# A Queueing Approach to Optimal Resource Replication in Wireless Sensor Networks

Christopher R. Mann [a] Rusty O. Baldwin [a]
Jeffrey P. Kharoufeh [b] Barry E. Mullins [a]

[a] *Dept of Electrical and Computer Engineering, Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson Air Force Base, OH 45433, USA*

[b] *Dept of Industrial Engineering, 1048 Benedum Hall, University of Pittsburgh, Pittsburgh, PA 15261, USA*

**Abstract**

We develop a queueing model for analyzing resource replication strategies in wireless sensor networks. 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 predetermined 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. While insufficient resource replication increases query failures and transmission rates, 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.

*Key words:* resource replication, search algorithms, wireless sensor networks

## 1 Introduction

Wireless sensor networks (WSN) are composed of a large number of sensing devices, called nodes, which are linked via a wireless transmission medium. These nodes are characterized by limited energy stores, local storage capacity,

*Email addresses:* christopher.mann@afit.edu (Christopher R. Mann), rusty.baldwin@afit.edu (Rusty O. Baldwin), jkharouf@pitt.edu (Jeffrey P. Kharoufeh), barry.mullins@afit.edu (Barry E. Mullins).

and computational ability. Additionally, the transmission range of individual nodes is typically much smaller than the span of the network, requiring cooperation between nodes to achieve network-wide communication.

Due to the limited capabilities of individual nodes and the distributed nature of data collection and computation in wireless sensor networks, it is unlikely that any single node will have local access to all resources (i.e., a specific service or information) needed to complete its assigned tasks. When a particular resource is unavailable locally, nodes are forced to locate the resource within the network. Such requests cause the network to expend precious energy reserves, reducing the useful lifetime of the network. Additionally, the utility of the collected information will decay in time-sensitive applications. Consequently, the useful lifetime of a resource previously located by a node is finite, requiring new searches to be initiated when the results obtained from past searches become stale.

Effective WSN search algorithms must strike a balance between guaranteeing a specified proportion of successful searches and minimizing the total energy expended by the network to locate the desired resource. Additionally, search algorithms must meet any deadlines imposed by the requesting node and should not return stale data. For example, consider a WSN that must detect intrusions within an area monitored by the network. This application has inherent time limitations on both the discovery and reporting of the collected data. In military applications, the node designated as the sink may change over time. An unmanned aerial vehicle (UAV) flying over a network may be within range of only a portion of the network at any given time. Thus, all nodes will need to both advertise and request information within the network to meet the requirements of the application.

It is well known that the energy expended to locate a resource can be reduced (compared to flooding techniques) if the availability of the resource is advertised to a subset of the network's nodes; this is the approach used in the original rumor routing protocol [7]. Rumor routing is a kind of "blind" search algorithm in which nodes have no prior knowledge of the location of the desired resource. Instead, nodes advertise the availability of a resource via a specialized packet that is routed from node-to-node via a random walk. Similarly, a node seeking a particular resource transmits its request to a randomly-chosen neighboring node. If this neighbor cannot answer the request, the request is propagated to another randomly-chosen neighbor. This process continues until either the request is answered or it expires.

In a manner similar to rumor routing, several search protocols, including those described in [3,5,8,19,24], conserve energy through resource advertising. In [19], a mathematical model of the total energy expended to advertise and locate a WSN resource is derived for a rumor routing variant based on network

size, node power consumption characteristics, resource popularity, and node density. Using this model, the resource replication profile that results in the minimum total expected energy expended by the network can be determined.

Although the mechanisms for advertising and locating resources are well-understood, none of these search protocols consider quality of service (QoS) issues such as query deadlines, the proportion of query failures, or the effect of limited resource lifetimes. Additionally, we found no mention in the literature of the effect of resource advertising on the intensity of network query traffic. Nodes aware of the availability 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 developing a node model of search algorithm behavior that minimizes total network transmissions while meeting specified QoS constraints.

Our work makes four contributions to the query-based WSN domain. First, we develop an analytical queueing model of WSN nodes 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 [7] 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, we determine the resource replication level that minimizes the total traffic intensity while ensuring a specified threshold on the proportion of query failures is not exceeded. Third, we explain the effects of various network parameters on search algorithm performance and show 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, we use simulation experiments to examine the effects of alternative agent/query lead time distributions on our metrics.

The remainder of this paper is organized as follows. Section 2 discusses the literature pertinent to our work. In Section 3, we develop a mathematical model of a WSN node's event table and transmission queue. We characterize the behavior of the system using a Markov chain and solve the resulting balance equations to determine the steady-state populations of the event table and transmission queue. In Section 4, we show how to solve discrete optimization problems to determine the optimal resource replication level by minimizing the total node transmission rate while satisfying query failure constraints. In Section 5, we present the results of simulations using alternative agent/query expiration time distributions. Section 6 provides our concluding remarks.

## 2 Related Work

Several variants of the original rumor routing algorithm have been proposed [3,5,8,24] but, to the best of our knowledge, there has been no work related to analytically determining the appropriate number of resource replicates for random-walk search algorithms when agents and queries have finite lifetimes. In [15], the optimal replication level for expanding-ring search algorithms was derived; however, this model did not incorporate the effects of limited agent and query lifetimes. Additionally, the developers of REDMAN [5] recognized the importance of managing resource replication levels, but they relied upon the results of simulation to determine the appropriate settings.

In the related field of unstructured peer-to-peer networks, there is an extensive body of literature focused on determining the appropriate replication level for a particular resource [6,9,10,11,18]. In general, this research determines the appropriate number of resource copies to be stored by the network based on each resource's relative popularity. However, nodes in peer-to-peer networks generally have much greater computational capability, storage capacity, and energy reserves than their WSN counterparts, and no time limits are placed on a resource's availability or its corresponding request(s). Hence, these efforts are focused primarily on reducing the latency associated with locating a particular resource rather than reducing total energy expenditure or query failures.

We have also found no attempt to ensure the resulting proportion of query failures does not exceed an application-specific threshold. However, this is a critical research area for energy-constrained WSNs. While limited agent lifetimes facilitate management of each node's finite local storage capacity and ensure stale data is not returned to the end-user, limited query lifetimes guarantee precious energy reserves are not expended to locate information the requesting application can no longer use.

## 3 Node Model

We assume 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. A node also produces a resource when it offers a particular service to the network. In addition to data gathering, nodes are also required to execute specific applications in support of the network's goals. When a node requires access to a resource that is not available locally, the node is forced to poll the network to locate the necessary information and/or services.

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. The agent's purpose is to increase the probability that the resource can be located without flooding the entire network with the request. We assume 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 least at 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 "search" 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.

Next, we develop a queueing model that captures the behavior of a node's event table and transmission queue. 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, we investigate the effects of various network parameters on the optimal agent replication level.

### *3.1 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 [22]. 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 reduces the probability of collisions.

A cost-based approach is frequently used to evaluate the efficiency of WSN search algorithms. Since packet transmission typically expends more energy than any other node activity, most cost models of search algorithms for query-response WSNs use the number of transmissions, messages, or hops as their primary performance metric (cf.[1,2,4,6,7,11,12,13,15,16,17,20,21,23,25]). 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 agents and queries.

When an agent arrives at a node, the node stores a copy of the agent in its onboard 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 transmission 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 the FIFO queueing discipline.

A node's transmission buffer can be modeled as a multi-class queue since we have 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.

In contrast to agents and queries, responses are assumed to be forwarded along the most direct route between the informed node and the node that originated the query. Therefore, the number of hops required 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, the amount of response traffic in the network is assumed to be small compared to the total number of agent and query transmissions.

### *3.2 Agent/query transmission traffic*

We now define our model parameters which are also summarized in Table 1. Denote by $R$ the 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 $\lambda_i$, $i = 1, 2, \ldots, R$. 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 $1/\delta_i$, $i = 1, 2, \ldots, R$. The total expected arrival rate of agents to a node's event table includes its local rate of agent generation, $\lambda_i$, plus a proportion of the agents received from the remaining $(N-1)$ nodes. Let $A_i$ denote the mean arrival rate of type $i$ agents to a single node given by

$$A_i = \alpha_i N \lambda_i, \quad i = 1, 2, \ldots, R. \tag{1}$$

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 mean arrival rate of agents to a node's transmission queue, $A_i^{xmt}$, is

$$A_i^{xmt} = (\alpha_i N - 1)\,\lambda_i, \quad i = 1, 2, \ldots, R. \tag{2}$$

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 it 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 $\gamma_i$. 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. (We assume nodes cannot be informed with probability 1; otherwise, a node would never need to transmit a locally-generated query. Likewise, nodes cannot be informed with probability 0 since this means a node never provides a resource or observes the phenomenon of interest.) Then the node adds locally-generated type $i$ queries to its transmission queue according to a Poisson process with rate parameter $\pi_{0,i}\gamma_i$.

A node may also 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, $\tau_i$, depends on the proportion of informed nodes in the network or

$$\tau_i = \pi_{0,i}\gamma_i\,(N-1)\left[\frac{1}{(N-1)\,(1-\pi_{0,i})}\right] = \frac{\pi_{0,i}\gamma_i}{1-\pi_{0,i}}, \quad i = 1, 2, \ldots, R. \tag{3}$$

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 transmis-

sion queue is $\tau_i$. 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, we determine the total expected arrival rate of type $i$ agents and queries,$f(\alpha_i)$, to each node, or

$$\begin{aligned} f(\alpha_i) &:= \alpha_i N\lambda_i + (\gamma_i + \tau_i)\,\pi_{0,i} + \tau_i\,(1 - \pi_{0,i}) \\ &= \alpha_i N\lambda_i + 2\pi_{0,i}\gamma_i + \gamma_i \frac{\pi_{0,i}^2}{1 - \pi_{0,i}}, \qquad i = 1, 2, \ldots, R. \end{aligned} \tag{4}$$

Now, $\pi_{0,i}$ is a function of $\alpha_i$ while $N$, $\lambda_i$, and $\gamma_i$ are parameters; therefore, our objective is to choose $\alpha_i$ such that (4) is minimized. The mathematical programming formulation is

$$\begin{aligned} \min \quad & f(\alpha_i) = \alpha_i N\lambda_i + 2\pi_{0,i}\gamma_i + \gamma_i \tfrac{\pi_{0,i}^2}{1-\pi_{0,i}} \\ \text{s.t} \quad & \alpha_i \in \{2/N, 3/N, 4/N, \ldots, \alpha_{i,\max}\}, \end{aligned} \tag{5}$$

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 $\alpha_i$ that can be supported by the transmission medium. (Flooding an agent to all network nodes has been shown to be an inefficient means for advertising a resource [7]. Therefore, we assume $\alpha_{i,\max} \ll 1$.) Consequently, (5) is a discrete optimization problem which can be solved by enumerating all possible solutions and choosing the value of $\alpha_i$, say $\alpha_i^*$, that minimizes $f(\alpha_i)$. However, before this analysis can be completed, $\pi_{0,i}$ must be cast as a function of $\alpha_i$. This is accomplished in the next subsection by modeling a node's event table as an M/M/$\infty$ queue.

### *3.3 Event table as an M/M/$\infty$ 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 event type $i$, an event table can be modeled as an M/M/$\infty$ 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

Table 1
Definition of node model parameters.

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

type $i$ agents present in the event table. We are interested in the proportion of time the event table has no corresponding agents, $\pi_{0,i}$. This quantity is equivalent to the steady-state probability that an M/M/$\infty$ queue is empty and is, therefore, given by (see [14])

$$\pi_{0,i} = e^{-\alpha_i N \lambda_i / \delta_i}, \quad i = 1, 2, \ldots, R. \tag{6}$$

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 (6) accurately reflects the proportion of time a node is uninformed. Consequently, we may rewrite (4) as

$$f(\alpha_i) = \alpha_i N \lambda_i + 2\gamma_i e^{-\alpha_i N \lambda_i / \delta_i} + \gamma_i \frac{e^{-2\alpha_i N \lambda_i / \delta_i}}{1 - e^{-\alpha_i N \lambda_i / \delta_i}}, \quad i = 1, 2, \ldots, R. \tag{7}$$

The final step is to determine the value of $\alpha_i^*$.

### *3.4 Proportion of query failures*

Although we can now minimize the total arrival rate of agents and queries to a node's transmission queue, we also need to evaluate the proportion of queries that fail to locate an informed node. 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, we must not only minimize the rate of transmissions within the network, but also 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 our model and to maintain tractability, we assume 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, we consider only one type of agent and its corresponding query(ies). Later, we expand the model 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, we define the state of a node 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 fully characterized by the set of balance equations listed in Table 2.

Table 2
Node model balance equations.

| *State* | *Condition(s)* | *Balance Equation* |
|---|---|---|
| $(0,0,0)$ | None | $[\alpha_i N\lambda_i + \gamma_i + \tau_i]\, p_{0,0,0} = \delta_i p_{1,1,0} + (\beta_i + \mu)p_{0,0,1} + \delta_i p_{1,0,0}$ |
| $(0,0,q)$ | $q \geq 1$ | $[\alpha_i N\lambda_i + \gamma_i + \tau_i + \mu + k\beta_i]\, p_{0,0,q} = (\gamma_i + \tau_i)p_{0,0,q-1}$ $+ [\mu + (n+1)\beta_i]\, p_{0,0,q+1} + \delta_i p_{1,1,q} + \delta_i p_{1,0,q}$ |
| $(l,0,0)$ | $l \geq 1$ | $(\alpha_i N\lambda_i + \tau_i + i\delta_i)\, p_{l,0,0} = \delta_i p_{l+1,1,0} + [(l+1)\delta_i]\, p_{l+1,0,0}$ $+(\beta_i + \mu)p_{l,0,1} + \mu p_{l,1,0} + \lambda_i p_{l-1,0,0}$ |
| $(l,m,0)$ | $l, m \geq 1,$ $l \geq m$ | $[(l-m)\delta_i + \alpha_i N\lambda_i + \tau_i + m\delta_i + \mu]\, p_{l,m,0} = (m+1)\delta_i p_{l+1,m+1,0}$ $+ [\beta_i + \mu/(m+1)]\, p_{l,m,1} + \mu p_{l,m+1,0} 1_{l>m} + (\alpha_i N - 1)\lambda_i p_{l-1,m-1,0}$ $+ [(l+1-m)\delta_i]\, p_{l+1,m,0} + \lambda_i p_{l-1,m,0} 1_{l>m}$ |
| $(l,0,q)$ | $l \geq 1,$ $q \geq 1$ | $(l\delta_i + \alpha_i N\lambda_i + \tau_i + q\beta_i + \mu)p_{l,0,q} = [(q+1)\beta_i + \mu]\, p_{l,0,q+1}$ $+(l+1)\delta_i p_{l+1,0,q} + \tau_i p_{l,0,q-1} + \delta_i p_{l+1,1,q} + [\mu/(q+1)]\, p_{l,1,q}$ $+\lambda_i p_{l-1,0,q}$ |
| $(l,m,q)$ | $l, m \geq 1,$ $l \geq m,$ $q \geq 1$ | $[(l-m)\delta_i + \alpha_i N\lambda_i + \tau_i + m\delta_i + q\beta_i + \mu]\, p_{l,m,q} =$ $(m+1)\delta_i p_{l+1,m+1,q} + [(q+1)\beta_i + (q+1)\mu/(m+q+1)]\, p_{l,m,q+1}$ $+ [(m+1)\mu/(m+1+q)]\, p_{l,m+1,q} 1_{l>m} + (\alpha_i N - 1)\lambda_i p_{l-1,m-1,q}$ $+\tau_i p_{l,m,q-1} + [(l+1-m)\delta_i]\, p_{l+1,m,q} + \lambda_i p_{l-1,m,q} 1_{l>m}$ |
| $(l,m,q)$ | $l < m,$ $q \geq 0$ | This state cannot occur as the number of agents in the transmission queue will never exceed the number of agents in the event table. |

The final row in Table 2 indicates a node can never have more agents in its transmission queue awaiting transmission than agents stored in its event table. 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

$$1_x = \begin{cases} 1, \text{ if condition } x \text{ holds} \\ 0, \text{ otherwise} \end{cases}. \tag{8}$$

Due to the presence of three infinite state variables, the system characterized by the balance equations in Table 2 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, 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 a blocking probability to the model, this effect can be reduced by choosing $L$ and $Q$ large.

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

$$\sum_{l=0}^{L} \sum_{m=0}^{l} \sum_{q=0}^{Q} p_{l,m,q} = 1. \tag{9}$$

To determine the steady-state proportion of time in each state, we solve the linear system $AX = B$ 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 2 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 $A^{-1}$, we may obtain $X$ by

$$X = A^{-1}B. \tag{10}$$

To compute the proportion of query failures, we need only compare the rate of query failures, $q\beta_i$, in each possible state to the total rate of query arrivals. The total proportion of type $i$ query failures, denoted by $g(\alpha_i)$, is given by

$$g(\alpha_i) = \sum_{l=0}^{L} \sum_{m=0}^{l} \sum_{q=1}^{Q} \left[ \frac{q\beta_i}{\gamma_i} p_{l,m,q} \right], \quad i = 1, 2, \ldots, R. \tag{11}$$

### *3.5 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 2. Additionally, agents and queries related to other types of resources (i.e., other than the particular resource in which we are interested) 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 $\theta$. 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 3.

## 4 Numerical Example

In this section, we use a numerical example to illustrate the determination of the optimal replication level for a specific resource based on the results of Section 3. We also discuss the tradeoffs associated with the minimum transmission strategy (the energy-centric approach) and the minimum query-failure strategy (the failure-centric approach). Finally, the effect of various parameters on replication levels is explored.

### *4.1 Example: 5000-node network*

For the purpose of analyzing the performance of a 5000-node network, we first define a variation of the optimum energy-centric replication level, $\alpha_i^*$. Let $\kappa_i$ 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 replication level required to satisfy the network's query failure requirement. Consequently, $\alpha_{\kappa_i}^*$ is equivalent to the smallest possible value of $\alpha_i$, $2/N \leq \alpha_i \leq \alpha_{\max}$, such that $g(\alpha_i) \leq \kappa_i$.

Suppose the time to successfully transmit an agent or query at a single node

Table 3
Balance equations revised for other network traffic.

| *State* | *Condition(s)* | *Balance Equation* |
|---|---|---|
| $(0,0,0)$ | None | $[\alpha_i N\lambda_i + \gamma_i + \tau_i]\, p_{0,0,0} = \delta_i p_{1,1,0} + [\beta_i + \mu/(1+\theta)]\, p_{0,0,1}$ $+\delta_i p_{1,0,0}$ |
| $(0,0,q)$ | $q \geq 1$ | $[\alpha_i N\lambda_i + \gamma_i + \tau_i + q\mu/(q+\theta) + q\beta_i]\, p_{0,0,q} = (\gamma_i + \tau_i)\, p_{0,0,q-1}$ $+ [(q+1)\,\mu/(q+1+\theta) + (q+1)\,\beta_i]\, p_{0,0,q+1} + \delta_i p_{1,1,q} + \delta_i p_{1,0,q}$ |
| $(l,0,0)$ | $l \geq 1$ | $(\alpha_i N\lambda_i + \tau_i + l\delta_i)\, p_{i,0,0} = \delta_i p_{l+1,1,0} + [(l+1)\delta_i]\, p_{l+1,0,0}$ $+ [\beta_i + \mu/(1+\theta)]\, p_{l,0,1} + \mu/(1+\theta) p_{l,1,0} + \lambda_i p_{l-1,0,0}$ |
| $(l,m,0)$ | $l,m \geq 1,$ $l \geq m$ | $[(l-m)\,\delta_i + \alpha_i N\lambda_i + \tau_i + m\delta_i + m\mu/(m+\theta)]\, p_{l,m,0} =$ $(m+1)\,\delta_i p_{l+1,m+1,0} + [\beta_i + \mu/\,(m+1+\theta)]\, p_{l,m,1}$ $+(m+1)\,\mu/(m+1+\theta) p_{l,m+1,0} 1_{l>m} + (\alpha_i N - 1)\,\lambda_i p_{l-1,m-1,0}$ $+ [(l+1-m)\,\delta_i]\, p_{l+1,m,0} + \lambda_i p_{l-1,m,0} 1_{l>m}$ |
| $(l,0,q)$ | $l \geq 1,$ $q \geq 1$ | $[l\delta + \alpha_i N\lambda_i + \tau_i + q\beta_i + q\mu/(q+\theta)]\, p_{l,0,q} =$ $[(q+1)\beta_i + (q+1)\,\mu/(q+1+\theta)]\, p_{l,0,q+1} + (l+1)\delta_i p_{l+1,0,q}$ $+\tau_i p_{l,0,q-1} + \delta_i p_{l+1,1,q} + [\mu/(q+1+\theta)]\, p_{l,1,q} + \lambda_i p_{l-1,0,q}$ |
| $(l,m,q)$ | $l,m \geq 1,$ $l \geq m,$ $q \geq 1$ | $[(l-m)\,\delta_i + \alpha_i N\lambda_i + \tau_i + m\delta_i + q\beta_i + (m+q)/(m+q+\theta)\mu]\, p_{l,m,q} =$ $(m+1)\delta_i p_{l+1,m+1,q} + [(q+1)\beta_i + (q+1)\,\mu/\,(m+q+1+\theta)]\, p_{l,m,q+1}$ $+ [(m+1)\,\mu/\,(m+1+q+\theta)]\, p_{l,m+1,q} 1_{l>m} + (\alpha_i N - 1)\lambda_i p_{l-1,m-1,q}$ $+\tau_i p_{l,m,q-1} + [(l+1-m)\delta]\, p_{l+1,m,q} + \lambda_i p_{l-1,m,q} 1_{l>m}$ |
| $(l,m,q)$ | $l < m,$ $q \geq 0$ | This state cannot occur as the number of agents in the transmission queue will never exceed the number of agents in the event table. |

is an exponentially distributed random variable with mean $1/\mu = 0.2$. We are interested in optimizing the replication level for a specific resource with agent and query parameters defined by Table 4. For this particular example, we ignore the effect of traffic other than that related to the agents and queries of interest (i.e., $\theta = 0$), and let $L = Q = 9$. These values of $L$ and $Q$ are sufficiently large to minimize the effect of blocking probabilities on the solution.

Following the solution procedure described in Section 3, we solve the mathematical program (5). The objective function and corresponding optimal solution are shown in Figure 1. Based on the results of this energy-centric analysis, the total number of transmissions is minimized when $\alpha_i = 0.0052$; thus,

Table 4
Parameters for the 5000-node network example.

| *Parameter* | *Value* |
|---|---|
| $\lambda$ | 0.005 |
| $\delta$ | 0.300 |
| $\gamma$ | 0.050 |
| $\beta$ | 0.500 |

$f(0.0052) \approx 0.2546$ which corresponds to an agent TTL of $(\alpha_i^* N - 1) = 25$.

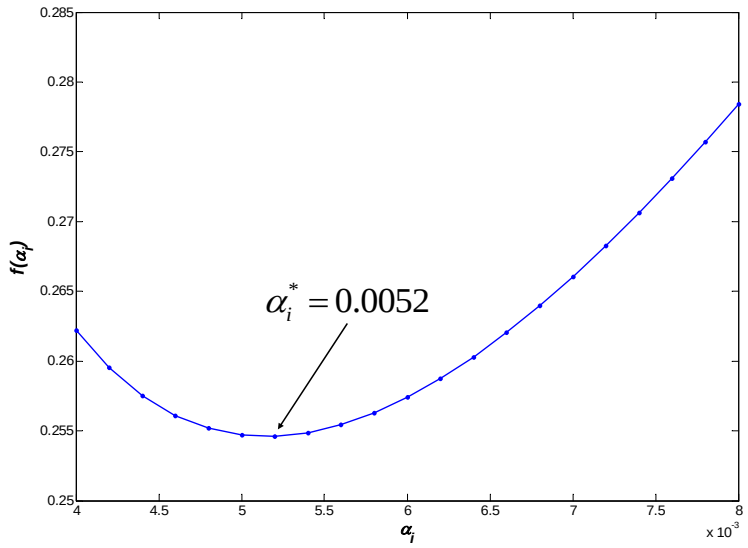


Fig. 1. Total rate of arrivals to a node's transmission queue as a function of $\alpha_i$.

We next determine if the proportion of query failures obtained at the computed value of $\alpha_i^*$ is acceptable, i.e., we check if $g(\alpha_i) \leq \kappa_i$. Using (11) yields the results shown in Figure 2. Based on these results, the proportion of query failures at $\alpha_i^* = 0.0052$ is $g(\alpha_i^*) \approx 0.2351$. Consequently, we conclude 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 $\alpha_i$ must be increased. The results achieved by examining a wider range of $\alpha_i$ values are presented in Figure 3. Based on this analysis, a value of $\alpha_{\kappa_i}^* = 0.0366$ (i.e., an agent TTL of 182) is necessary to achieve $g(\alpha_i) \leq 0.01$, 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 consequence, requires additional energy

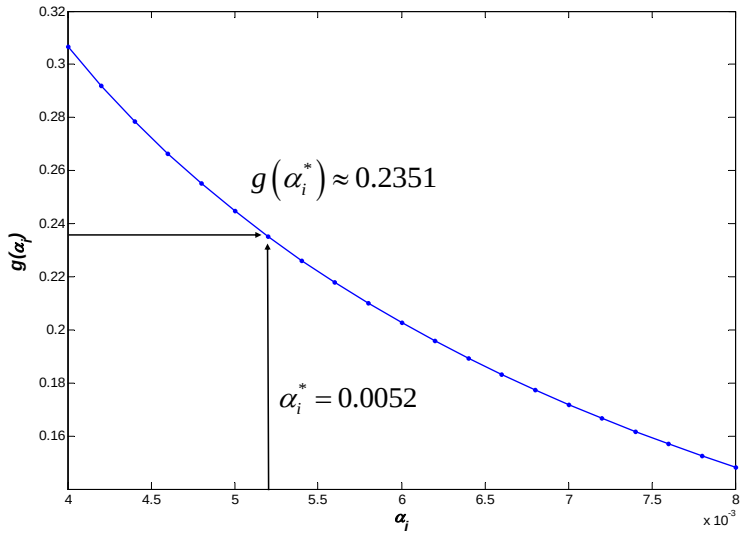


Fig. 2. Proportion of query failures as a function of $\alpha_i$.

expenditure to support. Furthermore, practical values of $\alpha_i$ 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, we posit that the proportion of query failures cannot be reduced indefinitely by increasing the number of resource replicates without bound. On the contrary, the value of $\alpha_i$ must be chosen carefully to prevent excessive query failures due to either insufficient replication or excessive traffic levels.

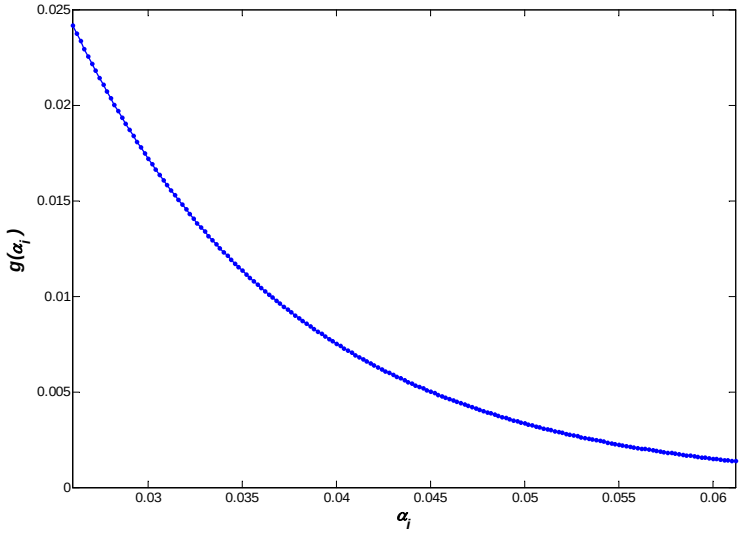


Fig. 3. Effect of increasing $\alpha_i$ on query failure rates.

The effect of $\alpha_i$ 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.

### *4.2 The effect of network parameters on the optimal resource replication level*

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 the environment, node addition/deletion/failure, and 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 analytic model, we are able to observe the resulting effects on the corresponding values of $\alpha_i^*$, $f(\alpha_i^*)$, and $\alpha_{\kappa_i}^*$. The effects of various network parameters are summarized in Table 5.

Table 5
Effects of parameter changes.

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

## 5 Simulation results

In Section 3, we developed an analytical queueing model for a WSN random-walk search algorithm that computed (approximately) the total mean arrival rate at a node. Subsequently, we showed how to determine the replication level that minimizes the node's arrival rate while simultaneously ensuring the proportion of query failures does not exceed a predetermined maximum. This model can be extremely useful when the interarrival and lead times of wit-

nessed events and query requests at a node are exponentially distributed. However, depending on the characteristics of the network and its associated applications, the lead times of arriving agents and queries may be distinctly non-exponential. Additionally, whereas our analytical model assumes that nodes are independent, such independence cannot be guaranteed in the context of rumor routing. Therefore, to examine the significance of the node independence assumption and different lead time distributions on the node model, we conducted a few simulation experiments using OPNET, a discrete-time network simulator.

First, we compared the total mean arrival rate and proportion of query failures obtained by the OPNET model with those obtained using our queueing model ((4) and (11)) over a range of replication levels. For each simulated response, three independent replications were conducted, each with a run length of 96 hours. We also constructed 95%-confidence intervals for each resource replication level. The OPNET simulation model *does not* assume node independence; therefore, it can be used to benchmark the results predicted by the analytical models. For the sake of consistency, the simulation parameters are identical to those given in Table 4. As seen in Figures 4 and 5, the results obtained from the OPNET simulator are very similar to those predicted by (4) and (11), indicating that the assumption of node independence does not appreciably impact the long-term predictive value of the analytical model.

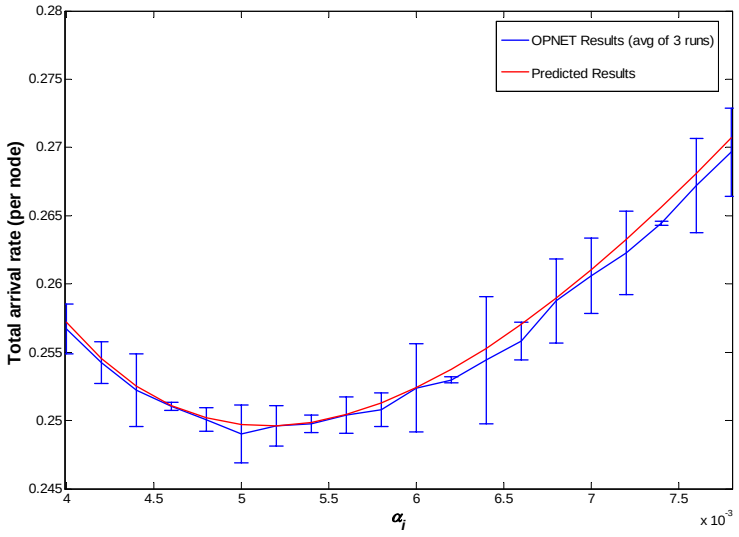


Fig. 4. Predicted versus simulated total arrival rates (Markovian model).

Next we examined the effect of a non-exponential lead time distribution for arriving agents and queries. As in the previous examples, the mean values of all parameters remain as in Table 4, and the mean service time is 0.2 units. However, the lead times of arriving agents and queries are now assumed to be uniformly distributed on the intervals $(0, 6.6666]$ and $(0, 4]$, respectively. Since the lead times of arriving agents and queries are no longer exponentially

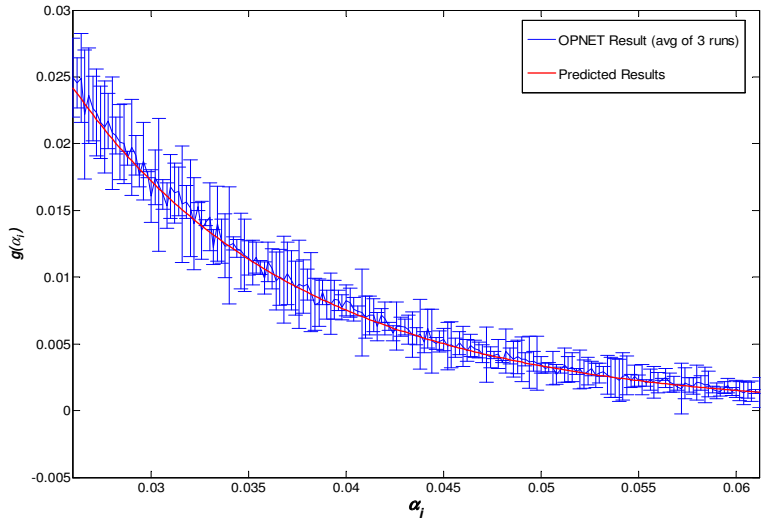


Fig. 5. Predicted versus simulated proportion of query failures (Markovian model).

distributed, the behavior of the event table is described by an M/G/$\infty$ queue. Despite the change in the distribution of the "service" time, (6) can still be used to compute the probability that a node's event table contains no applicable agents [14].

Since the assumption of Poisson agent and query arrivals is unchanged for both cases, Figure 4 still depicts the total rate of arrivals at a node in this system. Thus, we compare the proportion of query failures of this system to that predicted by the Markovian model. Figure 6 shows that the simulated proportion of query failures is significantly less than that predicted by the Markovian model over the range of resource replication levels. Thus, as anticipated, the Markovian model provides a conservative estimate of the proportion of query failures, $g(\alpha_i)$, when the leads times are uniform rather than exponential. Therefore, the exponential model may be used to bound the proportion of query failures from above for a given replication level.

## 6 Conclusions and future work

Our work characterizes the performance of random-walk WSN search algorithms when both agents and queries are assigned expiration times. Using a queueing approach, we analytically determine 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. We conclude that re-

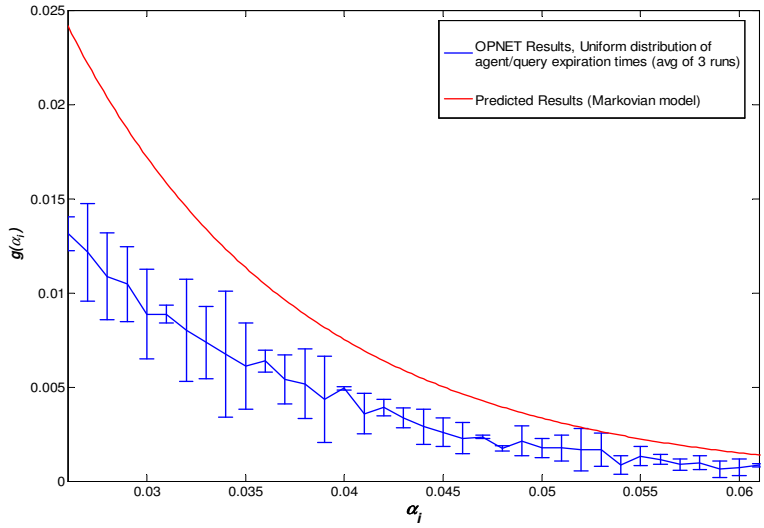


Fig. 6. Predicted versus simulated proportion of query failures (uniform model).

source replication levels must be carefully managed in order to strike a proper balance between energy efficiency and query failures. This work provides a means by which to determine an appropriate replication level. It has been shown that insufficient resource replication may actually increase energy expenditure (due to excessive query transmissions) and may lead to possible application failure. On the other hand, excessive resource replication reduces the query failure rate but needlessly consumes the network's aggregate storage capacity while consuming excessive energy to propagate agents. Excessive replication also increases traffic levels and congestion, resulting in a higher proportion of query failures.

Due to the computationally intensive nature of the proposed analytical model, it is better suited for use during the development phase of wireless sensor network design rather than the deployment phase. However, suitable approximations for the entire network may be devised to significantly reduce overall computational effort.

**Acknowledgements**: The authors acknowledge, with gratitude, the helpful comments of two anonymous referees and the Associate Editor.

## References

[1] 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.

[2] 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.

[3] 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–Volume II*, pp. 562-567, 2005.

[4] 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 (ICWN)*, Las Vegas, Nevada, 2003.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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)*, June 2006.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] C.R. Mann, R.O. Baldwin, J.P. Kharoufeh and B.E. Mullins, "A trajectory-based selective broadcast query protocol for large-scale, high-density wireless sensor networks," in *Telecommunication Systems: Modeling, Analysis, Design and Management*, Vol. 35, No. 1-2, pp. 67-86, 2007.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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*, September 2004.