Optimizing Data Replication for Expanding Ring-based Queries in Wireless Sensor Networks

Bhaskar Krishnamachari and Joon Ahn
Department of Electrical Engineering
University of Southern California
Los Angeles, CA 90089
{bkrishna, joonahn}@usc.edu
October 23, 2005

***Abstract*— We consider the problem of optimizing the number of replicas for event information in wireless sensor networks, when queries are disseminated using expanding rings. We obtain closed-form approximations for the expected energy costs of search, as well as replication. Using these expressions we derive the replication strategies that minimize the expected total energy cost consisting of search and replication costs, both with and without storage constraints. In both cases, we find that events should be replicated with a frequency that is proportional to the square root of their query rates. We validate our analysis and optimization through a set of realistic simulations that incorporate non-idealities including deployment boundary effects and lossy wireless links.**

I. Introduction

While the nodes in a sensor network can be operated in a continuous data gathering mode, this approach is not useful except for very simple applications. Continuous data gathering from all sensors is generally very inefficient if most of the sensed information is not essential, or if there are multiple sinks that may need to request different subsets of the sensed information at different times. In such contexts it is better to think of the sensor network as a decentralized data storage system (see [1] for an excellent survey of data-centric storage techniques). In such a data-centric storage approach, the sensed data can be either stored locally or at one or more remote locations within the network. Event information is obtained by sinks through queries that are issued on an on-demand basis.

In this work, we focus on the case of replicated event information stored at multiple storage points in the network in a randomized manner. Multiple replicas of an event (or in the case of large data items, pointers to where the original event information is stored) can be either placed carefully at predetermined locations or randomly. The former approach is exemplified by hash-based data centric storage techniques such as GHT [2], DIM [3], etc., and can be efficient since queries can be sent directly to the storage location. However, randomized storage of replicated information is justified in some scenarios when there is a high overhead for maintaining shared predetermined location information across the entire network (due to dynamics such as changes, movements and failures of nodes in the network). Randomized storage can also provide for a more load-balanced storage over time, and, in some cases, provide greater security by making it difficult to identify and target nodes containing critical information.

With unstructured, randomized storage, however, the querying nodes must resort to some form of blind search. We focus on expanding ring queries in which there are successive rounds of controlled floods with increasing TTL-values to detect the nearest copy of the queried information. While the tradeoffs we explore can be generalized to other search techniques, our motivation for focusing on expanding ring-based queries is that these have been relatively well studied [4], [5]. In particular, we use the dynamic programming algorithm proposed by Chang and Liu [5] to perform optimal expanding ring searches.

Intuitively, the performance of a TTL-based expanding ring search improves with additional replicas. When there are more randomly placed replicas in a network, the likelihood that the event being searched for is located within a smaller number of steps, close to the sink, becomes higher. However, this reduction in the expected search energy cost comes at the expense of an increased energy cost for replication. Our goal is to minimize the total expected energy cost consisting of search and replication costs by carefully selecting the optimal number

0-7803-9550-6/06/$20.00

of replicas. We assume there can be limited storage at each sensor node in some networks. In such scenarios, the optimization must explicitly consider storage constraints. We therefore consider both constrained and unconstrained versions of this optimization problem.

This paper is organized as follows. We first model the search cost of optimal TTL-based/exanding ring search in section II. While a constructive solution to optimal search is provided by modifying the dynamic programming algorithm developed by Chang and Liu [5], we find that obtaining a closed form exact expression for the cost of the optimal search as a function of number of replicas appears to be intractable. We therefore first develop bounds on the optimal search cost. An upper bound is provided by an expression we derive for the step-by-step expanding ring search. We also derive a lower bound using a genie argument. We show that both bounds decrease inversely with the number of replicas, motivating an approximation for the expected optimal search cost.

We then present expressions for the expected replication cost for disk and square deployments in section III. Then, we combine these expressions to provide the total combined cost of search and replication as a function of the number of replicas and solve for the optimal number of replicas with and without storage constraints in section IV.

We validate our analysis through a set of simulations in section V. These simulations are performed using a realistic wireless network topology generator [9]. Although we find that the node placement distributions and optimal search sequences can be significantly different between simulations and analysis, we find that the corresponding expected search and replication costs are quite similar and that the optimal replication number obtained through analysis matches the simulation results quite closely. Finally, we present concluding comments in section VI.

II. Modeling Search Cost

A. Scenario, assumptions, notation

We consider a circular area with nodes deployed with a uniform random distribution. Each node can communicate with any other node that is placed within a radio range R, and it is assumed that the network is sufficiently dense so that all nodes within a distance kR of the sink can be reached in k hops. The nodes in the circular area are all located within L hops of the sink. When modelling the search cost we assume that the sink is located in the center of the region (we will relax these assumptions in the simulation study in section V.)

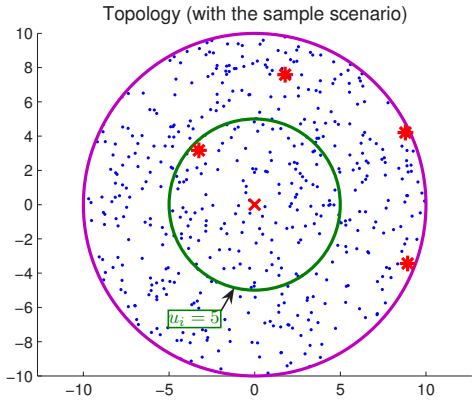


Fig. 1. Circular Topology used for Analysis

We assume that there is always a copy of the information being queried for, within the network. We also assume that all replicated copies of this information are placed at random uniformly within the network. The sink issues the query in successively expanding rings according to a search sequence (we shall describe below how the optimal search sequence is derived using dynamic programming). The cost of querying and replication is modelled as being directly proportional to the number of transmissions incurred for each.

Figure 1 illustrates a sample network for $R = 1$. The sink is denoted by an 'x' while the replicas of a particular event that is being queried for are denoted by a star. Say the expanding ring search is denoted by the query sequence $\{5, 10\}$, then the query is carried out in two steps. First all nodes within the first five hops (i.e. within a distance 5) are searched through a controlled flood. If the nearest copy of the replicated information is located within this distance (as in the figure), the search stops right at this point. Else, another flood covering the whole network within a distance 10 is issued.

Table I summarizes the notation to be used in the analysis.

B. Nearest replica location distribution

We first derive the distribution of the nearest copy of an event when events are replicated randomly in the circular deployment area. This distribution is an important building block for the analysis as it aids in determining the optimal search strategy when events are replicated in the network.

The expected number of nodes within one hop of the sink is then $a = \pi R^2 \rho$, where ρ is the node density. The expected number of nodes that are exactly k hops away is $a(k^2 - (k-1)^2) = a(2k-1)$. The total number of

Symbol	*Meaning*
ρ	Node density
R	Radio range
N	Total number of nodes in the network
a	Number of first hop neighbors of sink
L	Maximum number of hops from the sink
n (n_i)	Number of replicas (for the i^{th} event)
u_i	i^{th} TTL element of search sequence u
$C_f(k)$	Cost of controlled flooding with a TTL value of k
$F(k\|c)$	Conditional tail probability of locating the nearest copy of the event beyond k hops, given that it is not located within c hops of the sink
$V(c)$	Value function for the dynamic program

TABLE I

NOTATION USED

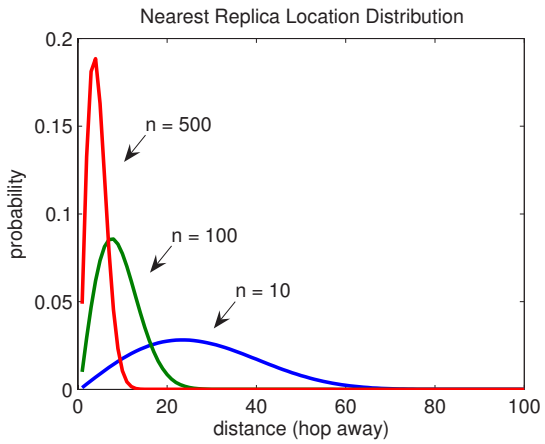


Fig. 2. Illustration of the probability mass function for the nearest replica (L=100, a = 10)

nodes that are located within the circular region of L hops from the sink is then given as

$$N = \sum_{k=1}^{L}(2k-1)a = aL^2 \tag{1}$$

Say n replicas of an event are created and placed randomly in the network (in addition to the original copy at the source sensor). Let $X_{min}(n)$ be the random variable representing the hop count of the nearest copy of the event from the sink. The probability that all $(n+1)$ copies of the event information are located more than k hops away from the sink is then given by the expression:

$$Pr\{X_{min}(n) > k\} = \left(1 - \frac{k^2}{L^2}\right)^{n+1} \tag{2}$$

Figure 2 illustrates how this distribution varies with the number of replicas in a typical network. As may be intuitively expected this distribution shifts to the left (i.e. the nearest copy is located closer to the sink) as the number of replicas increases. This should result in a lower search cost with increasing replication size.

A related quantity that is of use in determining the optimal expanding ring strategy is the conditional probability that the nearest copy of the event is located more than k hops away given that it is known that it is not located within c hops. This is expressed as follows (assuming $k \geq c$):

$$\begin{aligned} F(k \mid c) &= Pr\{X_{min}(n) > k \mid X_{min}(n) > c\} \\ &= \left(\frac{L^2 - k^2}{L^2 - c^2}\right)^{n+1} \end{aligned} \tag{3}$$

C. Optimal expanding ring search

Any expanding ring search can be characterized as a vector $u = \{u_1, u_2, \ldots u_m\}$ that describes the sequence of successive TTL values for controlled flooding in each step. To ensure that the entire area is covered in the worst case, u_m is set to L. For example, let $u = \{1, 5, 10\}$ for a network where the maximum hop count is $L = 10$. Then the expanding ring search would proceed as follows: first the nodes within 1-hop are searched for the event through a controlled flood with TTL value of 1. If no copies of the event are located in this first step, then all nodes within 5 hops are searched for the event through a larger controlled flood. If still no copies of the event are located in the second step, then all nodes in the network (within 10 hops) are searched. If at any step at least one copy of the event is located, the search terminates successfully at that step.

We will assume that each transmission (and the corresponding receptions) incurs a unit cost. The cost of the controlled flooding incurred in the i^{th} search step is given as:

$$C_f(u_i) = (1 + a(u_i - 1)^2) \tag{4}$$

For a given search sequence vector u, assuming there are $(n+1)$ total copies of the event in the network, the expected search cost is then

$$C_{search,u} = \sum_{i=1}^{m} C_f(u_i) \cdot Pr\{X_{min}(n) > u_{i-1}\} \tag{5}$$

where $Pr\{X_{min} > u_0\}$ is defined to be 1 (since the search sequence starts with u_1, and it is guaranteed that there is at least one copy of the event being queried somewhere in the network).

To minimize this search cost, the optimal TTL sequence must be obtained. Chang and Liu [5] have developed a dynamic programming solution to solve this problem. This dynamic program uses the following recursive property.

Let the value function $V(n)$ be the minimum expected cost-to-go (over all choices of TTL values), given that the most recently used TTL value k did not locate the object. Then

$$\begin{aligned} V(L) &= 0 \\ V(c) &= \min_{c+1 \le k \le L} \{C_f(k) + F(k|c) \cdot V(k)\} \end{aligned}$$

In the case of multiple replicas, we use the tail distribution $F(k|c)$ we obtained in equation (3). The optimal search sequence u is obtained by recursively calculating the value function, and then tracking back through the choices made at each step to determine the optimal TTL value for each stage. This search sequence can then be used in equation (5) to determine the expected cost of the optimal strategy. However, this algorithmic approach does not yield a tractable closed form expression for this cost as a function of the number of replicas. We therefore first try to derive lower and upper bounds on the cost, before developing an approximate expression for the optimal cost based on the bounds.

D. Genie-assisted lower bound on optimal search cost

We first obtain a genie-assisted lower bound for the optimal cost. Imagine, before each query, we had a genie or oracle that gave the exact distance from the sink (in number of hops) to the nearest located copy of the event. Let us denote this exact distance by x_{min}. Knowing x_{min} before the query is issued, the best possible search strategy for the sink to apply is to set the first TTL value of the search sequence to this value, i.e. set $u_1 = x_{min}$. Such a genie-assisted strategy is guaranteed to find the information in one step, with a cost of $C_f(x_{min})$. Consider any other expanding ring sequence u'; if $x_{min} > u'$, the expected cost of that strategy must be higher than that of the Genie-assisted strategy because the nodes in the first ring will have to be covered twice or more in the search; if $x_{min} < u'$, then also the expected search cost will be higher because a greater number of nodes will have to be searched in the first ring. Hence the genie-assisted strategy is guaranteed to provide a lower bound to any expanding ring strategy.

The expected cost of the genie technique $C_{s,lower}$ can be derived as follows [1]:

$$\begin{aligned} & C_{s,lower} \\ &= E\left[C_f(X_{\min})\right] \\ &= E\left[\left(1 + a\left(X_{\min} - 1\right)^2\right)\right] \\ &= \left(1 + a\left(E\left[X_{\min}^2\right] - 2E\left[X_{\min}\right] + 1\right)\right) \end{aligned} \quad (6)$$

Now,

$$P\left\{X_{\min} \le k\right\} = F_{\min}(k) = 1 - \left(1 - \frac{k^2}{L^2}\right)^{n+1}$$

The pdf for $X_{\min}$ is then derived as

$$f_{\min}(k) = \frac{dF_{\min}(k)}{dk} = \frac{2(n+1)}{L^2} k \left(1 - \frac{k^2}{L^2}\right)^n$$

Now we can obtain the necessary expectations as follows:

$$\begin{aligned} E\left[X_{\min}\right] &= \int_0^L k f_{\min}(k) dk \\ &= \frac{2(n+1)}{L^2} \int_0^L k^2 \left(1 - \frac{k^2}{L^2}\right)^n dk \\ &= \frac{L(n+1)\sqrt{\pi}}{2} \cdot \frac{\Gamma(n+1)}{\Gamma\left(n + \frac{5}{2}\right)} \end{aligned} \quad (7)$$

$$\begin{aligned} E\left[X_{\min}^2\right] &= \int_0^L k^2 f_{\min}(k) dk \\ &= \frac{2(n+1)}{L^2} \int_0^L k^3 \left(1 - \frac{k^2}{L^2}\right)^n dk \\ &= \frac{2(n+1)}{L^2} \frac{L^4}{2n^2 + 6n + 4} \\ &= \frac{L^2}{n+2} \end{aligned} \quad (8)$$

Substituting equations (7) and (8) into equation (6), we get that

$$C_{s,lower} = \left(1 + a\left(\frac{L^2}{n+2} - \sqrt{\pi} L (n+1) \frac{\Gamma(n+1)}{\Gamma\left(n + \frac{5}{2}\right)} + 1\right)\right)$$

Proposition 1: The search cost of the optimal expanding ring strategy is lower-bounded by a function that decreases with the number of replicas n as $\frac{1}{n+2}$.

Proof: Note that the Gamma function $\Gamma(k)$ is a monotonically increasing function for $k \ge 2$ that has the property

[1] We have used a continuous probability domain approximation to obtain closed-form expressions here. We have verified through numerical simulations that the obtained expression for the lower bound matches the bound from the discrete version of the problem very closely.

that $\Gamma(k) = (k-1)!$ for any integer $k \geq 2$. Then,

$$\begin{aligned} C_{s,optimal} &> C_{s,lower} \\ &> \left(1 + a\left(\frac{L^2}{n+2} - \sqrt{\pi}L(n+1)\frac{\Gamma(n+1)}{\Gamma(n+2)} + 1\right)\right) \\ &= \left(1 + a\left(\frac{L^2}{n+2} - \sqrt{\pi}L\frac{(n+1)n!}{(n+1)!} + 1\right)\right) \\ &= \left(1 + a\left(\frac{L^2}{n+2} - \sqrt{\pi}L + 1\right)\right) \end{aligned}$$

□

E. Upper bound on optimal search cost

We now derive an upper bound on the cost of the optimal search strategy. One simple search strategy that is found empirically to match the performance of the optimal strategy closely for large number of replicas n is the step-by-step expanding ring search, in which the search sequence is simply $\{1, 2, 3, 4, \ldots\}$. The expected cost for this strategy is given as:

$$\begin{aligned} & C_{s,upper}(n) \\ &= \sum_{k=1}^{L} C_f(k) P\{X_{\min} > k-1\} \\ &= \sum_{k=1}^{L} \left(1 + a(k-1)^2\right)\left(1 - \frac{(k-1)^2}{L^2}\right)^{n+1} \end{aligned}$$

This expression can be closely approximated by a continuous integral:

$$\begin{aligned} C_{s,upper}(n) &\approx \int_0^L ak^2\left(1 - \frac{k^2}{L^2}\right)^{n+1} dk \\ &= \frac{\sqrt{\pi}aL^3}{4}\frac{\Gamma(n+2)}{\Gamma(n+3.5)} \end{aligned}$$

Proposition 2: The search cost of the optimal expanding ring strategy is upper-bounded by a function that is proportional to $\frac{1}{n+2}$.

Proof: Once again, from the properties of the Gamma function, we get that

$$\begin{aligned} C_{s,optimal} < C_{s,upper}(n) &< \frac{\sqrt{\pi}aL^3}{4}\frac{\Gamma(n+2)}{\Gamma(n+3)} \\ &= \frac{\sqrt{\pi}aL^3}{4}\frac{1}{n+2} \end{aligned}$$

□

Figure 3 compares the upper and lower bounds for the search cost with the numerically computed optimal search cost.

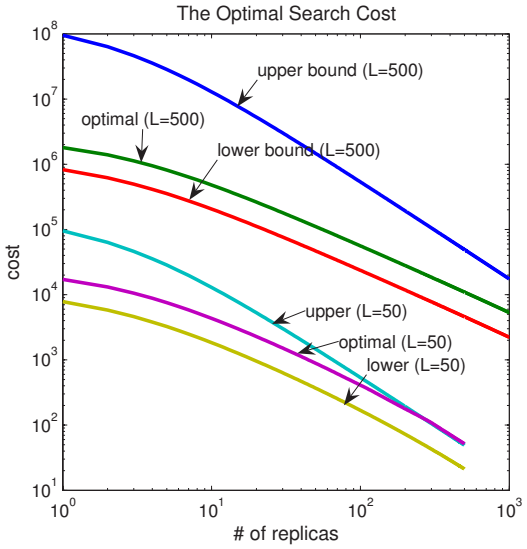


Fig. 3. Bounds on optimal search cost

L (max TTL)	c (Curve-fit constant)
10	1.47845
50	1.99568
100	2.07722
500	2.14608
1000	2.15476

TABLE II
BEST-FIT CONSTANT FOR SEARCH COST APPROXIMATION

F. Approximation for optimal search cost

Based on propositions 1 and 2, it is reasonable to model the search cost of the optimal strategy as being proportional to $\frac{1}{n+2}$. We thus obtain the following approximation for the search cost of the optimal expanding ring strategy:

$$C_{search,optimal} \approx c \cdot aL^2 \cdot \frac{1}{n+2} \tag{9}$$

In this approximation, c is a curve-fitted constant, that is seen to converge to a value close to 2.15 as the size of the deployment area increases (i.e. for large L), as shown in table II.

Figure 4 compares the approximate search cost expression with the numerically optimal search strategy. We see a close match, particularly when the network is large and the number of replicas is relatively small.

III. Modeling the Replication Cost

We are assuming that events are likely to be generated at any location in the network, and that they are replicated at the different locations at random. We assume

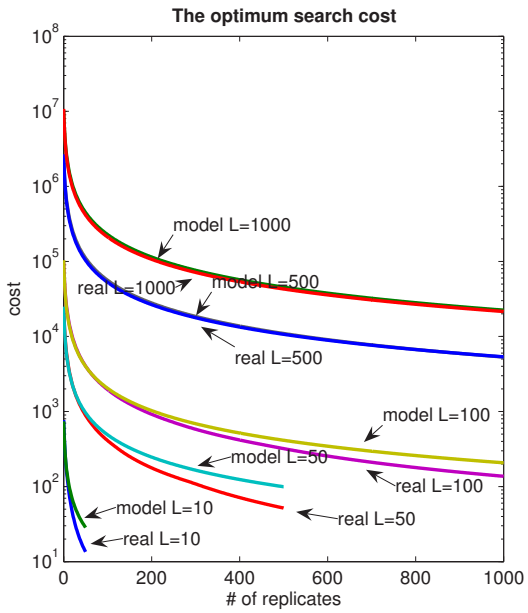


Fig. 4. Approximation for Optimal Search Cost

that n replicas of the original are created and individually placed at each location through unicast routing on the shortest path between the random source and storage point.

Assuming sufficient node density, the number of transmissions required to move data between any pair of locations a distance d apart along the shortest path between them is approximately d/R. Thus the expected cost of creating any replica is given by the ratio of expected distance between any pair of points in the area and the radio range R. We present expressions for the expected distance between two points, for circular and square regions.

A. Circular area

For a circular region, there is a known geometric result referred to as disk line picking [6], which gives the expected distance between any two points in a unit circle to be:

$$\begin{aligned} & E\left[d_{circle}\right] \\ = & \frac{1}{\pi}\int_0^1\int_0^1\int_0^\pi \sqrt{r_1+r_2-2\sqrt{r_1 r_2}\cos\theta}\, d\theta dr_1 dr_2 \\ = & \frac{128}{45\pi} \end{aligned}$$

Using this result, we get the following expression for the expected cost of creating n replicas of the event information in a circular region of radius LR to be:

$$C_{replication,circle}(n) = \frac{128LRn}{45\pi R} = \frac{128Ln}{45\pi} \tag{10}$$

B. Square area

Similarly, for a square, the expected distance in the square of width wR is

$$\begin{aligned} & E[d_{square}] \\ = & wR\underbrace{\int\cdots\int_0^1}_{4}\sqrt{(x_1-y_1)^2+(x_2-y_2)^2}\underbrace{dx_1\cdots dy_2}_{4} \\ = & wR\frac{2+\sqrt{2}+5\ln\left(1+\sqrt{2}\right)}{15} \approx 0.521405wR \end{aligned}$$

From this, we get that

$$C_{replication,square}(n) \approx 0.52wn \tag{11}$$

IV. Optimization Formulation

We can formulate the problem of optimizing the number of replicas for each event as follows:

$$\begin{aligned} \textit{Minimize} \quad & C_{NET}(\bar{n}) \\ \textit{s.t} \quad & g(\bar{n}) = \sum_{i=1}^{m} n_i + m \le S \\ & 0 \le n_i \le N-1, \quad \forall i \end{aligned} \tag{12}$$

where

$$C_{NET}(\bar{n}) = \sum_{i=1}^{m} q_i C_{search}(n_i) + \sum_{i=1}^{m} C_{replication}(n_i) \tag{13}$$

Here q_i is the query rate for the i^{th} of m events, n_i is the number of replicas of event i, and S is the total network storage limit. For a circular region, the expressions for $C_{search}(n_i)$ and $C_{replication}(n_i)$ are as obtained in equations (9) and (10), respectively. We solve this problem using the method of Lagrange multipliers. The Lagrangian function for this inequality-constrained optimization problem can be expressed using a slack variable s as follows:

$$L(\bar{n},\lambda) = C_{\text{NET}}(\bar{n}) + \lambda\left(g(\bar{n}) - S + s^2\right) \tag{14}$$

It can be shown that the objective function is convex; hence, the following first-order conditions are sufficient for global minimization:

$$\frac{\partial L}{\partial n_i} = -\frac{q_i a L^2 c}{(n_i+2)^2} + \frac{128L}{45\pi} + \lambda = 0 \tag{15}$$

$$\frac{\partial L}{\partial \lambda} = \sum_{i=1}^{m} n_i + m - S + s^2 = 0 \tag{16}$$

$$\frac{\partial L}{\partial s} = 2\lambda s = 0 \tag{17}$$

i) When the constraint is inactive we can solve directly from equation (15), setting $\lambda = 0$:

$$n_i^* = \sqrt{\frac{45\pi aLc}{128}} \cdot \sqrt{q_i} - 2 \tag{18}$$

ii) When the constraint is active, (i.e. s=0, $\lambda \geq 0$), we get from equation (15):

$$n_i^* = \sqrt{\frac{acL}{\frac{128}{45\pi} + \frac{\lambda}{L}}} \cdot \sqrt{q_i} - 2 \tag{19}$$

λ is a constant that can be solved by substituting the above equation into equation (16), setting $s = 0$:

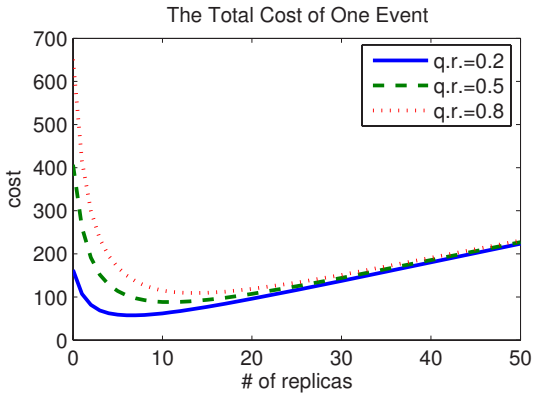
$$\lambda = \frac{acL^2 \left(\sum_{i=1}^{m} \sqrt{q_i}\right)^2}{(S+m)^2} - \frac{128L}{45\pi} \tag{20}$$

Substituting this back into (19), we get the following simplified expression:

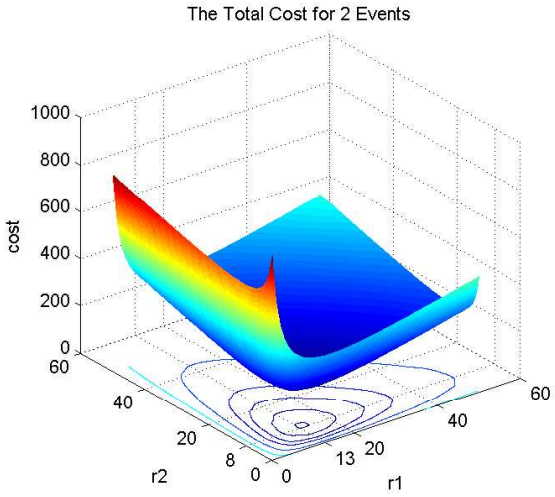
$$n_i^* = \frac{\sqrt{q_i}}{\sum_{i=1}^{m} \sqrt{q_i}}(S+m) - 2 \tag{21}$$

To determine whether the constraint is inactive or active, it is sufficient to verify whether the sum of n_i^* obtained from equation (18) is less than $S - m$. If not, then equation (21) should be used to compute the optimal constrained n_i^*. A striking observation is that in both cases the optimal strategy is to have the replication number of each event to be proportional to the square root of the query. We note that this outcome is very similar to a result in unstructured peer-to-peer wired networks [8], which also argues for replicating content with a rate proportional to the square root corresponding frequency of access. However, there are key differences between that work and ours, including the type of search analyzed (expanding rings in a wireless network with a geometrically defined 2-D structure versus random walk on an arbitrary wired network graph), and the absence of replication cost.

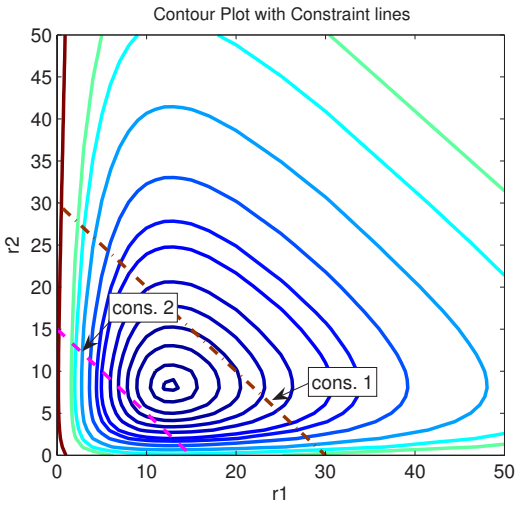
Figure 5(a) shows the total cost of querying and replication C_{NET} as a function of the number of replicas for different query rates for a single event. Figure 5(b) illustrates how the total cost may vary for the case of two events, as a function of the number of replicas for each event. Figure 5(c) shows the contours of this function, along with two sets of lines that represent different storage constraints. With the first storage constraint (a large value of S), there is sufficient storage available that the unconstrained optimal point A can be selected as the operating point, by allocating the corresponding optimal



(a)



(b)



(c)

Fig. 5. (a) Total expected cost for a single event showing that the optimal replication number varies as a function of query rates (b) a surface plot showing total expected cost for two events, and (c) a contour plot of the total cost for two events showing storage constraints (1, 2) and corresponding optimal solution points (A, B)

Parameter		*Value*
Channel	path loss exponent	3.0
	shadowing std. deviation	3.8
	PL(d_0)	55.0
	d0	1
Radio	Modulation	3 (NCFSK)
	Encoding Option	3 (Manchester)
	Radio Output Power	-21.0
	Noise Floor	-105.0
	Preamble Length	2 bytes
	Frame Length	50 bytes
Topology	Number of nodes	1010
	Physical Terrain	(80, 80)
	Option	Uniform Deployment

TABLE III
RADIO PARAMETERS FOR SIMULATION

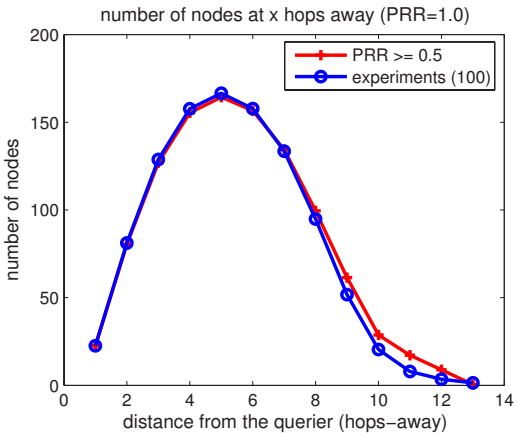


Fig. 6. Experiment for the node number distribution

number of replicas for both events. However, under the tighter storage constraint 2 (smaller S), the original unconstrained optimal solution lies outside the feasible operation region. Hence, point B, which minimizes the function while maintaining storage feasibility, provides the optimal constrained solution in this case.

V. REALISTIC SIMULATIONS

A. Methodology

We use a realistic link layer model generator for wireless sensor networks [9], which determines the location of each node and the packet reception rate (PRR) of each pair of nodes. Table III shows parameters for our wireless sensor network topology to simulate on (corresponding to a dense deployment of Mica 2 motes). Given the realistic topology, our simulator performs the following procedures at each round:

1) Randomly choosing a source node which is considered to have the original event information
2) Counting the actual replication cost for n replicas chosen randomly
3) Randomly choosing a querier node in the given node pool.
4) Counting the actual search cost using the optimal search strategy.

Our numerical results are computed based on 10000 rounds for each n value.

1) Counting the actual replication cost: The replication is done not by flooding, but rather through individual unicast transmissions from the source to the requisite number of random replication locations. We use the ETX (expected number of transmissions with retransmissions) metric [10] to define the routing strategy for the unicast transmissions. Specifically, the transmission on the edge from i to j costs $\frac{\beta}{PRR_{i,j}}$, where β is the cost of a single transmission, and $PRR_{i,j}$ is the packet reception rate from i to j, and a message between any pair of nodes in the network is routed along the shortest cost path between them. Here, we have assumed that acknowledgement packets (which are likely to be much shorter) are always received reliably. In the simulation results we count the actual replication cost by counting the actual total number of transmissions on the shortest unicast path and multiplying it by β.

2) Counting the actual search cost: In order to find out the search cost, we need to find out the optimal search strategy. In order to use the optimal search strategy from the dynamic programming methodology [5], we need to know the distribution of number of nodes with respect to the hop distance from the querying node. However, it is not easy to determine the hop-distribution in the realistic wireless topology considered in the simulations, where the links are lossy and asymmetric. Even for a given topology, the number of nodes of i^{th} hop (for a single query event) is a random variable whose expectation is not easily obtained. Since we need to compute the distribution for any querying node in the network, it is particularly important to obtain an approximation that can be calculated simply. The approach we have taken is to look at the hop-distribution of the subgraph formed when all links with packet reception rates below 0.5 are blacklisted from the network. As a sanity check, we have compared the results obtained from this process with the average of the number of nodes at each hop from 100 simulation experiments where these are determined probabilistically in each run according to the PRR values at each edge. The two approaches show remarkably similar results (see figure 6).

Let $H[i]$ denote the set of nodes that are reachable from

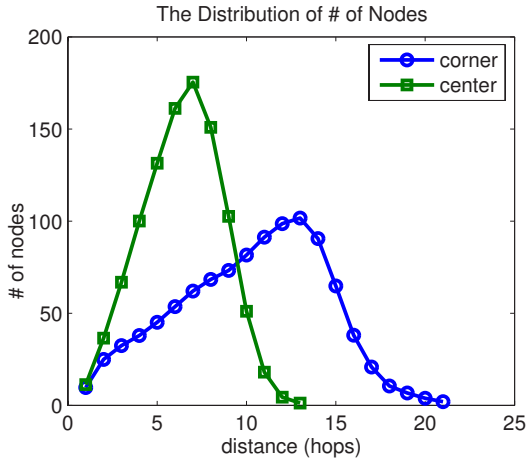


Fig. 7. Distribution of the number of nodes as a function of distance from querier in a uniform square area deployment

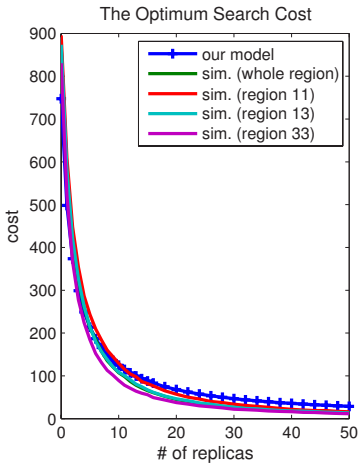


Fig. 8. Comparison of analytical and simulated search costs as a function of replication size (Region 13 refers to the area of first row and third column when the whole region is divided into 3 rows and 3 columns, and so a querier is randomly selected only in region 13 for the simulation result of region 13.)

the querying node at a distance of i hops. We use the following conditional tail distribution for the dynamic programming,(for $k \geq c$):

$$P\{X_{min} > k \mid X_{min} > c\} = \left(\frac{\sum_{i>k} |H[i]|}{\sum_{i>c} |H[i]|} \right)^{n+1}$$

Following the resulting optimal search strategy, the simulator floods a series of queries until it finds one of the copies of the event. Note that in our simulations, although two queries have the same TTL value, one query might find the event but the other might not, in the same network. It is because the coverage of first query is not necessarily same as that of the second one (because of the lossy wireless links).

B. Results

In our simulations, we relax several assumptions from the mathematical analysis, so that (1) the querier can be any node in the network, (2) the network topology is not necessarily circular (it is the square area for our simulation), and (3) there might be the boundary effect. With these relaxations, the actual optimal search sequence of a node might be different from that of another node when they are considered as a querier at each time. For example, the optimal search sequence of a corner node is $\{2, 8, 12, 15, 17, 19, 20\}$, while that of a center node is $\{2, 4, 6, 7, 8, 9, 10, 11\}$ when there are two replicas of the queried event.

First of all, the theoretical values of our model are as follows;

$$C_{search}(n) = \frac{cN}{n+2} = \frac{1.48 \times 1010}{n+2}$$

$$C_{replication}(n) = \frac{128Ln}{45\pi} = \frac{128 \times 10}{45\pi} n$$

where the node density variable $a \approx 10$ is obtained from the simulation (since this depends on radio and deployment settings), the value of $L \approx 10$ is obtained from $aL^2 = N = 1010$, and c is obtained accordingly from table II. Therefore, assuming the query rate is 1, the optimal number of replicas is as follows by equation (18);

$$\begin{aligned} n^*_{th} &= \sqrt{\frac{45\pi}{128} qaLc} - 2 \\ &= \sqrt{\frac{45\pi}{128} \times 10 \times 10 \times 1.48} - 2 \\ &= 10.7852 \approx 11 \end{aligned}$$

The optimal number of replicas from the simulation is found to be $n^*_{sim} = 12$ (see figure 10. Figure 8 shows the optimal search cost of the simulation and our model, and figure 9 shows the replication cost. As we can see from these figures, our model meets the simulation results very well even with relaxed assumptions. The similarity of the results despite seeing very different hop distance distributions in the simulations suggests that the cost of the optimal search is quite robust to this distribution, particularly in the presence of replicas.

VI. Conclusions

We have shown how the number of replicas of event information can be optimized for expanding ring-based queries in sensor networks. We have found that a square-root-proportional replication strategy provides optimal

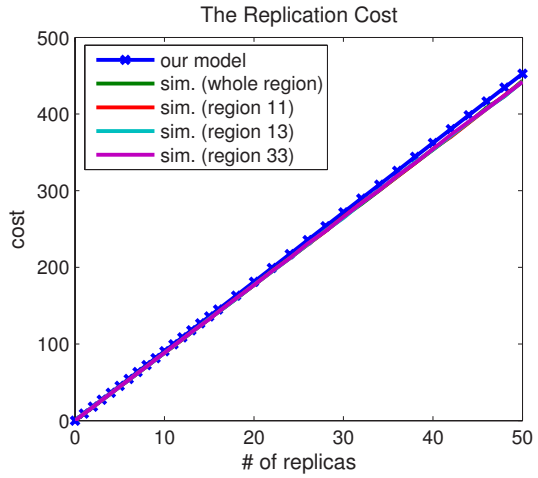


Fig. 9. Comparison of analytical and simulated replication costs as a function of replication size

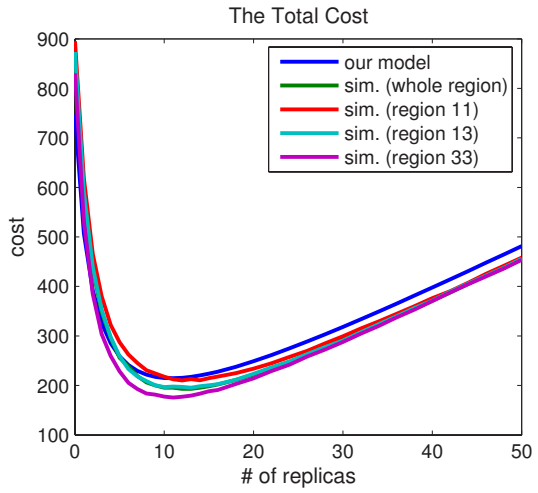


Fig. 10. Comparison of analytical and simulated total cost of search and replication as a function of replication size

performance both with and without storage constraints. Detailed realistic simulations validate the analysis.

There are several directions in which these results can be extended. The analysis could be extended to other querying mechanisms, including structured storage, since there is a similar tradeoff between search and replication costs in many other settings. The analysis could also be extended to consider more irregular deployment areas, including three-dimensional deployments. We plan to develop distributed implementations which allow for optimal or near-optimal replication without global knowledge of the relative query rates for all events. We also plan to investigate the scaling behavior of querying in storage-constrained sensor networks.

VII. Acknowledgements

This work is supported in part through NSF research grants numbered 0325875, 0347621, 0435505, and 0430061.

References

[1] R. Govindan, "Data-centric Routing and Storage in Sensor Networks," in *Wireless Sensor Networks*, Eds. Raghavendra, C.S., Sivalingam, K.M., and Znati, T., Kluwer Academic Publishers, 2004, pp. 185-206.

[2] 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", In *Mobile Networks and Applications (MONET), Special Issue on Wireless Sensor Networks*, 8:4, Kluwer, August 2003.

[3] X. Li, Y.J. Kim, R. Govindan, and W. Hong, "Multi-dimensional Range Queries in Sensor Networks", *The First ACM Conference on Embedded Networked Sensor Systems (Sensys03)*, November 2003.

[4] Z. Cheng and W. Heinzelman, "Flooding Strategy for Target Discovery in Wireless Networks," *Proceedings of the Sixth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2003)*, September 2003.

[5] N. Chang and M. Liu, "Revisiting the TTL-based Controlled Flooding Search: Optimality and Randomization," *Proceedings of the Tenth Annual International Conference on Mobile Computing and Networks (ACM MobiCom)*, September, 2004.

[6] E. W. Weisstein, "Disk Line Picking", From *MathWorld–A Wolfram Web Resource*, available online at http://mathworld.wolfram.com/DiskLinePicking.html

[7] B. Krishnamachari, J. Ahn, "Optimizing Data Replication for Expanding Ring Queries in Wireless Sensor Networks", Full Length Technical Report, available online at http://ceng.usc.edu/∼bkrishna/

[8] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," *ACM SIGCOMM*, August 2002.

[9] M. Zuniga, B. Krishnamachari, A Realistic Wireless Link Quality Model and Generator, available online at http://ceng.usc.edu/∼anrg/downloads.html

[10] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-Hop Wireless Routing", ACM MobiCom 2003.