A Modeling Framework for Computing Lifetime and Information Capacity in Wireless Sensor Networks

Enrique J. Duarte-Melo[1], Mingyan Liu[1] and Archan Misra[2]

[1] Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, USA
ejd@eecs.umich.edu, mingyan@eecs.umich.edu
[2] T. J. Watson Research Center at IBM, NY, USA
archan@us.ibm.com

Abstract. In this paper we investigate the expected lifetime and *information capacity*, defined as the maximum amount of data (bits) transferred before the first sensor node death due to energy depletion, of a data-gathering wireless sensor network. We develop a fluid-flow based computational framework that extends the existing approach, which requires precise knowledge of the layout/deployment of the network, i.e., exact sensor positions. Our method, on the other hand, views a specific network deployment as a particular instance (sample path) from an underlying *distribution* of sensor node layouts and sensor data rates. To compute the expected information capacity under this distribution-based viewpoint, we model parameters such as the node density, the energy density and the sensed data rate as continuous spatial functions. This continuous-space flow model is then discretized into grids and solved using a linear programming approach. Numerical studies show that this model produces very accurate results, compared to averaging over results from random instances of deployment, with significantly less computation. Moreover, we develop a robust version of the linear program, which generates robust solutions that apply not just to a specific deployment, but also to topologies that are appropriately perturbed versions.

1 Introduction

Maximizing the collective *functional lifetime* of sensor devices is clearly one of the biggest design objectives of any wireless sensor network deployment. This lifetime, and the amount of information that can be collected, depends on (a) the layout of the sensor network, (b) the initial battery capacity on the individual sensor nodes, (c) the characteristics of the sensor data generated at the individual nodes, and (d) the communication costs in transferring such generated data to a set of designated *collector* nodes. In this paper, we present a mathematical framework that accepts each of the above variables as input, and outputs an estimate of the maximum amount of sensory data that can be collected. We develop a linear programming tool that allows us to rapidly compute the lifetime of a sensor network. Unlike previous work, our linear program is formulated on the basis of *probabilistic spatial distributions* on the battery capacity, node layout and data generation rate. Such a formulation allows the linear program to be used as an efficient and fast design tool to study how changes to these network parameters affect the overall functional lifetime. In particular, we use this tool to determine the optimal node distribution for sensor network layouts, and also study the properties of the traffic paths that maximize the network lifetime.

We consider a wireless sensor network that is deployed over a specific geographic area (the "field"). Nodes of the sensor network are engaged in *sensing and collecting* data from the field, and then transporting it to one or more *collectors* (which lie within a "collector field") for further processing. The operations of data sensing and data forwarding may be done continually, periodically or intermittently. Our goal is to determine limits on how long the network can last, and more importantly, how much data the network can collect. In this paper, we concentrate on maximizing the *information capacity* of the network, defined as the maximum amount of information (bits) that can be transferred from the sensing field to the collector regions until the *first* sensor node gets completely drained of its battery and dies, as well as the *lifetime* of the network, defined as the time till this first sensor death. We shall however see that attaining this information capacity effectively results *in the simultaneous death of all the sensor nodes*. Also, we shall see that, if the information generation rate is not time-varying, *maximizing the amount of information transferred is equivalent to simply maximizing the lifetime*.

Our work in this paper is inspired by that of [1], which presents a linear programming approach for computing the lifetime of a specific sensor network deployment. While [1] shows how maximizing the network lifetime is equivalent to a linear flow maximization problem, its problem formulation is based on the *precise location* of the individual sensors: *should the sensor locations or their data rate be subject to even small changes, the lifetime needs to be recomputed from scratch.* Accordingly, this linear programming model cannot handle scenarios where the network topology cannot be deterministically described. In contrast, our approach aims to determine the network lifetime, based on *probability distributions* of the node densities and the data rates over the sensing field. Indeed, in several sensor networking scenarios, the precise location of an individual node cannot be deterministically controlled. Instead, the layout may be definable only in terms of probability distributions (using metrics such as "node density"), with the actual node deployment being a *randomly generated instance* of these distributions.

As in [1], we reduce the lifetime determination problem to a fluid-based flow maximization problem, where the maximization is over routing choices or flow distributions. We essentially compute the relative usage of different paths (from a sensor node *generating* data to a collector acting as a *data sink*) for transferring the sensed data. In our study the key to developing a fluid flow model to handle distributions of nodes is to assume that these distributions are represented by spatially continuous (over the sensing and collector fields) functions. This approximation is particularly appropriate for sensor network environments that, in contrast to conventional networks, exhibit much higher node density. The continuous-space fluid flow model is eventually numerically solved through *discretization*, i.e., by breaking up the continuous field into small, but discrete, individual grids. Interestingly, our numerical results shall show that this approach not only generates very accurate estimates on lifetime and information delivered for dense sensor fields, but remains accurate even for fields with sparsely deployed sensors.

Our formulation allows the network designer to specify energy and node densities as two independent design variables, and study different combinations of these very easily, without having to specify the actual battery capacity of each node explicitly. More importantly, viewing *a particular* deployment as the sample path of an underlying probabilistic process motivates a study of two important properties of the linear program, namely its *stability* and *robustness*. In particular, we are able to derive *robust* solutions to the capacity maximization problem, i.e., flow distributions that remain close to feasible even when the location of the individual nodes is perturbed within specified tolerance range.

The remainder of the paper is organized as follows. We present the details of our formulation and the solution technique in Section 2. In Section 3 we present numerical results. Section 4 shows how we use this modeling method to obtain an optimal node distribution in the example of a linear network, and properties of the optimal routing strategy. Section 5 discusses the stability and robustness properties of the linear program. Related work is presented in Section 6 and Section 7 concludes the paper.

2 Problem Formulation

In this section we develop a fluid-flow model for maximizing the lifetime or the total information delivered/transferred by the sensor network and discuss its unique features. As in [1], we only consider *time* to elapse when a node is either actively transmitting or receiving, and thus ignore any time spent idling. Alternatively, it is *as if* all transmissions and receptions can happen concurrently. In practice, they will need to be properly scheduled (e.g, MAC) to avoid collisions, and one has to wait from time to time for its scheduled transmission. Our model focuses only on the *operational lifetime*, by assuming an ideal condition where nodes spend no power in an idle state, and by ignoring any signaling-related overhead. Our formulation also abstracts the communication overhead in terms of "communication energy per bit". While this value may vary with changes to the specific channel settings (e.g., the actual link bit rate, error correcting codes, etc.), it does not affect the overall applicability of our model.

2.1 A Continuous Model

Suppose we have a sensing field with very densely deployed sensors. At its extreme, the field may be regarded as being *continuously* filled with sensors. Accordingly, let the following continuous functions represent various network parameters as a function of the location (x, y) in the sensing field.

$\rho(x, y)$: The number of sensors per unit space (e.g., m^2) at point (x, y). For example, if N sensors are uniformly deployed over a sensing field of area A, then this density is $\rho(x, y) = N/A$, for all $(x, y) \in A$ by using A to denote both the size and the range of the area.

$i(x,y)$: The information rate density, the amount of information (e.g., number of bits) generated per second per unit space at point (x,y). For example, if every sensor is generating b number of bits/sec, then $i(x,y) = b \cdot \rho(x,y)$.

$e(x,y)$: The initial energy density, or the amount of energy (e.g., joule) present in the beginning per unit space at point (x,y). Suppose in the beginning every sensor carries e joules, then $e(x,y) = e \cdot \rho(x,y)$. We assume that batteries are not rechargeable.

Given these definitions we have the following identities: $\int\int_A \rho(x,y)dxdy = N$, $\int\int_A e(x,y)dxdy = E$, and $\int\int_A i(x,y)dxdy = B$, where N is the total number of sensors in the field, B is the total number of bits generated per second by the field, and E is the total amount of energy available in the beginning. The above definitions can also be generalized to time-dependent parameters.

For simplicity of notation, we will use σ to represent a point in a two-dimensional space, i.e., let $\sigma = (x,y)$, $\sigma' = (x', y')$, and so on. Our previous definitions can now be written as $\rho(\sigma)$, $i(\sigma)$ and $e(\sigma)$. Note that $d\sigma = dxdy$. Define also the "flows" - $f(\sigma, \sigma')$ to be the amount of data delivered/transmitted from location σ to location σ'. This value has the unit of "number of bits per unit-source-space per unit-sink-space" or equivalently "number of bits per unit-space-squared". In general, the data is transported to a collector (or base station), whose location σ_* can be either within or outside the sensing field. Let A denote the area of the sensing field (where the sensors are distributed), and C denote the area of the base station/collectors. Without loss of generality, we can assume that A and C are non-overlapping (as long as a collector/base station is not considered a sensor simultaneously). This distinction becomes trivial when the density functions are replaced by sampling functions at single points, as we will show later.

For our analysis, we do not consider time dependence. Thus, $i(\sigma)$ is only a function of the location σ, but is constant over time. All information is transmitted to the collector. Nodes eventually transmit all data received and do not keep any of the data by the time the network lifetime ends. We then have the following formulation ($\mathbf{P}$) for maximizing the total information transferred from A to C:

$$\max_f \; t \cdot \int_{\sigma \in A} i(\sigma) d\sigma \quad \sim \quad \max_f \; t \tag{1}$$

$$\text{S.t.} \int_{\sigma' \in A} f(\sigma, \sigma') d\sigma' + \int_{\sigma' \in C} f(\sigma, \sigma') d\sigma' = \int_{\sigma' \in A} f(\sigma', \sigma) d\sigma' + i(\sigma) \cdot t, \quad \forall \sigma \in A \tag{2}$$

$$\int_{\sigma' \in A} f(\sigma, \sigma') p_{tx}(\sigma, \sigma') d\sigma' + \int_{\sigma' \in C} f(\sigma, \sigma') p_{tx}(\sigma, \sigma') d\sigma' + \int_{\sigma' \in A} f(\sigma', \sigma) p_{rx} d\sigma' + t \cdot \epsilon_s(\sigma, i(\sigma)) \leq e(\sigma), \quad \forall \sigma \in A \tag{3}$$

$$f(\sigma, \sigma') \geq 0, \quad \forall \sigma, \sigma' \in A \cup C \tag{4}$$

$$f(\sigma, \sigma') = 0, \quad \forall \sigma = \sigma' \tag{5}$$

$$f(\sigma, \sigma') = 0, \quad \forall \sigma \in C, \forall \sigma' \in A \; . \tag{6}$$

The equivalency ($\sim$) in (1) is due to the fact that $i(\sigma)$'s are time-invariant and given. The first constraint (2) is a statement of *flow conservation*, i.e., over the lifetime of a sensor, the total amount transmitted must equal the total amount received plus total amount generated/sensed. (3) is the *energy constraint*, i.e., the total energy consumed by a sensor, including transmission, reception, and sensing, cannot exceed the initial energy equipment; (4) is the non-negativity constraint; (5) states that any sensor should not transmit to itself. and (6) means that data does not flow from the collector *back* to the sensors. (In a practical scenario there might be broadcasts from the collector to the nodes. However, as long as we assume that the collector is not energy constrained, then this model remains valid as one that concentrates on the delivery of the sensed data.) Here $p_{tx}(\sigma, \sigma')$ is the *energy* dissipation instead of *power* dissipation, in transmitting from location σ to σ', in J/bit. p_{rx} is the energy dissipation in receiving. $\epsilon(\cdot)$ is the energy spent per unit time in sensing, and $e(\cdot)$ is the initial energy.

The formulation ($\mathbf{P}$) is equivalent to a more generic "max-min" formulation that allows nodes to have arbitrarily different lifetimes t, and that maximizes the minimum of these arbitrary lifetimes.

Some important points of the above model should be noted. Implicitly $i(\sigma) \geq 0$ and $\int_{\sigma \in A} i(\sigma) d\sigma > 0$ are assumed to ensure that the optimization does not become trivial. In (2) the conservation principle is expressed in terms of rate, i.e., in terms of bits per unit space rather than bits. The actual conservation comes by considering the inflow/outflow over the infinitesimal area $d\sigma = \; dx \; dy$, which gives

$$\left(\int_{\sigma^{'}\in A} f(\sigma,\sigma^{'})d\sigma^{'}\right)d\sigma + \left(\int_{\sigma^{'}\in C} f(\sigma,\sigma^{'})d\sigma^{'}\right)d\sigma = \left(\int_{\sigma^{'}\in A} f(\sigma^{'},\sigma)d\sigma^{'}\right)d\sigma + i(\sigma)d\sigma\cdot t, \tag{7}$$

where the three integrands can be written in terms of a point $(x,y)\in\sigma$ as $\int_{\sigma^{'}\in A} f(x,y,\sigma^{'})$, $\int_{\sigma^{'}\in C} f(x,y,\sigma^{'})$, and $\int_{\sigma^{'}\in A} f(\sigma^{'},x,y)$, respectively, by using the "intermediate value theorem"[3]. In essence, σ in function $f(\cdot)$ refers to a single point, but the conservation principle refers to the infinitesimal area around that point. Since $d\sigma$ cancels out on all terms in (7), we get (2).

The total amount of information delivered to the collector is $\int_{\sigma\in A}\int_{\sigma^{'}\in C} f(\sigma,\sigma^{'})d\sigma d\sigma^{'}$. Note that, in this model,

$$\int_{\sigma\in A}\int_{\sigma^{'}\in C} f(\sigma,\sigma^{'})d\sigma d\sigma^{'} = \int_{\sigma\in A} i(\sigma)d\sigma\cdot t, \tag{8}$$

by taking one more integral over $\sigma\in A$ on both sides of (2). Thus, the *objective of maximizing lifetime is equivalent to maximizing total amount of data delivered.* As a matter of fact, we can completely eliminate t from the formulation by replacing t in formulation (**P**) with the equivalent relationship defined by (8). The optimization problem (**P**) can thus be simplified to a maximization on a set of arbitrary non-negative flow variables $f(\cdot,\cdot)$. This linear program will be denoted by (**P1**). For the rest of our discussion we will concentrate on formulation (**P1**) rather than (**P**). Accordingly, we shall focus on directly maximizing the information capacity, rather than the indirect *lifetime* variable. This formulation presented above is essentially a generalization of the formulation in [1]. To see this, note that if we know the precise location of each sensor, then the continuous functions become impulse ($\delta()$) functions sampling at these particular locations, and the integrals reduce to summation at the sampled locations (identical to [1]).

It's worth mentioning that, if the network functionality involves in-network data compression/aggregation, this fluid flow based model will need to be modified, since such compression/aggregation violates the flow conservation constraint. The optimization (P) does apply to the scenario where distributed data compression [2], e.g., of the Slepian-Wolf type, is used. In [3] we used a similar math programming approach (via a non-linear program) to jointly design the allocation of data rate for each sensor node and the routing/flow pattern to maximize the lifetime of the network.

2.2 Solution Approach – Discretization

The formulation (**P1**) is in itself intractable, since it is an infinite-dimensional optimization problem due to the continuous and integral nature of its elements. An immediate thought is to solve the discretized version of this formulation. This corresponds to dividing the sensing field into grids of equal or variable sizes. This inevitably introduces error. However, if the sensing field is very densely populated, then, with relatively high resolution grids, we expect the discretization to produce reasonably accurate results. Accordingly, let the sensing field be partitioned into M non-overlapping areas/zones, indexed by $m, m = 1,2,\cdots,M$. each of size A_m. That is $A_m\cap A_n=\phi$ for $m\neq n$, and $A_1\cup\cdots A_M = A$. Again we will abuse the notation and let A_m indicate both the size and area itself. Then the original objective function becomes

$$\max_f \int_{\sigma\in A}\int_{\sigma^{'}\in C} f(\sigma,\sigma^{'})d\sigma d\sigma^{'} = \int_{\sigma\in A} f(\sigma,\sigma_C)Cd\sigma = \sum_{m=0}^{M}\int_{\sigma\in A_m} f(\sigma,\sigma_C)Cd\sigma = \sum_{m=0}^{M} f(\sigma_m,\sigma_C)A_mC,$$

where σ_m is some location within area A_m, and σ_C is some location within C. The first and third equalities are due to the theorem of intermediate value since $f()$ is a continuous function over the two-dimensional space.

The two constraints can be discretized in a similar way. For example, we obtain the following discretized version of the flow conservation constraint and energy constraint:

$$\sum_{k\in M} f(\sigma_m^1,\sigma_k^1)A_k + f(\sigma_m^2,\sigma_C(m))C = \sum_{k\in M} f(\sigma_k^2,\sigma_m^3)A_k + i(\sigma_m^4)t \quad \forall\, m\in M \tag{9}$$

and

$$\sum_{k\in M,k\neq m} f(m,k)p_{tx}(m,k)A_k + f(m,C)p_{tx}(m,C)C + \sum_{k\in M,k\neq m} f(k,m)p_{rx}A_k + t\cdot\epsilon_s(m,i(m)) \leq e(m)\cdot A_m \quad \forall\, m\in M, \tag{10}$$

[3] It is while applying the intermediate value theorem that we require the functions $i(\sigma)$ and $e(\sigma)$ to be continuous in constraints (2) and (3) respectively.

where $e(m)$ is the approximated nominal energy density within area m, and $\sigma_m^i, i = 1, \cdots 4$ are points within area A_m.

This discretization has the effect of creating a regular or possibly irregular grid/partition in the field. Moreover, for every region A_m in the field, its entire information and energy mass is assumed to be concentrated at a single point within A_m. In essence, to estimate the average information capacity for a network with a certain node distribution pattern, this method assumes that nodes are deployed at certain grid-points.

2.3 Choice of Grid Points

We next discuss possible choices in selecting the discrete points for the computational grid. To explain the choices simply, let us consider a network deployed along a straight line segment $[0, L]$. Let X be the n-element random vector denoting the location of n nodes, such that the i^{th} element is a random variable representing the location of the i^{th} node on this line segment (nodes are ordered in increasing distance). For each realization of X, denoted by x, which represents a specific deployment of the network, let $C(x)$ denote the information capacity of the deployment. That is, $C(x)$ is the objective function value obtained via a linear program constructed using node positions specified by this realization x; given the precise node locations, $C(x)$ can be computed using the discretized version of (**P1**) using x as the grid points. Note that $C(x)$ is a continuous function in x due to the linear programming nature of the problem, as long as there exists at least one feasible solution to (**P1**), which is clearly true in our case. We thus obtain $C(X)$ as a function of the random vector X. However, note that $C(X)$ is not, in general, a convex function of X.

Let $p_X(x)$ be the pdf of the deployment. This pdf is n-dimensional. It can be shown to be continuous in all n dimensions provided that the deployment of each node is done following a continuous distribution. The quantity we are interested in is the average capacity obtainable given a deployment distribution, i.e., $E[C(X)]$. If we use linear programs constructed using specific deployment layouts, then this value can only be approximated by averaging over many realizations of the deployment. On the other hand, we have

$$E[C(X)] = \int_{[0,L]^n} C(x)p_X(x)dx = C(x_o)p(x_o), \tag{11}$$

where x_o is some random vector on $[0, L]^n$, and the second equality is due to the intermediate value theorem and the fact that both $C(x)$ and $p_X(x)$ are continuous. *Thus, there is at least one deployment (or sample path) whose computed capacity equals the expected capacity over all possible deployments.*

The precise value of x_o is typically not obtainable, as $C(x)$ is a complicated function of x (our numerical results have shown that it does not possess properties like monotonicity or convexity). The discretization shown in the previous subsection essentially highlights that suitably selecting such a vector $x^{'}$ (along with its dimensionality) might lead to an approximation of this average, i.e., $C(x^{'})p_X(x^{'}) \approx C(x_o)p_X(x_o)$. This is done by first partitioning the network area into regions and then selecting a point in each area (the number of regions corresponds to the dimension of vector $x^{'}$, and a point corresponds to an element in $x^{'}$).

In our numerical experiments presented in the next section, we will use two ways of selecting this vector (or constructing the grid), denoted by **G1** and **G2**, respectively. Under **G1**, the network is first divided into rectangles that contain equal amount of energy, and then the center of each rectangle is selected. In particular, in the case of a uniform node distribution, G1 divides the network into n equal-sized regions and selects the center point of each region as the point of concentration for all the energy and information of the nodes in that region. Under **G2**, $x^{''} = E[X]$ is used to approximate x_o. This approach selects n node positions corresponding to the expected values from the distribution $p_X(x)$; the expected values of these n nodes can be derived from the node density $\rho()$ and a desired level of granularity (or number of grids) n. Details on computing G2 are given in [4].

3 Numerical Experiments

The main purpose of our numerical experiments is to examine whether our model can provide accurate results, and whether such results are sensitive to changes in a range of parameters as well as the granularity of the discretization. Therefore, almost all results presented in this section compare the result obtained by using the discretized version of optimization (**P1**) to that obtained by averaging over 100 randomly chosen instances (sample paths) of actual node deployment. For fairness, the the number of

nodes in the sensing field, the location of the collector, as well as the size of the field will be kept the same for each pair of comparisons. The total energy will be kept at $E = 1$ joule.

We adopt the following energy model. Total energy consumed by a sensor in transmission is $E_t(r) = (e_t + e_d r^\alpha)b$, where e_t and e_d are specifications of the transceivers used by the nodes, r is the transmission distance, b is the number of bits sent, and α depends on the characteristics of the channel and will assumed to be time invariant. Energy consumed in receiving is $E_r = e_r b$. Finally $E_s = e_s b$ is the energy spent in sensing/processing data that is quantized and encoded into b bits. Again e_r depends on the transceivers. In this section we will use the following parameter values taken from [1]: $e_t = 45\text{x}10^{-9}$, $e_r = 135\text{x}10^{-9}$, and $e_s = 50\text{x}10^{-9}$, all in J/bit, and $e_d = 10\text{x}10^{-12}$ in J/bit-meter$^\alpha$. α ranges between 2 and 4. As stated earlier, we will ignore idling energy in this model.

Note that (**P1**) only seeks to maximize the amount of data delivered till the first sensor node dies, and equivalently, the time till the first death. However, in all the experiments we run, the residual energy of all sensors are virtually zero when the maximum is achieved (due to space limit we are not able to show the results here). This essentially means that the maximum is achieved by balancing the lifetime of each individual sensor– when the first sensor dies, there is virtually no energy left in any of the other sensors, either. Therefore, in effect (**P1**) maximizes the total amount of data delivered till all sensors die.

3.1 Varying Grid Size

Suppose 225 nodes are uniformly distributed over a square field of size 1000×1000 square-meters (lower left corner at $(0, 0)$). The collector is located at (500,-1000). The result obtained by averaging over 100 random deployments is 46615 with [46292 , 46938] as its 95% confidence interval. The maximum and minimum values among these random samples are 49577 and 43593. All results are in bits. From these results, we see that the information capacity of a particular instance of node deployment can be almost 10% lower or higher than the mean value. We solve (**P1**) considering both discretization approaches G1 and G2. For the rest of our discussion we will use the term "number of grids" to indicate the number of regions in a grid partition. As the number of grids increases, the computation is done on an increasingly finer granularity. The left side of Figure 1 shows the complete results for G1 and G2, both for $\alpha = 2$.

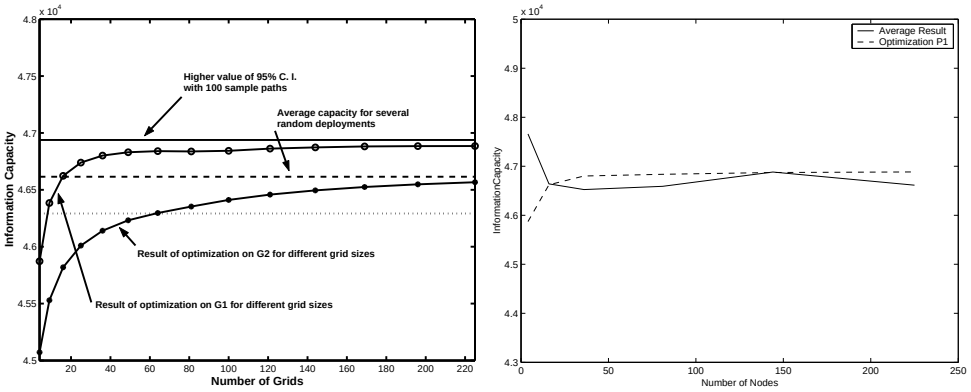

Figure 1. Information capacity (in bits) with varying number of grids (left) and Information capacity (in bits) with varying the number of sensor nodes (right)

The above results first of all showed very good accuracy of our model, with most results within the 95% confidence interval. Secondly, we see that the coarser grained computation (with fewer number of grids) also generates accurate results in many cases. This suggests that we could obtain sufficiently accurate results with very few number of grids (as few as 16 on G1 and as few as 64 on G2). As the number of grids increases, the estimate given by our model also increases[4]. This is because as the number of grids increases, the sensors of an actual deployment are closer to the point where energy and information are concentrated in the grid. Note that the values for G1 are larger than the values for G2. This is because although G1 is much simpler to create, it actually represents the expected deployment of a slightly larger field closer to the collector than the actual field. Therefore G1 provides a slight overestimate.

[4] We did not evaluate partitions with more than 225 grids since we only have 225 nodes in the field.

All results are obtained in Matlab. The results for G1 and G2 are obtained in a matter of seconds or minutes (the finer the grid the longer it takes to solve the optimization problem). On the other hand, we need hours of computation to average the results of 100 samples. Thus our model can indeed serve as a very powerful computational tool.

3.2 Insensitivity to Other Parameters

We examined the robustness of our model (distinct from the robustness of the linear program itself, an issue we shall discuss later) by varying a range of parameters. As a representative result, we report on results obtained by varying the size of the field. We considered fields of sizes 10×10, and 1000×1000, as shown in Table 1. α is set at 2, the number of nodes and grids in the partition is 225 for all cases, and the total energy of the network is again held constant. Note that for a smaller field this implies a larger information capacity since the average transmission range is also smaller. We see that in all cases the result of our optimization model closely approximates that obtained by averaging over random deployments. Again we present the results for both G1 and G2. We also examine the effect of varying

Table 1. Information capacity (in bits) with varying field size.

Field size	AVG	P1 on G1	%error (G1)	P1 on G2	% error (G2)
10^2	10138000	10137000	-0.01%	10147000	0.08%
1000^2	46615	46885	0.58%	46567	-0.10%

the number of sensor nodes in the field, while keeping the total energy constant. We vary the number of sensor nodes over a sensing field of size 1000×1000, with $\alpha = 2$. Note that such a change does not affect the results from our model (**P1**), since (**P1**) only relies on the node distribution, energy distribution, and the granularity of the partition. Having different number of physical nodes results in the same discretized version of (**P1**) in this case. The comparison results for G1 are shown in Figure 1 on the right side. In each case the result is compared to the result of (**P1**) using the same number of grids as the number of nodes. ¿From these results we see that a change in the number of nodes does not affect our results as long as the number of nodes is not too small (below 16 in this case). When the number of nodes is very small, e.g., 4, the error between our method and averaging over random deployments increases. This is due to the stability condition of the corresponding LP. As the number of nodes decreases, the "distance" between any two randomly generated deployments also increases. Thus the LP based on the regular grid deployment represents a very large perturbation.

4 Application of this Work

In this section we illustrate some of the potential uses of **P1**, via the example of a "linear" network where sensor nodes as well as the collector are lined up on a straight line. Note that although our method can also be easily applied to two-dimensional or even three-dimensional networks, we choose the linear network for the simplicity in representation and interpretation.

First we discuss the use of **P1** as a tool to determine the optimal node distribution that maximizes the network lifetime. While most reported work on sensor networks assumes a uniform node distribution, it is intuitively clear that the network would last longer if we place more nodes closer to the collector, the point of traffic concentration. As an experiment, we consider the following family of exponential node distribution densities:

$$f_X(x) = cx^a, \quad 0 \le x \le D, \quad a \ge 0 \ , \tag{12}$$

where x is the location on the line segment $[0, D]$ with the collector located at $D + d, d > 0$, c is a normalizing constant (function of a), and a is a variable. As a increases, more nodes are being deployed closer to the collector. For simplicity, the term "optimal distribution" will refer to the best distribution among this family of distributions.

Using **P1** we can calculate the capacity bounds for different distributions generated by varying the value of a, while fixing $D = 1000$, $d = 10$, and using 100 grids for the computation. Results are shown in Figure 2. We see that the maximum amount of data is delivered when a is approximately 0.7677. *The size of a grid is chosen such that each grid contains an equal amount of probability mass of the node*

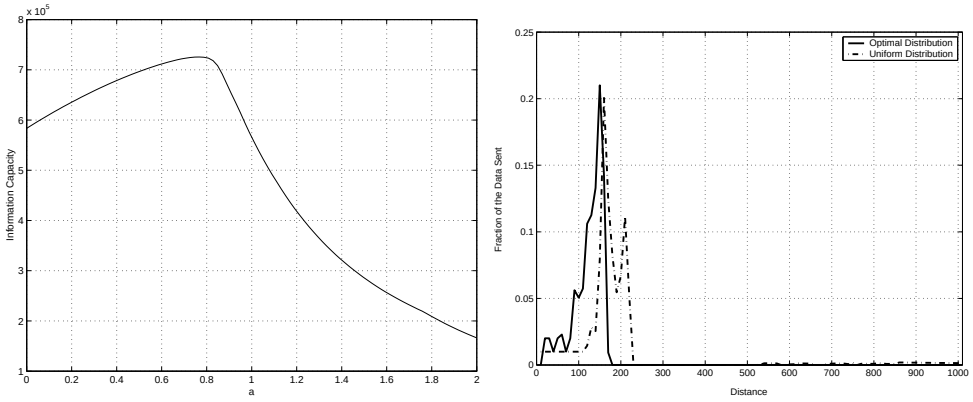

Figure 2. Varying node distributions (left) and Fraction of the data sent vs. distance of one hop transmission (right)

distribution. In other words, each grid contains an equal number of nodes and possesses equal initial energy.

We have also used **P1** as a tool to examine the characteristics of flow patterns that achieve the maximal lifetime bound. By studying the optimal flow allocation patterns, we can obtain useful insights into the characteristics that an optimal (or close to optimal) routing strategy should possess. One interesting result from our studies is that the maximal lifetime is achieved when a significant portion of the traffic is transmitted over moderately large distances. From Figure 2 we see that there is a clear "distance threshold" beyond which data rarely travels in one hop. This threshold seems to be around 220 meters in the uniform case. This result indicates that a node's transmission range can be limited to a certain level and not have a big effect on how much data can be delivered.

5 Stability and Robustness

In previous sections we showed the accuracy of our model and some potential uses of this method. In this section we consider stability and robustness issues associated with the linear program and further illustrate the utility of our approach.

For ease of explanation consider a grid (discretized) version of a network of certain random node distribution, and a network of a specific realization of the random distribution, represented by vectors $x^{'}$ and x, respectively. Denote the linear program constructed for the former LP1 and for the latter LP2. Note that under LP1, nodes are taken to be located at positions derived via G1, and under LP2, nodes are located at positions specified by the particular realization. Thus LP2 can be viewed as a *perturbed* version of LP1, known as the *nominal* version. In previous sections we have used LP1 to approximate the average of many instances of LP2.

There are two general questions related to the nominal linear program LP1 and its perturbed version LP2. The first one is how much does the optimal objective value (i.e., the information capacity in our case) to LP2 differ from that to LP1, referred to as the *stability* property of LP1. The second one is whether or not optimal solutions (i.e., the optimal flow patterns) to LP1 remain feasible (even if they may not be optimal for LP2) under the constraints of LP2, referred to as the *robustness* property of LP1. These questions are of great interest to us for the following reasons. Stability directly points to whether solving LP1 provides a good approximation on information capacity (or lifetime), and whether we would be able to bound the approximation error. Robustness on the other hand addresses the issue of whether solutions obtained from LP1 are of practical value since any real deployment is going to be a perturbed version of the grid upon which LP1 is built.

The average error in using LP1, using the previous linear network example, is as follows

$$\bar{e} = \int_{[0,L]^n} (C(x^{'}) - C(x)) p_X(x) dx.$$

where $C(x^{'})$ is the objective value resulting from LP1 and $C(x)$ is from LP2. The difference $C(x^{'}) - C(x)$ can be bounded using known results from [5], but the bounds are functions of the solutions to the dual

of LP1 and LP2. Thus the evaluation of this error is inevitably numerical as it requires solving the dual problem of LP1 and LP2. Seeking an analytical estimate of this error is an active aspect of our on-going research.

Robustness addresses whether or not solutions to LP1 (in terms of the routing flow pattern) remain feasible under a perturbed version LP2. In particular, a robust solution to LP1 is one that is feasible under LP1 and only violates any constraint under LP2 by a small tolerance δ when the locations of nodes under LP2 (x) are within a bounded range of that under LP1 ($x^{'}$). The problem of robustness translates into whether solutions obtained via the grid based computation can be implemented in a random layout, which is of great practical interest. In other words, because of the uncertainty in the actual node locations, we may be more interested in a robust solution that is feasible under both LP1 and LP2, rather than a solution that is optimal for LP1 but may not be feasible for LP2. We are also interested in the difference in the objective function value achieved using a robust solution and that using the optimal solution. The objective function value of LP2 under the robust solution can be obtained as follows. Once we have a robust solution to LP1, we can take the flow allocations specified by this robust solution and define a "robust routing strategy" that allocates the flow from any node to other nodes in proportion to the corresponding flow allocations of this robust solution to LP1. We can then apply this robust routing strategy (with the same proportional flow allocation algorithm) to LP2 and determine the maximal *proportion-preserving information capacity*, i.e., the information capacity under LP2 that can be achieved with the added constraint that the flow allocations on different links (f_{ij}) are *in the same proportion as the allocations under the robust solution to LP1.*

Note that we need to be able to bound the difference (element by element in terms of the location vector) between the node locations in LP1 and the node locations in LP2. Once this bound is known we can proceed to use the robust optimization theory developed in [6–8]. A brief sketch of the optimization is shown below. The uncertainty in the location of the nodes translates into uncertainty in the coefficients of the constraint matrix of LP1. The constraints of LP1 can be written as: $\mathbf{Qy} \leq \mathbf{b}$, where $\mathbf{y}$ is the vector composed of all the flows $f_{i,j}$, and $\mathbf{b} = \{b_i\}$. The uncertainty of the coefficients of $\mathbf{Q}$ is unknown but bounded. Since it is bounded, one can determine ϵ such that the true value of an uncertain coefficient $q_{i,j}$ is in the range $[q_{i,j} - \epsilon|q_{i,j}|, q_{i,j} + \epsilon|q_{i,j}|]$. Then the i^{th} constraint must be met with an error of at most $\delta\max[1, |b_i|]$, where δ is the infeasibility tolerance.

We now first seek a solution $\mathbf{y}$ that will be feasible for LP1 and will violate any constraint in LP2 by at most the tolerance δ. This solution is obtained by adding extra constraints to LP1. Let J_i be the set of uncertain coefficients, then the new constraints are of the form:

$$\sum_j q_{i,j}y_j + \epsilon \sum_{j \in J_i} |q_{i,j}||y_j| \leq b_i + \delta\max[1, |b_i|] \quad \forall i \tag{13}$$

Table 2 shows the results obtained with a uniform node distribution on a linear network topology with 50 nodes and different values of ϵ (the maximum allowed deviation in the node positions). It shows the objective function value (i.e. information capacity) achieved under the optimal solution to the nominal version LP1; achieved under the robust solution to LP1; and achieved under a perturbed LP2 (by following the proportional flow allocation specified by the robust solution to LP1 as described earlier). We can see that when the allowed perturbations are small ($\epsilon \approx 1\%$), the objective function value obtained by the robust solution (which violates the feasibility constraints by $\delta = 5\%$) is practically the same as the one obtained with the optimal solution. As we allow for bigger perturbations (ϵ between 10% and 25%) the robust solution becomes sub-optimal, which is the price we pay for the robustness obtained. However, we see that the information capacity achieved using this robust solution to LP1 serves as a very accurate estimate for that obtainable under LP2 even when the perturbation is large. Hence the robust solution, while suboptimal, provides an accurate estimate on the information capacity attainable under different realizations of the sensor network layout.

Table 2. Information capacity using optimal and robust solutions

ϵ	nominal result	robust result (LP1)	robust result (LP2)	diff. between the robust results
1%	573,750	573,750	568,230	-0.96%
10%	573,750	561, 420	534,680	-4.76%
25%	573,750	511,540	487,180	-4.76%

6 Related Work

Several different approaches have been used to measure or quantify the lifetime of sensor network deployments. As already stated, the fluid-flow based technique used in [1, 9] is closest to our approach–in contrast to our emphasis on finding capacity bounds that are representative of different actual network deployments realized from a common underlying distribution, [1, 9] considers lifetime bounds for a specific instance of sensor network deployment. [1] further determined lifetime bounds in the presence of a) traffic aggregation (where intermediate nodes would compress the incoming data), and b) multiple locations (where a sensor node could be located at multiple discrete points with different probabilities). In a related problem, a similar linear program is used in [10] to determine how routing should be done in order to increase network lifetime. [11] studies the impact of interference in a model similar to the ones mentioned above.

The power consumption of specific sensor network technologies and deployments has also been studied in [12, 13]. However, these studies are not concerned with the computation of theoretical bounds. Instead, they focus on novel algorithms and protocols for reducing the routing-related energy overhead in sensor networks. For example, [13] proposed LEACH, a clustering protocol that uses data aggregation over a hierarchical topology to reduce the power consumed by individual sensor nodes. Alternatively, the lifetime of specific sensor network topologies has also been studied using hybrid automata modeling [14]. In [14], a model-based simulator is used to determine the variation in network lifetime with changing distances between the sensing nodes and the collector node.

7 Conclusion

This paper presented a modeling methodology that drastically reduces the time needed for determining the expected information capacity of a data-gathering wireless sensor network. By developing a robust generalization of our method, we are able to obtain objective function values that are more representative, in that different network layouts that are "perturbations of one another" can achieve capacity and lifetime bounds that are very close to this robust capacity bounds. We conducted various numerical experiments, under a variety of parameters. We showed that results generated under this formulation are quite insensitive to the change in a range of parameters, including field size, grid size and the attenuation parameter α.

References

1. M. Bhardwaj and A. P. Chandrakasan, "Bounding the lifetime of sensor networks via optimal role assignments," in *IEEE INFOCOM*, 2002.
2. S. S. Pradhan and K. Ramchandran, "Distributed source coding: Symmetric rates and applications to sensor networks," in *IEEE Data Compression Conference (DCC)*, Snowbird, Utah, March 2000.
3. E. J. Duarte-Melo, M. Liu, and A. Misra, "A computational approach to the joint design of distributed data compression and data dissemination in a field-gathering wireless sensor network," in *Forty-First Annual Allerton Conference on Communication, Control, and Computing*, October 2003.
4. E. J. Duarte-Melo, M. Liu, and A. Misra, "An efficient and robust computational framework for studying lifetime and information capacity in sensor networks," Tech. Rep. CSPL-348, University of Michigan.
5. K. G. Murty, *Linear Programming*, John Wiley and Sons, 1983.
6. A. Ben-Tal and A. Nemirovski, "Stable truss topology design via semidefinite programming," *SIAM Journal on Optimization*, vol. 7, 1997.
7. A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Mathematics of Operations Research*, vol. 23, 1998.
8. A. Ben-Tal and A. Nemirovski, "Robust solutions to uncertain linear programs," *OR Letters*, vol. 25, 1999.
9. M. Bhardwaj, T. Garnett, and A. P. Chandrakasan, "Upper bounds on the lifetime of sensor networks," in *IEEE ICC*, 2001.
10. J. Chang and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *IEEE INFOCOM*, 2000.
11. K. Jain, J. Padhye, V.N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Procceedings of the Ninth Annual International Conference on Mobile Computing and Networking (MobiCom)*, San Diego, California, U.S.A., September 2003.
12. S. Lindsey and C. Raghavendra, "Pegasis: Power efficient gathering in sensor information systems," in *IEEE Aerospace Conference*, March 2002.
13. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy efficient communications protocols for wireless microsensor networks," in *Hawaii International Conference on System Sciences (HICSS '00)*, January 2000.
14. S. Coleri, M. Ergen, and T. Koo, "Lifetime analysis of a sensor network with hybrid automata modeling," in *1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, September 2002.