 27: Wireless sensor node model in OPNET.

forwarding to a neighboring node. The splitter has no effect on queries other than to forward the query or its corresponding response directly to the transmission queue. Since the splitter performs a simple function, it adds no additional processing delay to arriving agents or queries.

Each agent arriving to the event table is retained until its expiration time passes. Hence, the operation of the event table resembles that of a $G/G/\infty$ queue. If the event table contains at least one unexpired agent of a particular type, the node is considered to be informed of that event and capable of answering related queries. When the node's application generates a query, the node first checks its local event table for a corresponding agent. If a matching agent is found, the query is answered locally; there is no need to add the query to the transmission queue. However, if the node is uninformed, or if the query originated externally, the query (response) is sent to the transmission queue and scheduled for transmission using a FIFO service discipline. Due to contention for access to the transmission medium, as well as the potential for retransmissions, it is assumed each agent/query requires an exponentially distributed amount of time to be successfully transmitted to the designated receiver. Prior to the beginning of each query transmission, the node checks its event table for an agent that matches the query's request. If the desired information is found, the node transmits the appropriate response in place of the query. If no corresponding agents are found, the node transmits the query to a randomly-chosen neighbor. Agents and queries expiring prior to transmission are removed from the transmission queue. The transmission queue is therefore a FIFO G/M/1 queue with customer reneging as described in Chapter 5.

A network of nodes based on the analytical node model in Chapter 5 resembles a Jackson network of queues. The random arrival of agents and queries to each node are assumed to occur according to a Poisson process, the random time between successive departures of agents and queries from a node's transmission queue is exponentially distributed, and agents/queries are either forwarded to another node or depart the system with specific probabilities. However, the problem is complicated by the existence of three customer types (i.e., agents, queries, and responses), and each customer type must vie for access to the transmission medium at each node. Moreover, the rate of arrival of agents to each node, as well as the expiration time assigned to each agent/query, determines the probability that a query will be forwarded to a neighboring node or depart the system (i.e., fail). Even so, it will be shown in Section 6.4 that the analytical node model provides an accurate prediction of mean network performance.

Node parameters that can be modified by the user prior to execution of the simulation model are summarized in Table 9. All nodes within the network are assumed to be indistinguishable with respect to these parameters. The primary means for controlling the number of resource copies per agent stored in the network is through the TTL parameter. The next section discusses the TTL parameter and the significance of the chosen metrics.

6.3 Metrics

There are two primary indicators of network performance to be measured: the mean total arrival rate of agents and queries (as a proxy for energy expenditure) and the

Table 9: User-adjustable simulation parameters.

Module	*Parameter*	*Description*
On-board Sensor	TTL	The maximum number of times a single agent may be transmitted
	λ	The mean arrival rate of reportable (i.e., agent-generating) events
	δ	The mean lead time assigned to an agent upon its generation
Application	γ	The mean arrival rate of queries generated by the node's application
	β	The mean lead time assigned to a query upon its generation
Transmission Queue	μ	The mean time required to process and successfully transmit an agent/query to the intended recipient

total proportion of failed queries. Using these metrics, the agent TTL required to minimize the total transmission energy expended by the network while not exceeding the maximum tolerable level of query failures is estimated.

Since the node model assumes agents, queries, and responses are forwarded by the transmitting node to a single receiver, measuring the total rate of transmission arrivals at each node is indicative of the network's total energy expenditure and, hence, network lifetime. The goal of the energy-centric metric, then, is to minimize the total rate at which transmissions are received by each node and, as a consequence, to reduce the network's total energy expenditure. Sole reliance on an energy-centric metric, however, cannot guarantee nodes receive information at a rate that is sufficient to satisfy application requirements and also accomplish the network's objectives.

If a sufficient percentage of each node's queries remain unanswered, the probability of general network application failure increases. Therefore, it must be ensured that the total proportion of failed queries observed by each node is less than the application-specific threshold. Query failures are defined using the definition from Chapter 5.

> **Definition**: A *query failure* occurs when a query (or, if the node is informed, the query's corresponding response) expires in the node's transmission queue before it can be transmitted.

Based on this definition, the proportion of query failures in the network, ε, is obtained by dividing the total number of expired queries/responses observed in the network by the total number of unique queries generated. The goal, then, is to ensure ε does not exceed a specified maximum.

6.4 Simulation Results

An essential first step is to validate the simulation model by configuring it to adhere as closely as possible to the assumptions made in the analytical queueing model. Most importantly, the analytical model assumes agents are uniformly spatially distributed throughout the network. As noted previously, however, short node transmission ranges affect the uniformity of agent dispersal. Therefore, to ensure the simulation achieves a uniform distribution of informed nodes, the transmission range of the nodes is artificially extended (via simulation parameters) such that each node is a one-hop neighbor of every other node in the network; the effects of medium contention are momentarily ignored. The nodes are configured according to the parameters in Table 10.

The placement of nodes within the confines of the deployment area is determined randomly using the random topology generating feature of OPNET prior to the beginning of the simulation. This topology, once created, is held constant throughout each set of simulation experiments to ensure any effects due to node placement are identical across

Table 10: Parameters for simulation validation.

Parameter	*Distribution*	*Mean*
Agent interarrival time	Exponential (λ)	200.000 sec/agent
Agent lead time	Exponential (δ)	10.000 sec
Query interarrival time	Exponential (γ)	20.000 sec/query
Query lead time	Exponential (β)	40.000 sec
Transmission time	Exponential (μ)	0.200 sec/packet
Number of nodes	Constant (N)	1000 nodes
Deployment area	Constant	3335m x 3335m
Node transmission range	Constant	>5000m (Isotropic)

each test set. Experimental testing indicated that a warm-up period of 60 seconds was sufficient to cover the transient period. Therefore, for each set of parameters, the network is permitted to operate for a period of 60 seconds prior to the collection of performance data.

After initialization is complete, performance data is collected at every node in the network for a simulated time period of 900 seconds. The 900 second interval was selected because the results obtained after 900 seconds were determined to be statistically indistinguishable from the results obtained when using longer time periods (e.g., 24 hours), and the shorter time period enabled a larger number of experiments to be completed in a fraction of the time. Three replicates of each simulation experiment were conducted; at this level of experimental replication, the standard deviation in the results was consistently less than 0.01. The total arrivals per node per second and the total proportion of failed queries in the network are shown in Figures 28 and 29. Where depicted, 95% confidence intervals are used.

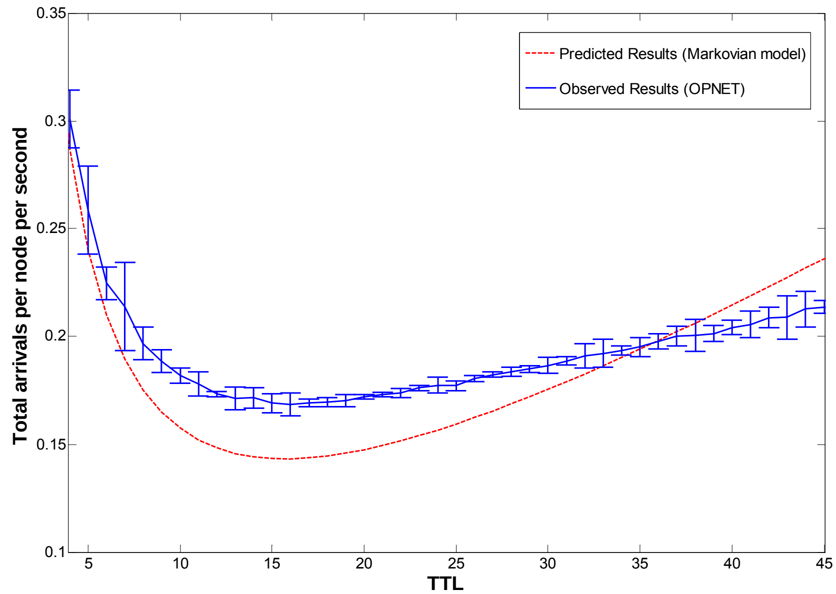


Figure 28: Total arrival rates, infinite transmission range.

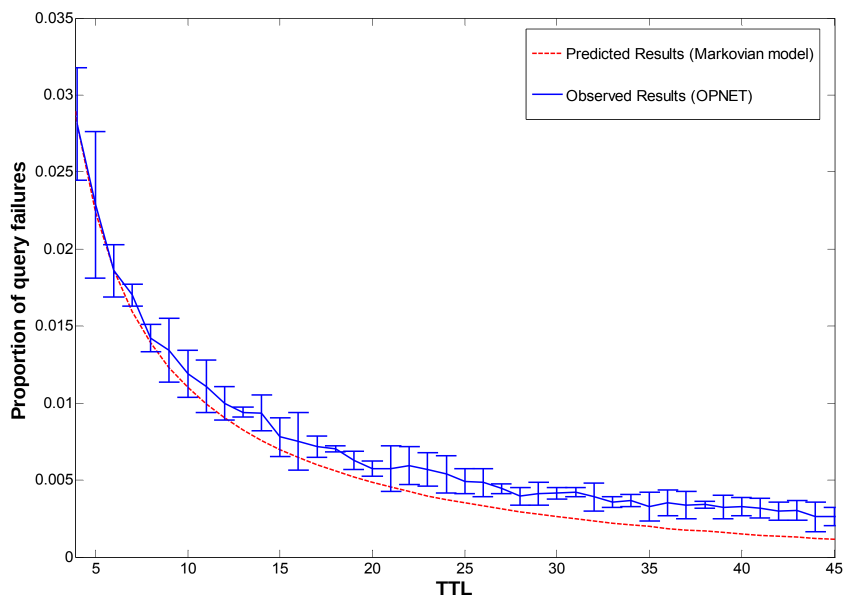


Figure 29: Total proportion of query failures, infinite transmission range.

The results of the simulation experiments using a large node transmission range indicate the analytical node model closely predicts the performance of the network. However, for TTL values less than 36, the arrival rate per node in the simulations is slightly higher than predicted. Although the y-axis scaling used in Figure 28 may imply a sizeable discrepancy between the analytical and simulation results, the maximum differential is a modest 2.7 additional packets per node per 100 seconds of simulation time. This additional traffic is attributed to the fact that agents generated in the simulation may expire prior to exhausting their TTL counters, whereas the analytical model assumes each agent is replicated exactly TTL times prior to expiration. The result is that the actual proportion of the network informed of an event at any given instant is smaller than that assumed by the analytical model. Lower replication levels require the network to support additional query transmissions to locate an informed node. As shown in Figure 29, the need for additional query transmissions causes a slightly higher query failure rate than predicted due to increased latency.

As TTL values increase beyond 36, the total arrival rate predicted by the analytical model is greater than that observed in the simulations. This occurs because only a fraction of the agents generated in the simulation will be replicated more than approximately 40 times as a consequence of the mean agent expiration time and the time required for each agent transmission, i.e., $E[\mu]/E[\delta]=40$. Based on the network parameters, TTL values in excess of 40 create few additional replicates due to agent expiration; hence, total arrivals per node and the proportion of query failures remain relatively constant despite an increase in TTL. Although the analytical model predicts higher arrival rates and lower failure rates than observed, this is anticipated by the $\alpha_{\max}$ parameter discussed in Chapter 5. The $\alpha_{\max}$ parameter recognizes that there is an upper limit to the proportion of the network that can be informed by agents as a consequence of network congestion and/or limited agent lifetimes. Momentarily ignoring the effects of congestion, the value of $\alpha_{\max}$ is approximately 40 for this network.

Despite the minor differences noted between the analytical and simulation models, the analytical model requires a TTL value of 16 to minimize the total arrival rate of traffic to each node and, thus, to minimize the mean total node arrival rate of the network. Additionally, the predicted proportion of query failures is within 0.001 of the observed value when the TTL is 16 and does not exceed 0.0015 for $\text{TTL} \leq 45$. Based on these results, it is concluded that the simulation model provides an accurate representation of the performance of a random walk search algorithm when both agents and queries are assigned expiration times. Although the queueing model developed in Chapter 5 was designed to predict the performance of a single node operating within a narrow set of assumptions, simulations indicate that the model provides a reasonable approximation of the performance of a general network with thousands of nodes. In the following subsections, the effects of node transmission range and decreasing mean agent/query expiration lead times on performance is examined.

6.4.1 Varying Node Transmission Range

When a node's transmission range is limited such that its one-hop neighborhood consists of only a small subset of the total network nodes, the distribution of informed nodes is less likely to conform to the uniform distribution assumed by the analytical model. Therefore, it is expected that shorter node transmission ranges will require higher TTL values to achieve the minimum rate of arrivals, and the minimum rate of arrivals will be higher than that predicted by the analytical model. Additionally, the proportion of failed queries will increase due to the greater number of hops each query is expected to make prior to locating an informed node. Experiments using maximum effective node transmission ranges of 300m, 400m, 600m, and >5000m were conducted using the same parameters shown in Table 10. The results of these experiments are shown in Figures 30 and 31.

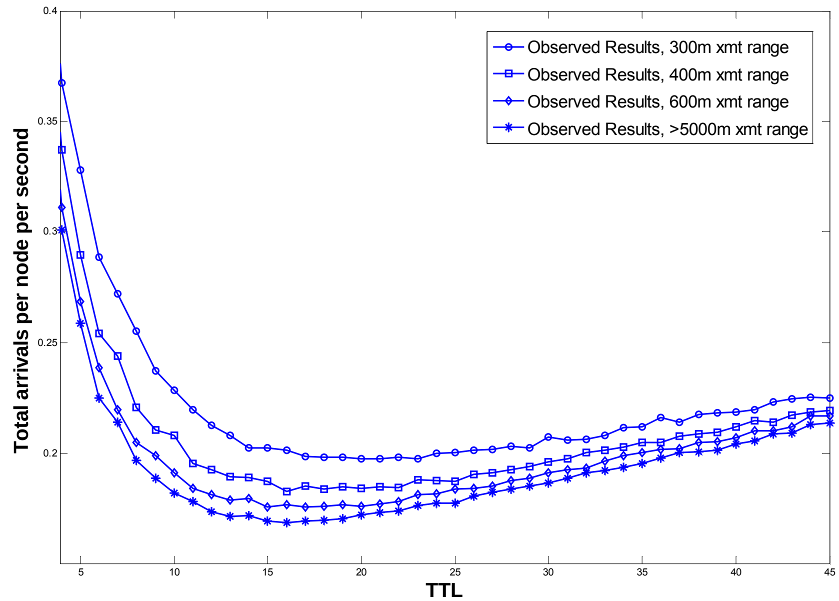


Figure 30: Mean total arrival rates, varying node transmission range.

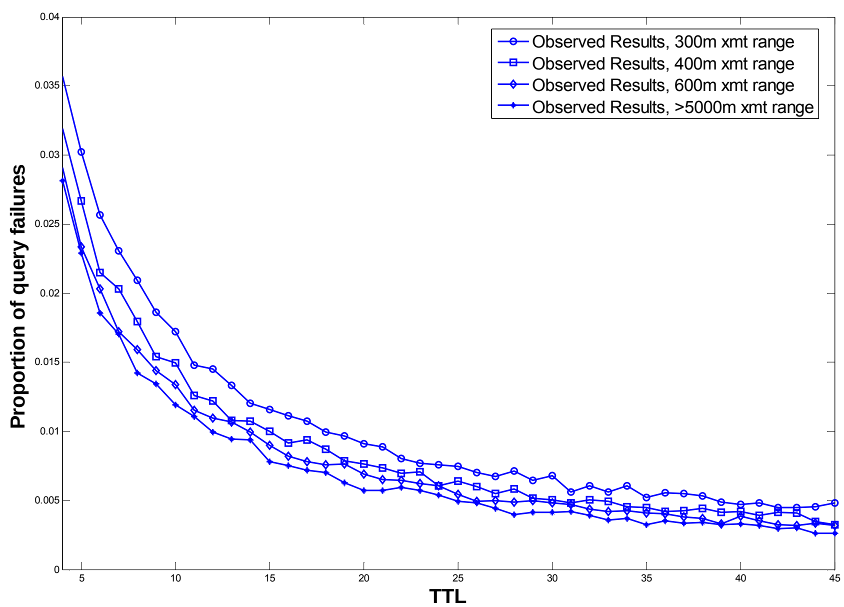


Figure 31: Proportion of query failures, varying node transmission range.

As expected, the simulations confirm higher TTL values are required to achieve the minimum mean total arrival rate as the maximum effective node transmission range is decreased (see Table 11). This implies that there is a tradeoff between the energy expended for transmission and the total number of transmissions required by the search protocol. While nodes with short transmission ranges expend less energy per transmission, and generally experience reduced contention for medium access as compared to nodes with longer transmission ranges, the number of transmissions required per node per second is higher.

Additionally, nodes with longer transmission ranges have a smaller proportion of query failures for a given TTL value. However, increasing the transmission range of wireless sensor nodes requires an exponential increase in energy expenditure [Rap96]. As long as the network remains connected, the resulting increase in total arrival rate observed when using reduced node transmission ranges is outweighed by the reduction in total energy required for transmission. Consequently, when considering energy efficiency, shorter node transmission ranges result in less total energy expenditure despite an increase in the minimum observed total arrival rate.

Table 11: Observed TTL values that minimize total arrival rates.

Transmission Range	*Observed TTL Value*	*Observed Arrival Rate*
300m	20	177.576
400m	16	164.297
600m	15‡	157.828
>5000m	16	151.611

‡ For the 600m transmission range case, the results observed for TTL values of 15 and 16 are statistically indistinguishable.

6.4.2 Decreased Mean Query Lifetimes

If query lifetimes are reduced in response to application requirements, preventing an unacceptably high proportion of query failures will necessitate decreasing the amount of time required by a query to locate an informed node. If the mean effective transmission rate of the network is fixed, the only remaining recourse is to increase the number of informed nodes in the network. To examine the effect of decreased mean query lifetime on network performance, additional experiments were conducted using exponentially-distributed query lifetimes with means of 10, 20, 30, and 40 seconds. The results of these experiments are shown in Figures 32 and 33. The maximum node transmission range for these experiments is fixed at 400m.

As shown in Figure 32, total arrival rates are only marginally reduced by decreasing the mean query lifetime (a consequence attributed to reduced traffic due to query expiration). However, the resulting increase in the proportion of query failures necessitates higher TTL values to achieve the same proportion of query failures observed

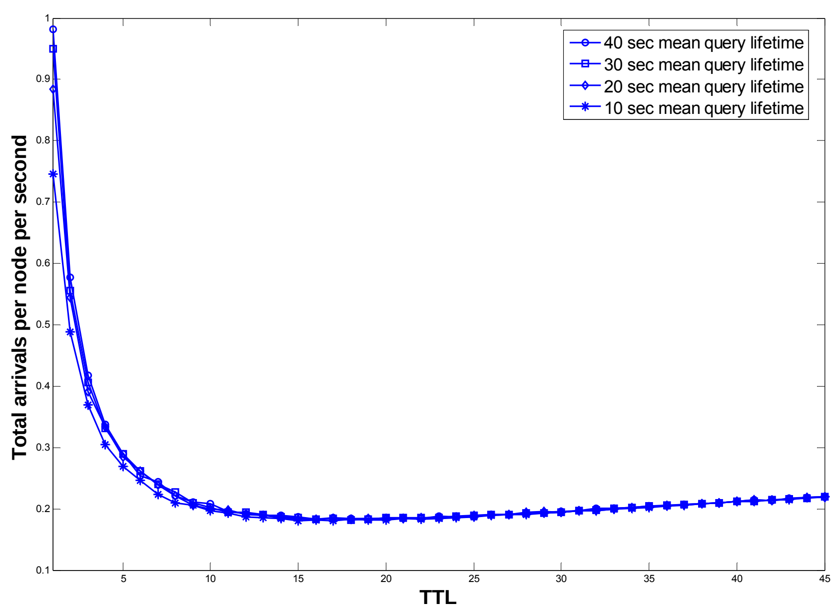


Figure 32: Total arrival rates, varying mean query lifetime.

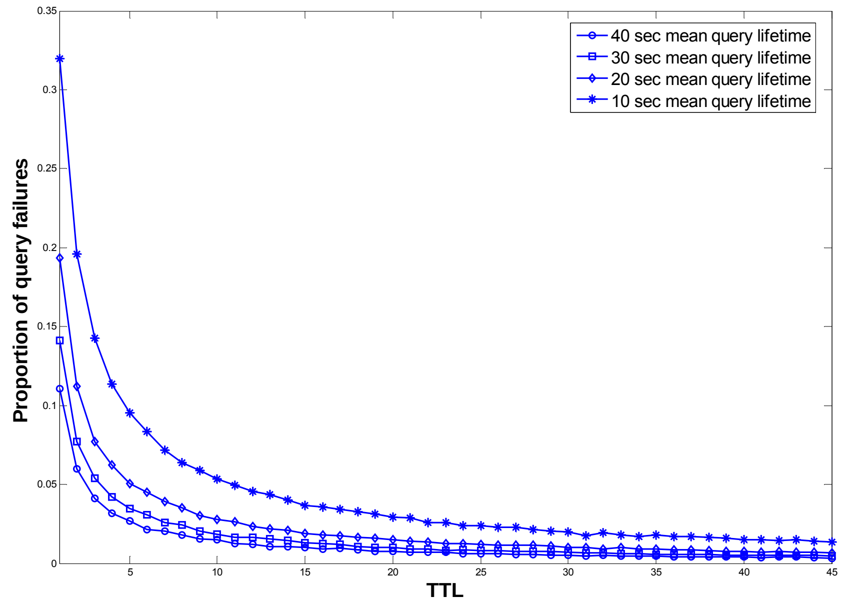


Figure 33: Proportion of query failures, varying mean query lifetime.

when queries have longer mean lifetimes. These results verify the intuitive link between query latency (i.e., the time required by the network to answer a query) and energy expenditure.

6.5 Summary

The choice of MAC protocol affects the performance of the network. In these simulation experiments, it was assumed that network traffic is very low; thus, the probability of a transmission collision is correspondingly small. This is a valid assumption in energy-constrained WSNs. Accordingly, the network's MAC protocol is modeled by requiring each node to expend an exponentially-distributed amount of time to successfully transmit a query or agent to a neighboring node. Additionally, the distribution of the random time required for a successful transmission is assumed to be unchanged across the range of traffic intensities tested. However, it is probable that the distribution of the time required by the MAC protocol to facilitate a successful transmission may change as node densities and/or traffic levels increase.

The simulations indicate the Markovian queueing node model in Chapter 5 provides a reasonable approximation for the performance of a random walk search algorithm in large-population sensor networks. However, it may be possible to refine the model to better predict the performance of large networks of nodes with varying transmission ranges and mean agent/query lifetime distributions. Most importantly, the proportion of nodes informed by an agent, α, could be modified to reflect the fact that some agents will not exhaust their TTL counters prior to expiration. Consequently, the proportion of informed nodes is somewhat smaller than expected.

7. Conclusions and Contributions

This chapter summarizes the key results and defines the specific contributions of this dissertation. These results and contributions are organized by the corresponding chapter in which the information is first presented. Future research is also proposed.

7.1 Trajectory-based Selective Broadcast Query Protocol

The TSBQ protocol is an original hybrid push-pull search protocol that minimizes the expected total energy expenditure of the network to advertise resources and answer queries in wireless sensor networks. Due to the inherent computational, memory, and energy limitations of wireless sensor nodes, the protocol is specifically designed for energy efficiency, scalability, and simplicity. A probabilistic model of the energy expended by the protocol was developed, and the model was analyzed to determine the optimum number of resource replicates required per witnessed event to minimize the expected total network energy expenditure. The protocol was extensively analyzed via simulation, and the results of the simulations were compared to the forecasts of the analytical model.

7.1.1 Results

The main results of this phase were:

- Via an analytical model and simulation experiments, the scalability of TSBQ was demonstrated by showing that TSBQ consumes a smaller percentage of the network's aggregate storage capacity as the number of nodes in the network increases.

- As the energy expended for transmission increases, the number of resource replicates required for minimum expected total energy expenditure decreases, and the optimum node density increases.
- As the energy expended for reception increases, the number of resource replicates required for minimum energy expenditure increases, and the optimum node density decreases.
- The expected total energy expended by TSBQ is significantly less than that consumed by unicast-based search algorithms.
- When the network's node density is less than or equal to the critical value, δ'^*, TSBQ performs at least as well as broadcast-based search algorithms. When the node density is greater than δ'^*, TSBQ consumes less total energy than broadcast-based search algorithms.
- Increasing the popularity of a resource by an order of magnitude results in a linear increase in the optimum number of resource replicates and an approximately linear decrease in the optimum number of designated receivers per query transmission, δ'^*.
- The effect of network boundaries on TSBQ performance is only significant at replication levels well below the value of α^*.
- The variance in total energy expenditure associated with a query decreases exponentially as the number of resource replicates in the network is increased. This insight provides a means to control the expected amount of latency

associated with a particular query, i.e., decreased query latency is achieved by increasing the number of resource replicates in the network.

7.1.2 Contributions

The unique contributions of this phase of research may be summarized as follows:

- A new search protocol, TSBQ, designed specifically to operate effectively within the computational, energy, and memory constraints of wireless sensor networks, was proposed. TSBQ is the first protocol to incorporate the hardware power requirements of the nodes and resource popularity when determining the optimum (energy efficient) number of resource replicates. Additionally, TSBQ is the first search protocol to take advantage of the broadcast nature of wireless transmissions to minimize energy expenditure by determining the optimum number of designated receivers for each query transmission.
- An analytical model of TSBQ was developed, and the means to optimize TSBQ's parameters for energy-efficient performance was demonstrated. Furthermore, it was shown how the TSBQ mathematical model can be extended to support analysis of other rumor routing-based search protocols.
- A feedback-driven caching mechanism was developed to provide improved performance at negligible additional energy cost to the network.

7.2 A Queueing Approach to Optimal Resource Replication

Although the mathematical model developed for analysis of TSBQ accurately predicts system performance, it is difficult to include the concepts of lifetime-limited resources and time-constrained queries into probabilistic models. Also, there are no analytical models in the current literature to assist in the analysis of the effects of agent/query expiration times on optimal resource replication levels. To address this void, an analytical node model of a random walk push-pull search algorithm was developed, and the model was analyzed to determine appropriate resource replication levels for large-scale wireless sensor networks. The optimum resource replication level was determined based on minimizing total expected energy expenditure while simultaneously ensuring the maximum specified proportion of query failures is not exceeded.

7.2.1 Results

- The effects of increasing resource replication levels on system performance were identified. It was shown that increasing the number of resource replicates beyond the optimum without bound causes total node arrival rates to increase linearly while only marginally decreasing the proportion of query failures.
- The effects of alternative agent/query lead time distributions were identified via a simulation model. Specifically, it was shown that a uniform distribution of agent/query lead times results in a decrease in the total proportion of query failures when compared to exponentially-distributed lead times with identical means.

7.2.2 Contributions

- An original analytical node model based on queueing theory was developed to analyze the effects of lifetime-limited resources and time-constrained queries on search protocol performance. This model is the first to (1) describe a node's event table as an M/M/∞ queue, (2) account for the effect of resource advertising on query traffic levels and transmission rates, and (3) permit analysis of deadlines associated with the availability of resources and application timing requirements.

- The concepts of "energy-centric" and "failure-centric" analyses were introduced as a means to differentiate between the dual objectives of reducing total network energy expenditure and ensuring the proportion of failed queries does not exceed a specified maximum.

7.3 Evaluation of the Analytical Node Model in Large Networks

In this phase of research, the ability of the previously-developed node model to predict the performance of a random walk search algorithm in highly-populated networks was determined. This was accomplished by incorporating the node model into a large-scale simulation using OPNET, a discrete-event network simulator. This permitted analysis of the effects of a wider spectrum of parameters on search algorithm performance than those that can be feasibly included in the queueing model. These additional effects include node transmission range and power, alternative agent/query interarrival time and lead time distributions, and replication limits based on expected agent lifetimes.

7.3.1 Results

- Although the analytical node model was developed to analyze the performance of a single node, it also provides an accurate approximation of the mean system performance of a random walk search algorithm in large-scale wireless sensor networks.

- Decreasing node transmission range increases the total rate of transmissions in the network. This was attributed to increased query traffic as a consequence of decreased spatial uniformity in the distribution of informed nodes. However, as long as the network remains connected, the resulting increase in energy expenditure as a consequence of higher transmission rates is outweighed by the lower energy costs per transmission.
- Decreasing the mean lifetime of a query only marginally decreases the mean total arrival rate (and, hence, has little effect on total energy expenditure), but increases the proportion of query failures compared to queries with longer lifetimes. To compensate, TTL values must be increased.

7.3.2 Contributions

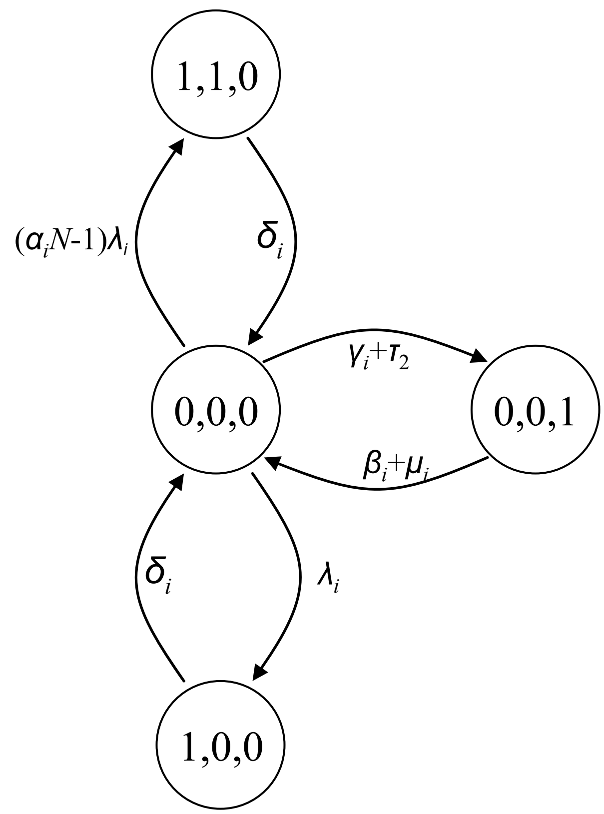
- This research demonstrated the ability of the analytical queueing model to predict search algorithm performance in large-scale wireless sensor networks. It was also the first to characterize and optimize the mean network-wide performance of a random walk search algorithm with agent and query timing constraints.
- The effect of node transmission range on network energy expenditure, transmission rates, and the proportion of query failures was identified.
- The relationship between agent/query deadlines and total expected network energy expenditure was established.

7.4 Future Research

The work detailed in this dissertation suggests several areas for subsequent research. Potential research topics listed below are based on enhancements to the existing research and/or extensions of the research into related focus areas.

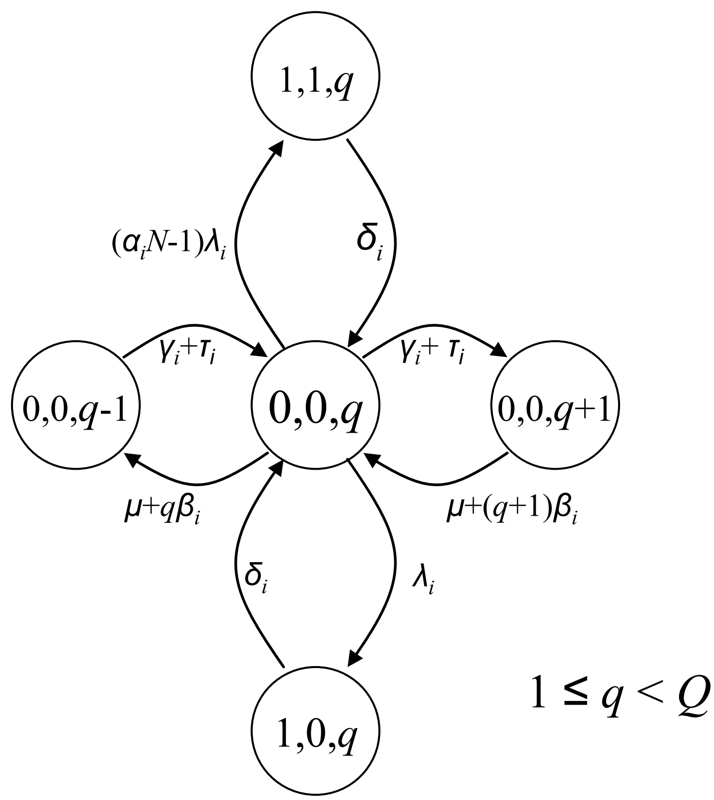
- Determine the effects of various deployment area shapes and different routing trajectories, such as curves, on TSBQ performance.
- Improve the TSBQ analytical model through explicit inclusion of the energy expended by specific MAC protocols in direct support of the search function. The current model assumes MAC energy expenditure is constant over the range of node densities; however, MAC energy expenditure may change as a consequence of node density.
- Extend the TSBQ analytical model by incorporating the effects of variable node transmission power and range. This permits determination of the optimum combination of node transmission range, the proportion of informed nodes, and the number of designated receivers per query transmission.
- Examine the effects of node mobility on TSBQ search protocol performance.
- Evaluate the effects of lifetime-limited resources and time-constrained queries on the optimum proportion of informed nodes in the TSBQ search protocol.
- Improve the analytical node model of Chapter 5 to include the effects of agent time limitations on the proportion of nodes that can be informed by an agent.
- Integrate node mobility into the network simulations of Chapter 6 and evaluate its effects on the total energy expenditure of a random walk search algorithm.

Appendix. Node Model State Diagrams



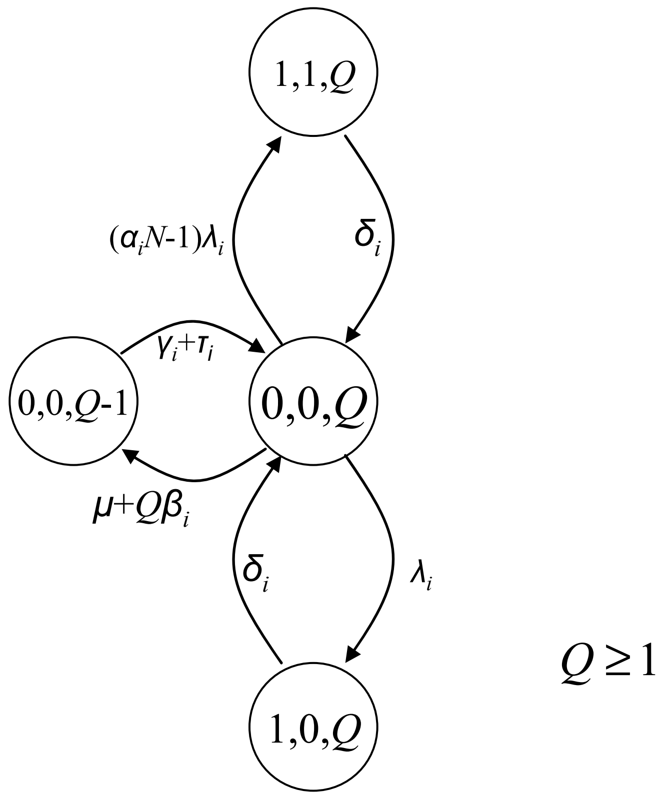
$$\left[\alpha_i N \lambda_i + \gamma_i + \tau_i\right] p_{0,0,0} = \delta_i p_{1,1,0} + (\beta_i + \mu) p_{0,0,1} + \delta_i p_{1,0,0}$$

Figure 34: Node state diagram, state (0,0,0).



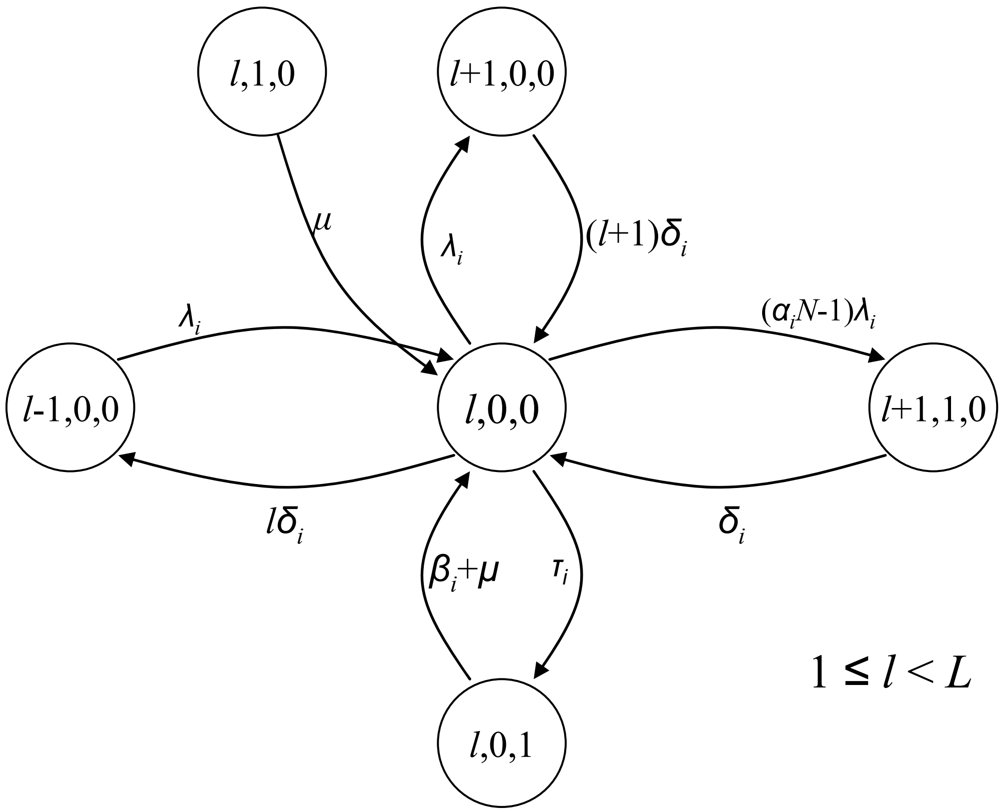
$$\left[\alpha_i N \lambda_i + \gamma_i + \tau_i + \mu + k\beta_i\right] p_{0,0,q} =$$
$$(\gamma_i + \tau_i) p_{0,0,q-1} + \left[\mu + (q+1)\beta_i\right] p_{0,0,q+1} + \delta_i p_{1,1,q} + \delta_i p_{1,0,q}$$

Figure 35: Node state diagram, state (0,0,q), $1 \le q < Q$.



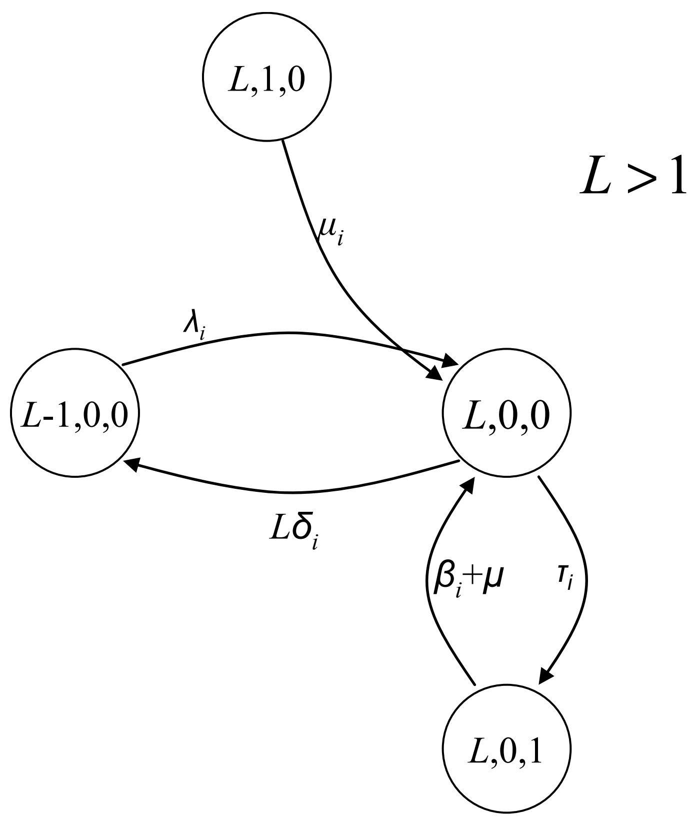
$$\left[\alpha_i N \lambda_i + \mu + Q\beta_i\right] p_{0,0,Q} =$$
$$(\gamma_i + \tau_i) p_{0,0,Q-1} + \delta_i p_{1,1,Q} + \delta_i p_{1,0,Q}$$

Figure 36: Node state diagram, state (0,0,*Q*), $Q \geq 1$.



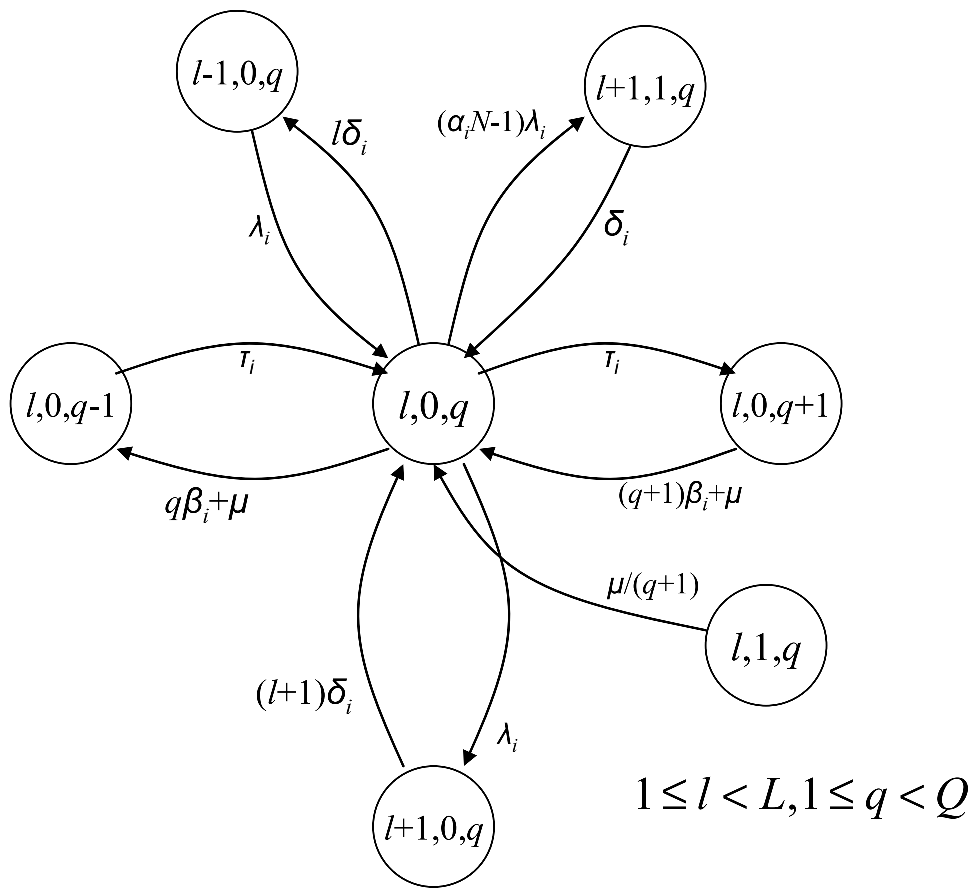
$$\left(\alpha_i N \lambda_i + \tau_i + l\delta_i\right) p_{l,0,0} = \delta_i p_{l+1,1,0} + \left[(l+1)\delta_i\right] p_{l+1,0,0}$$
$$+(\beta_i + \mu) p_{l,0,1} + \mu p_{l,1,0} + \lambda_i p_{l-1,0,0}$$

Figure 37: Node state diagram, state (l,0,0), $1 \le l < L$.



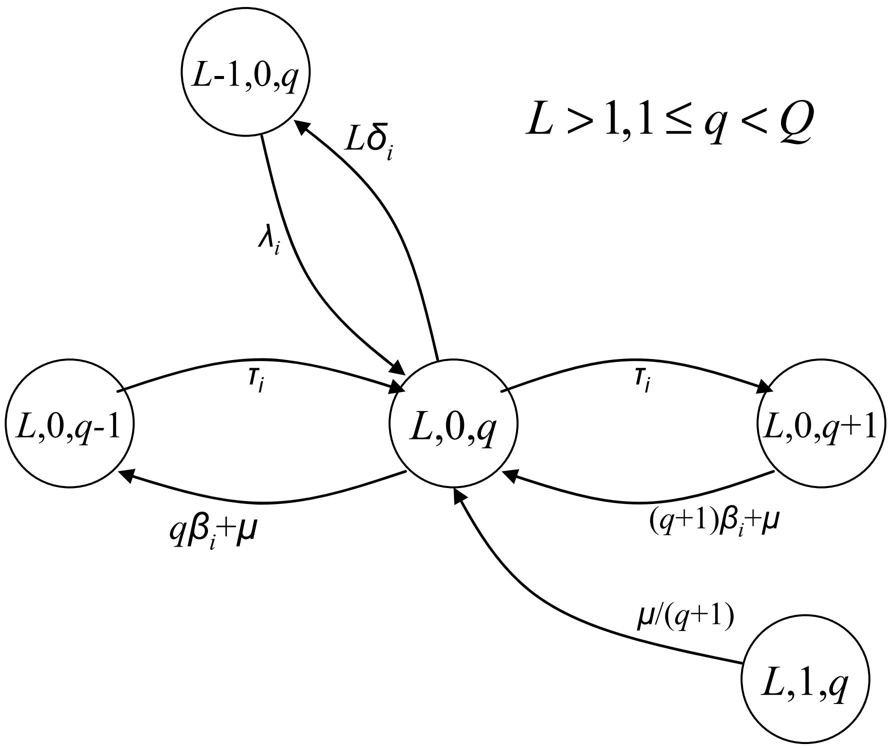
$$\left(\tau_i + L\delta_i\right) p_{L,0,0} = (\beta_i + \mu) p_{L,0,1} + \mu p_{L,1,0} + \lambda_i p_{L-1,0,0}$$

Figure 38: Node state diagram, state (L,0,0), $L > 1$.



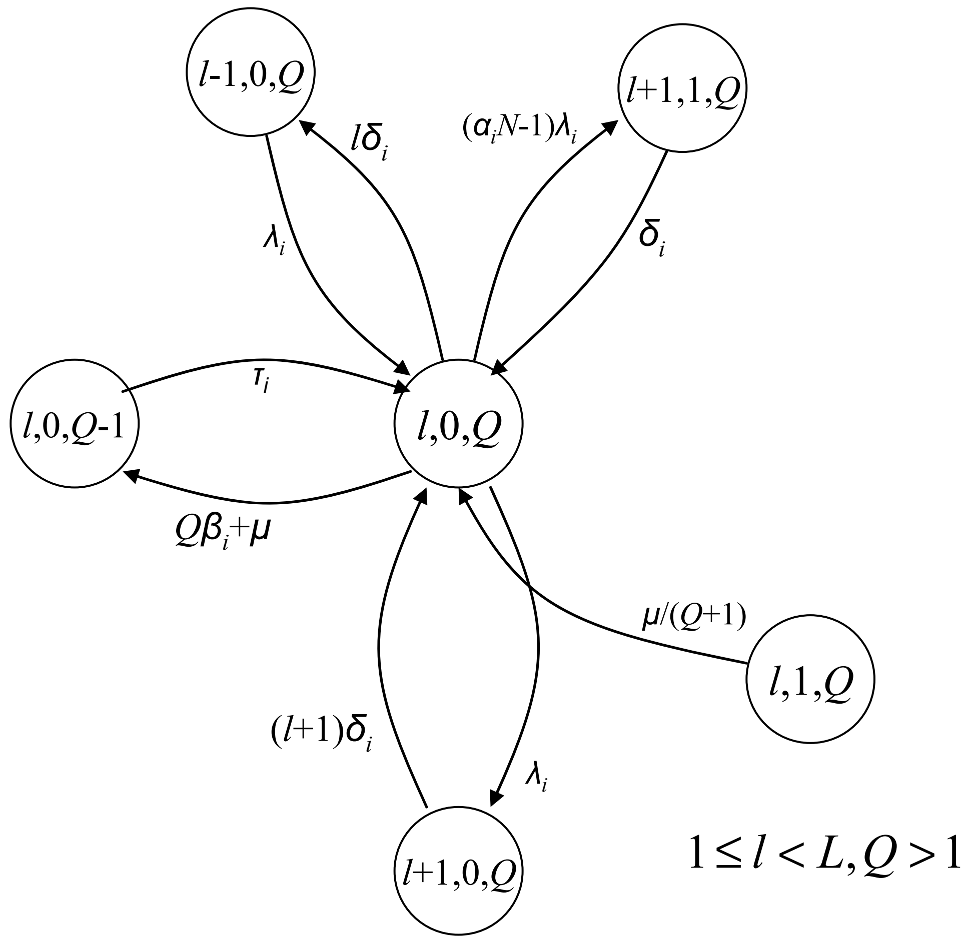
$$(l\delta_i + \alpha_i N\lambda_i + \tau_i + q\beta_i + \mu)p_{l,0,q} = \left[(q+1)\beta_i + \mu\right]p_{l,0,q+1} + (l+1)\delta_i p_{l+1,0,q} + \tau_i p_{l,0,q-1}$$
$$+\delta_i p_{l+1,1,q} + \left[\mu/(q+1)\right]p_{l,1,q} + \lambda_i p_{l-1,0,q}$$

Figure 39: Node state diagram, state (l,0,q), $1 \le l < L, 1 \le q < Q$.



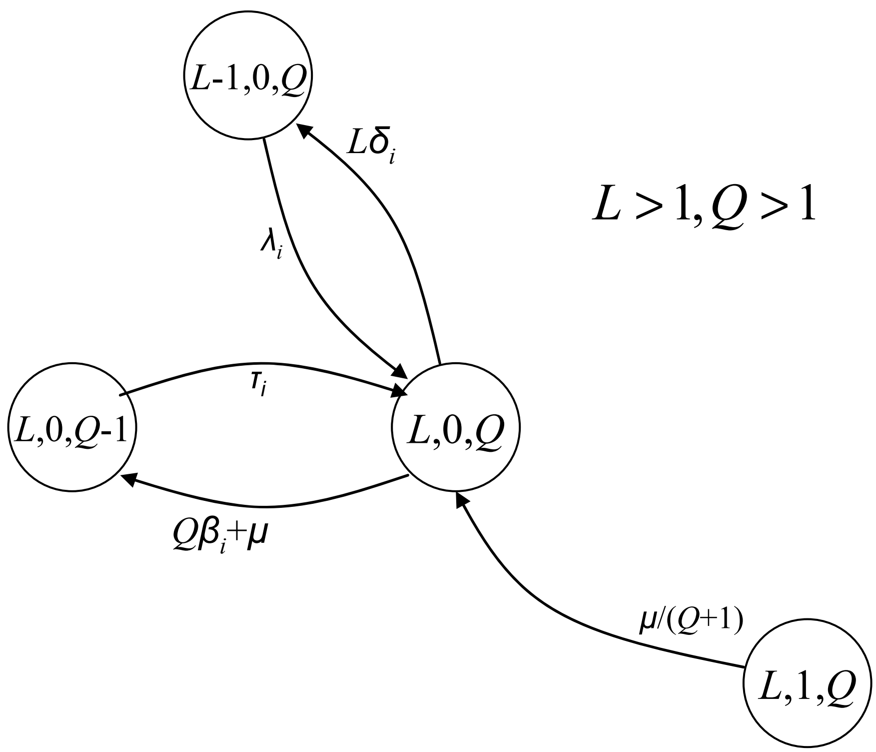
$$(L\delta_i + \tau_i + q\beta_i + \mu)\, p_{L,0,q} = \left[(q+1)\beta_i + \mu\right] p_{L,0,q+1} + \tau_i\, p_{L,0,q-1}$$
$$+ \left[\mu/(q+1)\right] p_{L,1,q} + \lambda_i\, p_{L-1,0,q}$$

Figure 40: Node state diagram, state (L,0,q), $L > 1, 1 \le q < Q$.



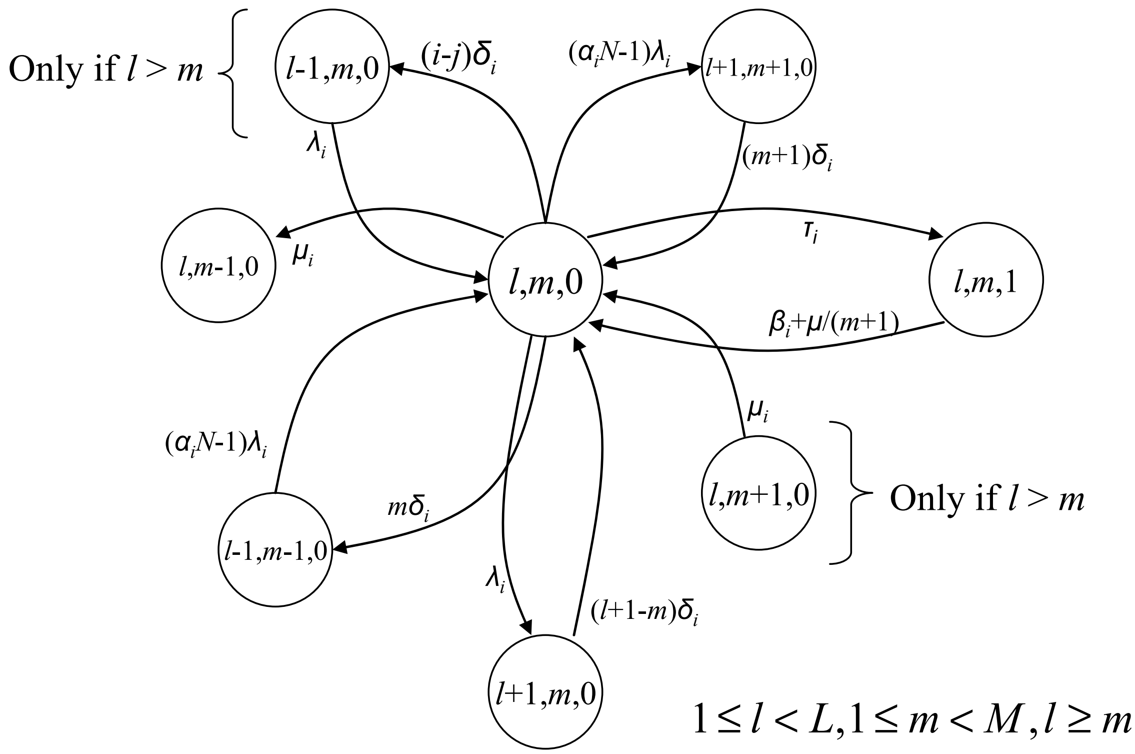
$$(l\delta_i + \alpha_i N\lambda_i + Q\beta_i + \mu)\, p_{l,0,Q} = (l+1)\delta_i\, p_{l+1,0,Q} + \tau_i\, p_{l,0,Q-1}$$
$$+\delta_i\, p_{l+1,1,Q} + \left[\mu/(Q+1)\right] p_{l,1,Q} + \lambda_i\, p_{l-1,0,Q}$$

Figure 41: Node state diagram, state (l,0,Q), $1 \le l < L, Q > 1$.



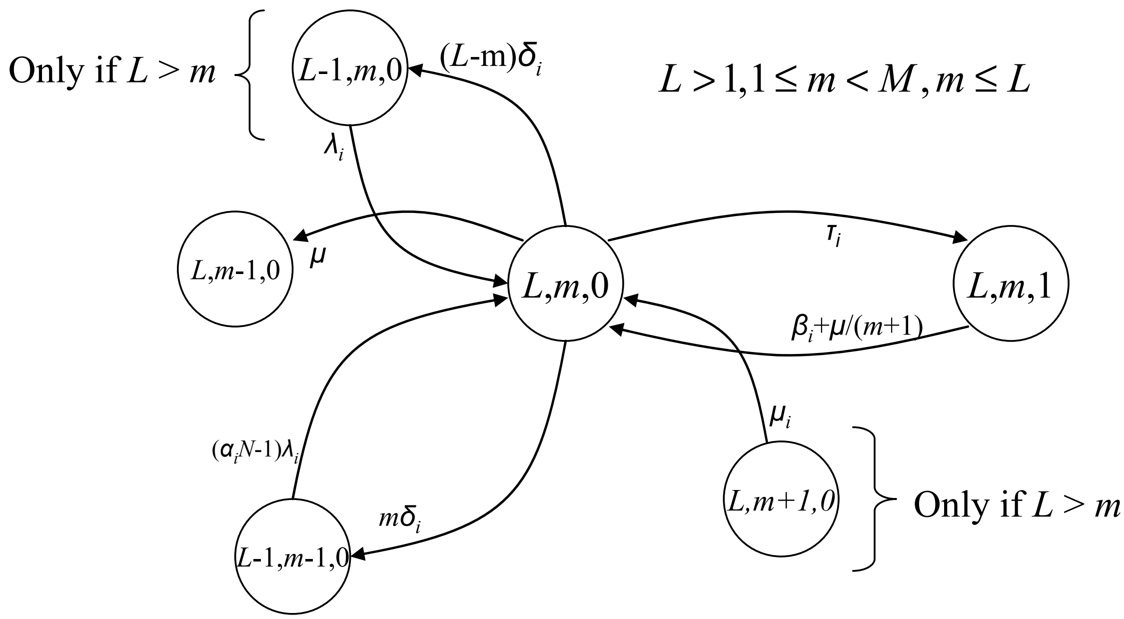
$$(L\delta_i + Q\beta_i + \mu)\, p_{L,0,Q} = \tau_i\, p_{L,0,Q-1} + \left[\mu/(Q+1)\right] p_{L,1,Q} + \lambda_i\, p_{L-1,0,Q}$$

Figure 42: Node state diagram, state (L,0,Q), $L > 1, Q > 1$.



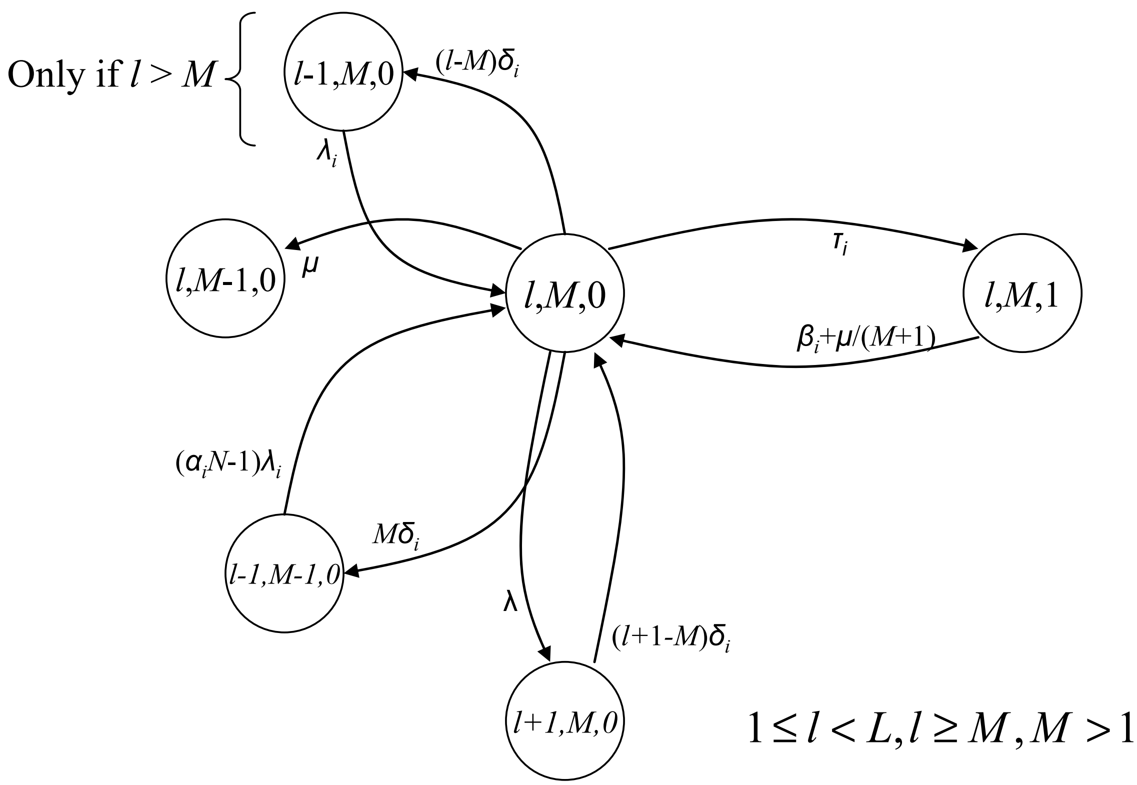
$$
\begin{aligned}
&\left[(l-m)\delta_i + \alpha_i N \lambda_i + \tau_i + m\delta_i + \mu\right] p_{l,m,0} = (m+1)\delta_i p_{l+1,m+1,0} \\
&+\left[\beta_i + \mu/(m+1)\right] p_{l,m,1} + \mu p_{l,m+1,0} 1_{l>m} + (\alpha_i N - 1)\lambda_i p_{l-1,m-1,0} \\
&+\left[(l+1-m)\delta_i\right] p_{l+1,m,0} + \lambda_i p_{l-1,m,0} 1_{l>m}
\end{aligned}
$$

Figure 43: Node state diagram, state (l,m,0), $1 \le l < L, 1 \le m < M, l \ge M$.



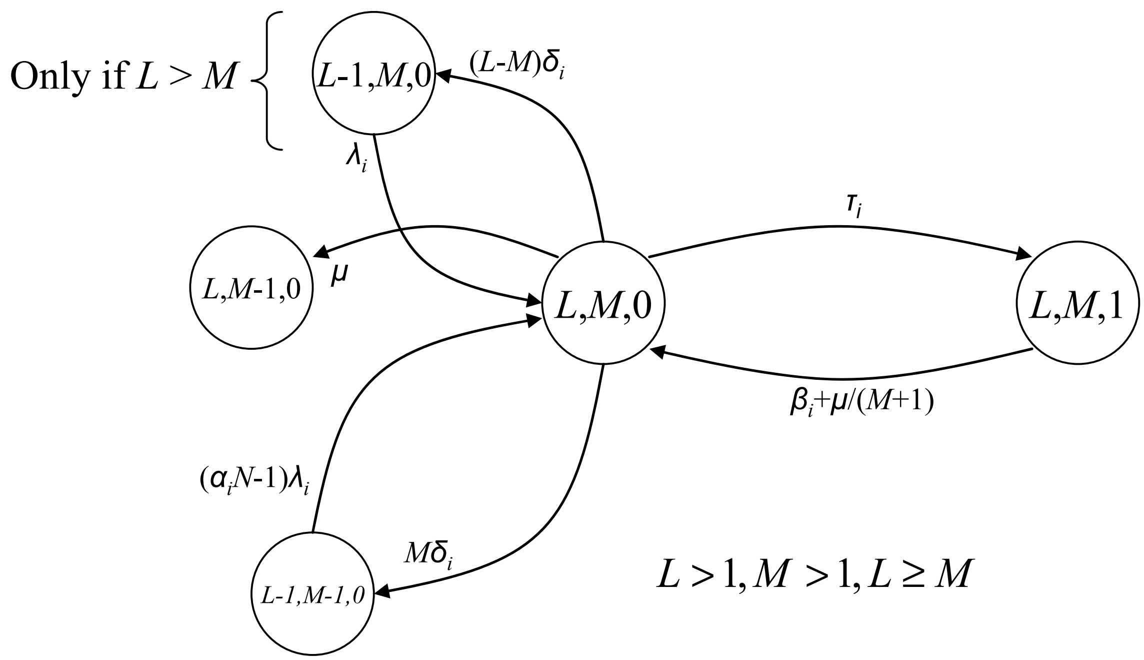
$$\left[(L-m)\delta_i + \tau_i + m\delta_i + \mu\right] p_{L,m,0} = \left[\beta_i + \mu/(m+1)\right] p_{L,m,1} + \mu p_{L,m+1,0} 1_{L>m}$$
$$+(\alpha_i N - 1)\lambda_i p_{L-1,m-1,0} + \lambda_i p_{L-1,m,0} 1_{L>m}$$

Figure 44: Node state diagram, state (L,m,0), $L > 1, 1 \le m < M, m \le L$.



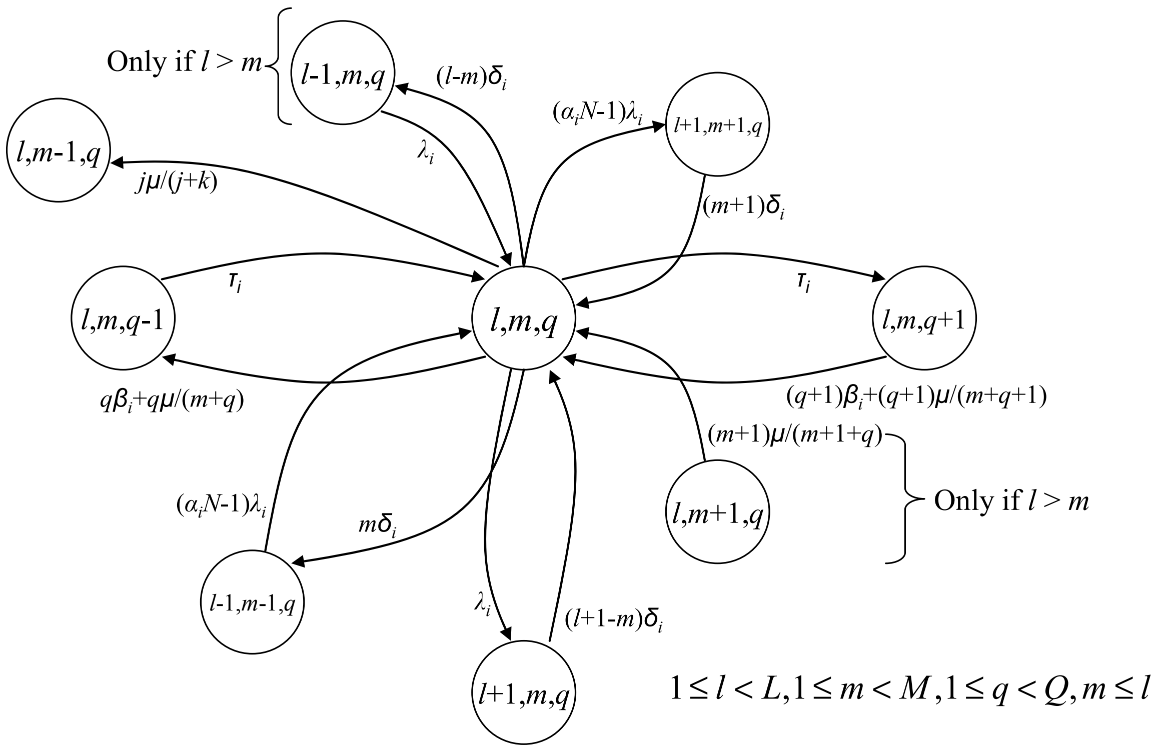
$$\left[(l-M)\delta_i+\lambda_i+\tau_i+M\delta_i+\mu\right]p_{l,M,0}=\left[\beta_i+\mu/(M+1)\right]p_{l,M,1}$$
$$+(\alpha_i N-1)\lambda_i p_{l-1,M-1,0}+\left[(l+1-M)\delta_i\right]p_{l+1,M,0}+\lambda_i p_{l-1,M,0}1_{l>M}$$

Figure 45: Node state diagram, state (l,M,0), $1\le l<L, l\le M, M>1$.



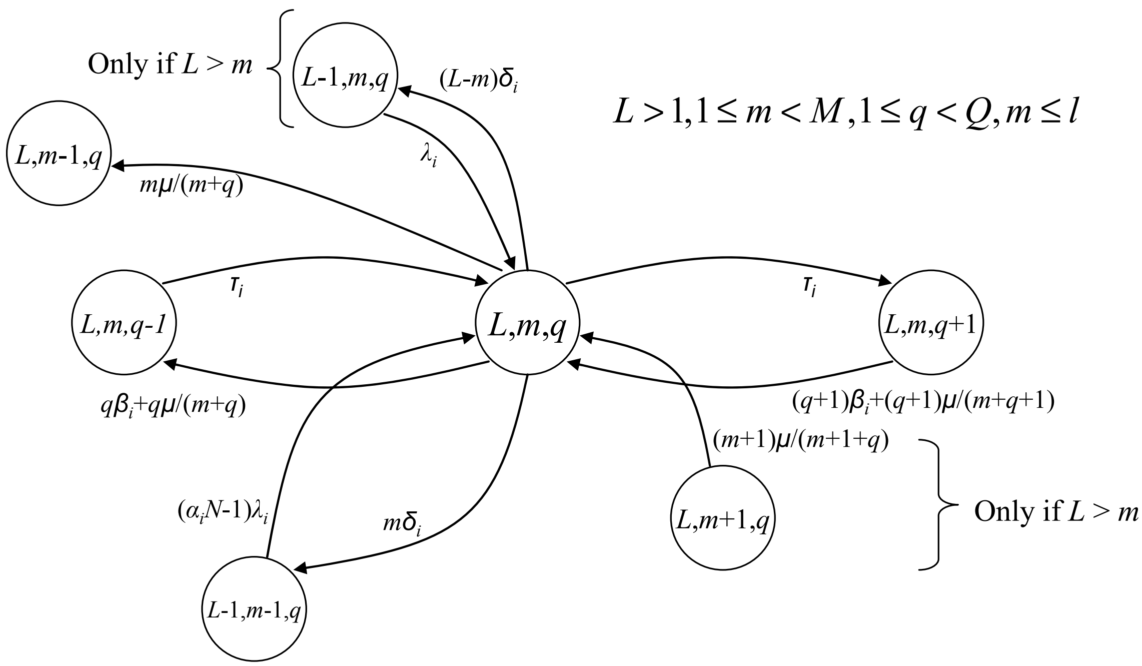
$$\left[(L-M)\delta_i+\tau_i+M\delta_i+\mu\right]p_{L,M,0}=\left[\beta_i+\mu/(M+1)\right]p_{L,M,1}$$
$$+(\alpha_i N-1)\lambda_i p_{L-1,M-1,0}+\lambda_i p_{L-1,M,0}1_{L>M}$$

Figure 46: Node state diagram, state (L,M,0), $L>1, M>1, L\geq M$.



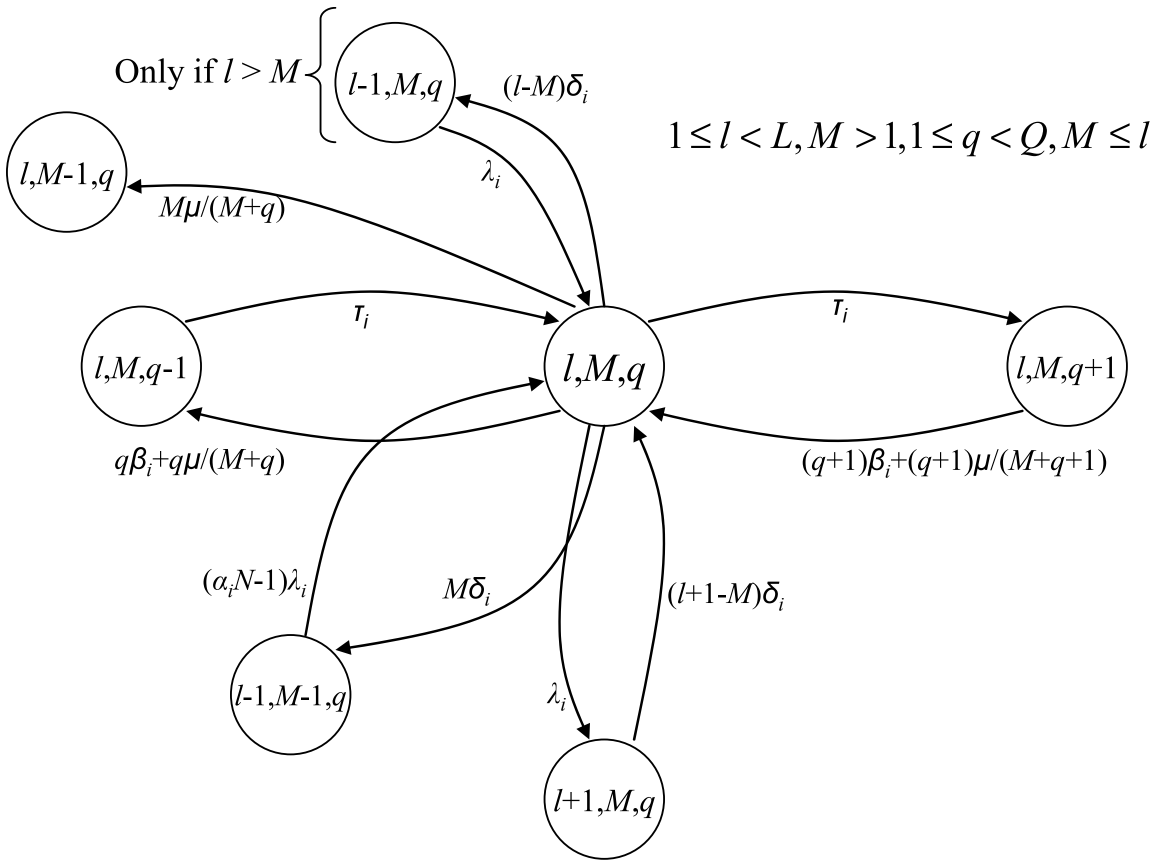
$$
\begin{aligned}
&\left[(l-m)\delta_i + \alpha_i N\lambda_i + \tau_i + m\delta_i + q\beta_i + \mu\right] p_{l,m,q} = \\
&(m+1)\delta_i p_{l+1,m+1,q} + \left[(q+1)\beta_i + (q+1)\mu/(m+q+1)\right] p_{l,m,q+1} \\
&+\left[(m+1)\mu/(m+1+q)\right] p_{l,m+1,q} 1_{l>m} + (\alpha_i N - 1)\lambda_i p_{l-1,m-1,q} + \tau_i p_{l,m,q-1} \\
&+\left[(l+1-m)\delta_i\right] p_{l+1,m,q} + \lambda_i p_{l-1,m,q} 1_{l>m}
\end{aligned}
$$

Figure 47: Node state diagram, state (l,m,q), $1 \le l < L, 1 \le m < M, 1 \le q < Q, m \le l$.



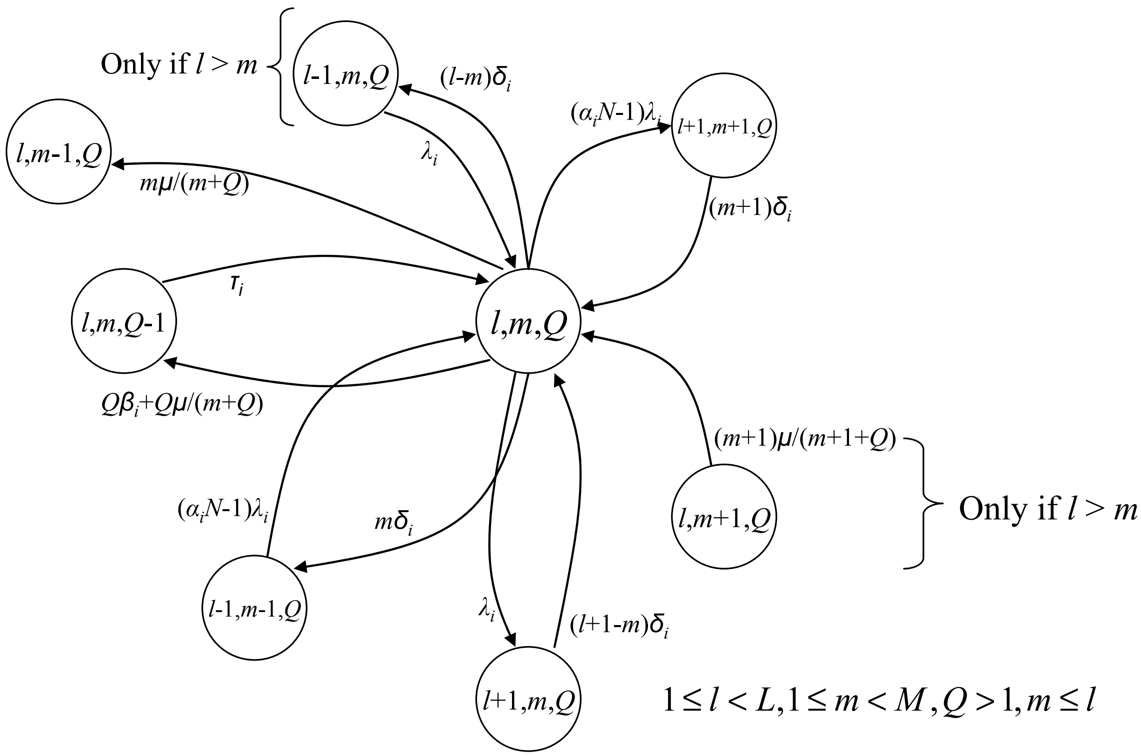
$$\left[(L-m)\delta_i+\tau_i+m\delta_i+q\beta_i+\mu\right]p_{L,m,q}=\left[(q+1)\beta_i+(q+1)\mu/(m+q+1)\right]p_{L,m,q+1}$$
$$+\left[(m+1)\mu/(m+1+q)\right]p_{L,m+1,q}1_{L>m}+(\alpha_i N-1)\lambda_i p_{L-1,m-1,q}+\tau_i p_{L,m,q-1}$$
$$+\lambda_i p_{L-1,m,q}1_{L>m}$$

Figure 48: Node state diagram, state (*L*,*m*,*q*), $L>1,1\le m<M,1\le q<Q,m\le l$.



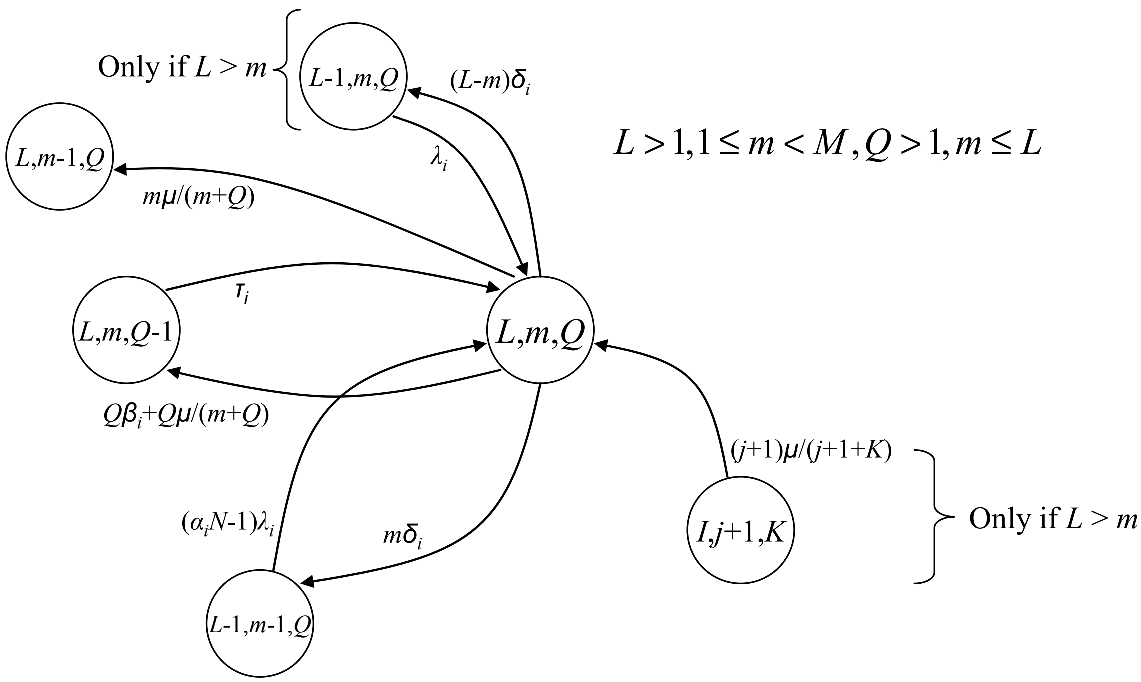
$$\left[(l-M)\delta_i+\lambda_i+\tau_i+M\delta_i+q\beta_i+\mu\right]p_{l,M,q}=\left[(q+1)\beta_i+(q+1)\mu/(M+q+1)\right]p_{l,M,q+1}$$
$$+(\alpha_i N-1)\lambda_i p_{l-1,M-1,q}+\tau_i p_{l,M,q-1}+\left[(l+1-M)\delta_i\right]p_{l+1,M,q}+\lambda_i p_{l-1,M,q}1_{l>M}$$

Figure 49: Node state diagram, state (l,M,q), $1 \le l < L, M > 1, 1 \le q < Q, M \le l$.



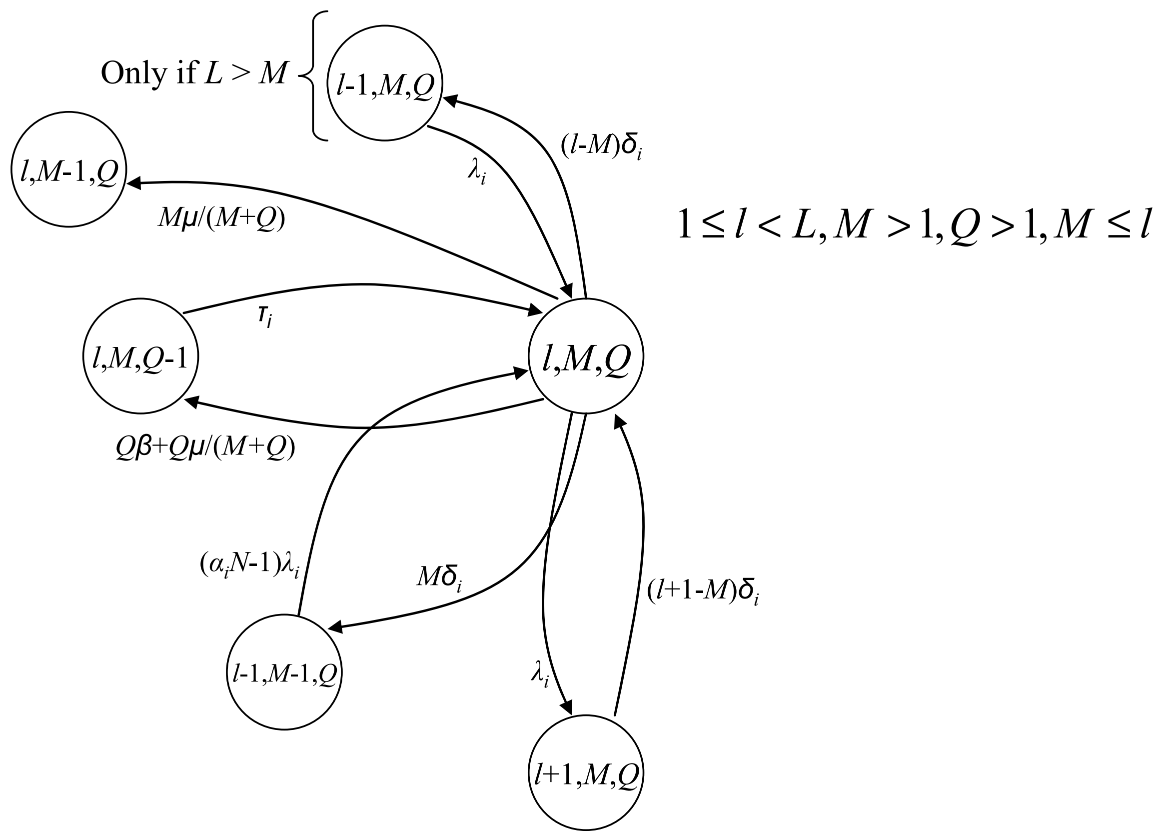
$$\left[(l-m)\delta_i + \alpha_i N\lambda_i + m\delta_i + Q\beta_i + \mu\right] p_{l,m,Q} = (m+1)\delta_i p_{l+1,m+1,Q}$$
$$+\left[(m+1)\mu/(m+1+Q)\right] p_{l,m+1,Q} 1_{l>m} + (\alpha_i N - 1)\lambda_i p_{l-1,m-1,Q}$$
$$+\tau_i p_{l,m,Q-1} + \left[(l+1-m)\delta_i\right] p_{l+1,m,Q} + \lambda_i p_{l-1,m,Q} 1_{l>m}$$

Figure 50: Node state diagram, state (l,m,Q), $1 \le l < L, 1 \le m < M, Q > 1, m \le l$.



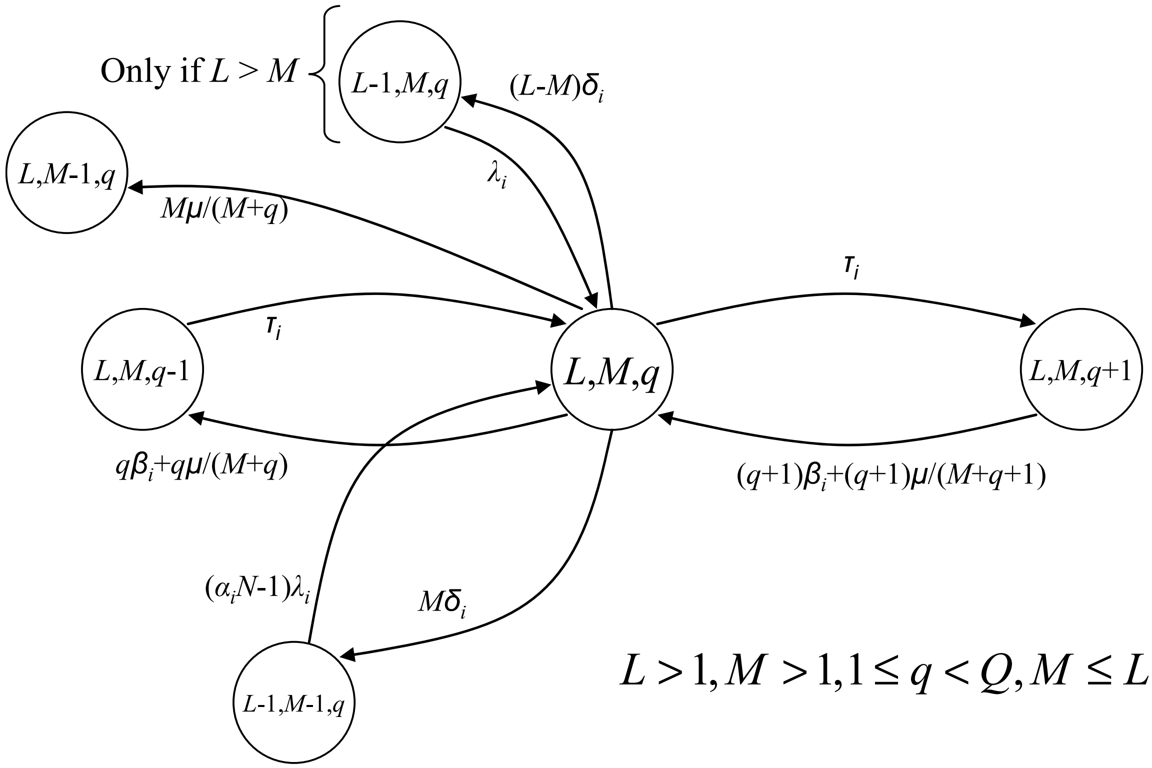
$$\left[(L-m)\delta_i + m\delta_i + Q\beta_i + \mu\right] p_{L,m,Q} = \left[(m+1)\mu/(m+1+Q)\right] p_{L,m+1,Q} 1_{L>m}$$
$$+(\alpha_i N-1)\lambda_i p_{L-1,m-1,Q} + \tau_i p_{L,m,Q-1} + \lambda_i p_{L-1,m,Q} 1_{L>m}$$

Figure 51: Node state diagram, state (L,m,Q), $L > 1, 1 \le m < M, Q > 1, m \le L$.



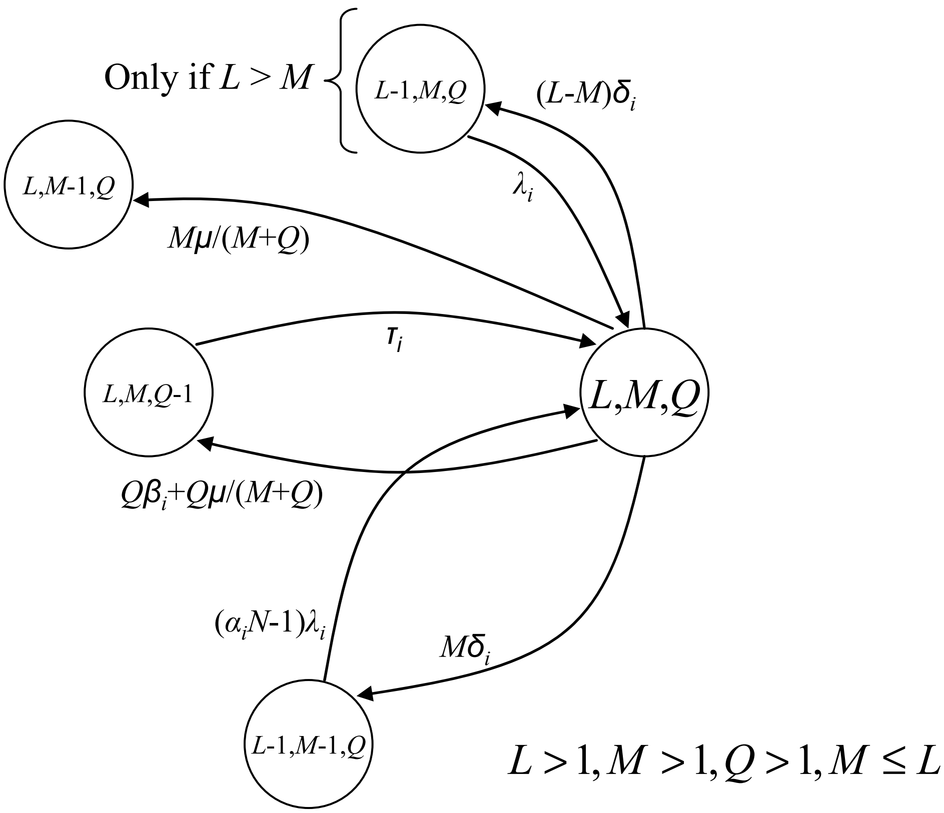
$$\left[(l-M)\delta_i+\lambda_i+M\delta_i+Q\beta_i+\mu\right]p_{l,M,Q}=(\alpha_i N-1)\lambda_i p_{l-1,M-1,Q}+\tau_i p_{l,M,Q-1}$$
$$+\left[(l+1-M)\delta_i\right]p_{l+1,M,Q}+\lambda_i p_{l-1,M,Q}1_{l>M}$$

Figure 52: Node state diagram, state (l,M,Q), $1 \le l < L, M > 1, Q > 1, M \le l$.



$$\left[(L-M)\delta_i+\tau_i+M\delta_i+q\beta_i+\mu\right]p_{L,M,q}=\left[(q+1)\beta_i+(q+1)\mu/(M+q+1)\right]p_{L,M,q+1}$$
$$+(\alpha_i N-1)\lambda_i p_{L-1,M-1,q}+\tau_i p_{L,M,q-1}+\lambda_i p_{L-1,M,q}1_{L>M}$$

Figure 53: Node state diagram, state (L,M,q), $L>1, M>1, 1\le q<Q, M\le L$.



$$\left[(L-M)\delta_i + M\delta_i + Q\beta_i + \mu\right] p_{L,M,Q} = (\alpha_i N - 1)\lambda_i p_{L-1,M-1,Q} + \tau_i p_{L,M,Q-1}$$
$$+\lambda_i p_{L-1,M,Q} 1_{L>M}$$

Figure 54: Node state diagram, state (L,M,Q), $L > 1, M > 1, Q > 1, M \le L$.

Bibliography

[AB04] C. Avin and C. Brito, "Efficient and robust query processing in dynamic environments using random walk techniques," in *Proceedings of the Third International Symposium on Information Processing in Sensor Networks*, pp. 277-286, 2004.

[AS03] J. Aspnes and G. Shah, "Skip graphs," in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms,* pp. 384-393, 2003.

[ASC02] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, pp. 102-114, 2002.

[AyS02] I. Aydin and C. C. Shen, "Facilitating match-making service in ad hoc and sensor networks using pseudo quorum," in *Proceedings of the Eleventh International Conference on Computer Communications and Networks*, pp. 4-9, 2002.

[BA05] N. Bisnik and A. Abouzeid, "Modeling and analysis of random walk search algorithms in P2P networks," in *Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems*, pp. 95-103, 2005.

[BCM05] P. Bellavista, A. Corradi and E. Magistretti, "Comparing and evaluating lightweight solutions for replica dissemination and retrieval in dense MANETs," in *Proceedings of the 10th IEEE Symposium on Computers and Communications*, pp. 43-50, 2005.

[BE02] 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*, pp. 22-31, 2002.

[BHE00] N. Bulusu, J. Heidemann and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications,* vol. 7, pp. 28-34, 2000.

[BK03] M. Barbeau and E. Kranakis, "Modeling and performance analysis of service discovery strategies in ad hoc networks," in *Proceedings of the International Conference on Wireless Networks,* 2003.

[BM02] P. Bergamo and G. Mazzini, "Localization in sensor networks with fading and mobility," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 2, pp. 750-754, 2002.

[BMJ+98] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the Annual International Conference on Mobile Computing and Networking*, pp. 85-97, 1998.

[BTJ05] T. Banka, G. Tandon and A. Jayasumana, "Zonal rumor routing for wireless sensor networks," in *Proceedings of the International Conference on Information Technology: Coding and Computing*, vol. II, pp. 562-567, 2005.

[CJB+02] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *Wireless Networks,* vol. 8, pp. 481-494, 2002.

[Cla04] R. Clauberg, "RFID and Sensor Networks," IBM Research, Zurich Research Laboratory, 2004. Available at http://www.m-lab.ch/rfid-workshop/ibm_paper.pdf . Last accessed on 16 July 2007.

[CS02] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," in *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 177-190, 2002.

[CSC05] C. F. Chou, J. J. Su and C. Y. Chen, "Straight line routing for wireless sensor networks," in *Proceedings of the 10th IEEE Symposium on Computers and Communications*, pp. 110-115, 2005.

[GBB+05] R. Gaeta, G. Balbo, S. Bruell, M. Gribaudo and M. Sereno, "A simple analytical framework to analyze search strategies in large-scale peer-to-peer networks," *Performance Evaluation,* vol. 62, pp. 1-16, 2005.

[GMS05] C. Gkantsidis, M. Mihail and A. Saberi, "Hybrid search schemes for unstructured peer-to-peer networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1526-1537, 2005.

[GZR01] C. Guo, L. C. Zhong and J. M. Rabaey, "Low power distributed MAC for ad hoc sensor radio networks," in *Proceedings of the 2001 IEEE Global Telecommunications Conference (GLOBECOM '01)*, pp. 2944-2948, 2001.

[HBE+01] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K. Lan, Y. Xu, W. Ye, D. Estrin and R. Govindan, "Effects of detail in wireless network simulation," in *Proceedings of the SCS Multiconference on Distributed Simulation*, pp. 3-11, 2001.

[IB05] S. S. Iyengar and R. R. Brooks, *Distributed Sensor Networks.* Boca Raton FL: Chapman & Hall/CRC, pp. 1123, 2005.

[IGE00] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceedings of the Annual International Conference on Mobile Computing and Networking*, pp. 56-67, 2000.

[Jai91] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.* Littleton MA: John Wiley & Sons, Inc., 1991.

[JBR+07] G. J. Jordt, R. O. Baldwin, J. F. Raquet and B. E. Mullins, "Energy Cost and Error Performance of Range-Aware, Anchor-Free Localization Algorithms," accepted for publication in *Ad Hoc Networks,* 2007.

[JM96] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *VLSI, Computer Architecture and Digital Signal Processing,* pp. 153, 1996.

[JW05] S. Jin and L. Wang, "Content and service replication strategies in multi-hop wireless mesh networks," in *Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 79-86, 2005.

[KA05] B. Krishnamachari and J. Ahn, "Optimizing data replication for expanding ring-based queries in wireless sensor networks," *USC Computer Engineering Technical Report CENG-05-14*, October 2005.

[KaK06] S. Kapadia and B. Krishnamachari, "Comparative analysis of push-pull query strategies for wireless sensor networks," in *Proceedings of the International Conference on Distributed Computing in Sensor Systems (DCOSS),* 2006.

[KH04] B. Krishnamachari and J. Heidemann, "Application-specific modeling of information routing in wireless sensor networks," in *Proceedings of the 2004 IEEE International Conference on Performance, Computing, and Communications*, pp. 717-722, 2004.

[KK00] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the Annual International Conference on Mobile Computing and Networking*, pp. 243-254, 2000.

[Kle75] L. Kleinrock, *Queueing Systems, Volume I: Theory*, A Wiley-Interscience Publication, 1975.

[LB04] H. Luo and M. Barbeau, "Performance evaluation of service discovery strategies in ad hoc networks," in *Proceedings of the Second Annual Conference on Communication Networks and Services Research*, pp. 61-68, 2004.

[LCC+02] Q. Lv, P. Cao, E. Cohen, K. Li and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceedings of the 16th International Conference on Supercomputing,* pp. 84-95, 2002.

[LHJ06] D. Liu, X. Hu and X. Jia, "Energy Efficient Protocols for Information Dissemination in Wireless Sensor Networks," *Lecture Notes in Computer Science,* vol. 3842, pp. 176-185, 2006.

[LHZ04] X. Liu, Q. Huang and Y. Zhang, "Combs, needles, haystacks: Balancing push and pull for discovery in large-scale sensor networks," in *Proceedings of the International Conference on Embedded Networked Sensor Systems*, pp. 122-133, 2004.

[LKR04] G. Lu, B. Krishnamachari and C. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks," in *Proceedings of the 18th International Parallel and Distributed Processing Symposium*, pp. 224, 2004.

[MKB05] R. Melamed, I. Keidar and Y. Barel, "Octopus: A fault-tolerant and efficient ad-hoc routing protocol," in *Proceedings of the 24th IEEE Symposium on Reliable Distributed Systems*, pp. 39-49, 2005.

[MNR02] D. Malkhi, M. Naor and D. Ratajczak, "Viceroy: A scalable and dynamic emulation of the butterfly," in *Proceedings of the Twenty-First Annual Symposium on Principles of Distributed Computing*, pp. 183-192, 2002.

[MNW04] G. S. Manku, M. Naor and U. Wieder, "Know thy neighbor's neighbor: The power of lookahead in randomized P2P networks," in *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pp. 54-63, 2004.

[MSK+01] S. Meguerdichian, S. Slijepcevic, V. Karayan and M. Potkonjak, "Localized algorithms in wireless ad-hoc networks: Location discovery and sensor exposure," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 106-116, 2001.

[NN03] B. Nath and D. Niculescu, "Routing on a curve," *ACM SIGCOMM Computer Communication Review,* vol. 33, pp. 155-160, 2003.

[NSC03] P. Nuggehalli, V. Srinivasan and C. F. Chiasserini, "Energy-efficient caching strategies in ad hoc wireless networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing,* pp. 25-34, 2003.

[OK04] F. Ordonez and B. Krishnamachari, "Optimal information extraction in energy-limited wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 1121-1129, 2004.

[PAK+05] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine,* vol. 22, pp. 54-69, 2005.

[PTL+05] J. Park, Y. Tscha, K. H. Lee and M. U. Çağlayan, "Query slipping prevention for trajectory-based matchmaking service in wireless sensor networks," in *Proceedings of the 2005 IEEE International Conference on Communications*, pp. 3050-3056, 2005.

[Rap96] T. S. Rappaport, *Wireless Communications: Principles and Practice.* Piscataway NJ: IEEE Press, 1996.

[RD01] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware),* pp. 329–350, 2001.

[RFH+01] S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Schenker, *A Scalable Content-Addressable Network.* New York NY: ACM Press, 2001.

[RKS+03] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin and F. Yu, "Data-Centric Storage in Sensornets with GHT, A Geographic Hash Table," *Mobile Networks and Applications,* vol. 8, pp. 427-442, 2003.

[RKY+02] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan and S. Shenker, "GHT: A geographic hash table for data-centric storage," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 78-87, 2002.

[ROG06] V. Rajendran, K. Obraczka and J. J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," *Wireless Networks,* vol. 12, pp. 63-78, 2006.

[Sha04] S. Shakkottai, "Asymptotics of query strategies over a sensor network," in *Proceedings of the IEEE INFOCOM*, March 2004.

[SKH03] N. Sadagopan, B. Krishnamachari and A. Helmy, "The ACQUIRE mechanism for efficient querying in sensor networks," in *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 149-155, 2003.

[SL01] I. Stojmenovic and X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," *IEEE Transactions on Parallel and Distributed Systems,* vol. 12, pp. 1023-1032, 2001.

[SMK+01] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the 2001 SIGCOMM Conference,* vol. 31, pp. 149-160, 2001.

[SRB01] C. Savarese, J. Rabaey and J. Beutel, "Locationing in distributed ad-hoc wireless sensor networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2037-2040, 2001.

[SRK+03] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan and D. Estrin, "Data-Centric Storage in Sensornets," *Computer Communication Review,* vol. 33, pp. 137-142, 2003.

[Sto99] I. Stojmenovic, "A scalable quorum based location update scheme for routing in ad hoc wireless networks," *TR 99-09, SITE, University of Ottawa,* 1999.

[TAH02] S. Tilak, N. B. Abu-Ghazaleh and W. Heinzelman, "Infrastructure tradeoffs for sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 49-58, 2002.

[TV04] J. B. Tchakarov and N. H. Vaidya, "Efficient content location in wireless ad hoc networks," in *Proceedings of the 2004 IEEE International Conference on Mobile Data Management*, pp. 74-85, 2004.

[TYD+04] N. Trigoni, Y. Yao, A. Demers, J. Gehrke and R. Rajaraman, "Hybrid push-pull query processing for sensor networks," in *Workshop on Sensor Networks as Part of the GI-Conference Informatik*, 2004.

[VL03] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 171-180, 2003.

[UCB06] "Wireless Sensor Nodes." http://webs.cs.berkeley.edu/800demo/dots.jpg. 25 April 2006.

[WTC03] A. Woo, T. Tong and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *First International Conference on Embedded Networked Sensor Systems*, pp. 14-27, 2003.

[XHE01] Y. Xu, J. Heidemann and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 70-84, 2001.

[XK06] F. Xue and P. R. Kumar, "On the θ-coverage and connectivity of large random networks," *IEEE/ACM Transactions on Networking (TON),* vol. 14, pp. 2289-2299, 2006.

[YHE02] W. Ye, J. Heidemann and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies,* , pp. 1567-1576, 2002.

[ZH04] H. Zhang and J. Hou, "On deriving the upper bound of α-lifetime for large sensor networks," in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 121-132, 2004.

[Zig06] ZigBee Alliance, "Zigbee Specification, Tech. Rep. 053474r13." Available online at http://www.zigbee.org. Last accessed 16 July 2007. Dec 2006.

[ZKJ01] B. Y. Zhao, J. Kubiatowicz and A. D. Joseph, "Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing," *Computer Science Division, University of California, Berkley, Tech. Rep. UCB/CSD-01-1141,* 2001.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 074-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
September 2007	Doctoral Dissertation	Aug 2004 – Sep 2007

4. TITLE AND SUBTITLE	
Energy-efficient Querying of Wireless Sensor Networks	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	**5d. PROJECT NUMBER** 06-114
Christopher R. Mann, Lt Col, USAF	**5e. TASK NUMBER**
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-7765	AFIT/DS/ENG/07-19
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/IFSC Attn: Mr. William Koenig 2241 Avionics WPAFB, OH 45433 DSN 785-4709 x3172	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Due to the distributed nature of information collection in wireless sensor networks and the inherent limitations of the component devices, the ability to store, locate, and retrieve data and services with minimum energy expenditure is a critical network function. Additionally, effective search protocols must scale efficiently and consume a minimum of network energy and memory reserves.

A novel search protocol, the Trajectory-based Selective Broadcast Query protocol, is proposed. An analytical model of the protocol is derived, and an optimization model is formulated. Based on the results of analysis and simulation, the protocol is shown to reduce the expected total network energy expenditure by 45.5 percent to 75 percent compared to current methods.

This research also derives an enhanced analytical node model of random walk search protocols for networks with limited-lifetime resources and time-constrained queries. An optimization program is developed to minimize the expected total energy expenditure while simultaneously ensuring the proportion of failed queries does not exceed a specified threshold.

Finally, the ability of the analytical node model to predict the performance of random walk search protocols in large-population networks is established through extensive simulation experiments. It is shown that the model provides a reliable estimate of optimum search algorithm parameters.

15. SUBJECT TERMS

Search algorithm, wireless sensor network, queueing theory, optimization

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**Rusty O. Baldwin, PhD (ENG)**
U	U	U	UU	212	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 4445 (rusty.baldwin@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18