Mobile Networks and Applications 10, 811–824, 2005
© 2005 Springer Science + Business Media, Inc. Manufactured in The Netherlands.
DOI: 10.1007/s11036-005-4440-x

An Efficient and Robust Computational Framework for Studying Lifetime and Information Capacity in Sensor Networks*

ENRIQUE J. DUARTE-MELO and MINGYAN LIU
Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor

ARCHAN MISRA
T. J. Watson Research Center at IBM, NY

Published online: 24 October 2005

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. This is especially important for a network designer studying the fundamental lifetime limit of a family of network layouts, since the lifetime of specific network deployment instances may differ appreciably. As an example of this model's use, we determine the optimal node distribution for a linear network and study the properties of optimal routing that maximizes the lifetime of the network.

Keywords: mathematical programming, linear program, optimization, system design, wireless sensor networks, lifetime, capacity, sensor deployment, node distribution, optimal routing, fluid flow model, robustness, stability

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, depend 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 has the effective of balancing energy consumption across the network, and in all the scenarios we considered it results in the simultaneous death of all sensor nodes (although in general, maximizing the time till the first death and maximizing the time till the death of all sensors are distinct objectives). Also,

*This work is partially supported by NSF ANI-0112801 and ANI-0238035.

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 Bhardwaj and Chandrakasan [4], which presents a linear programming approach for computing the lifetime of a specific sensor network deployment. While Bhardwaj and Chandrakasan [4] 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. For example, in many exploration and battlefield scenarios, nodes may simply be dropped remotely (from an airplane or a ship) over a physical area. The network operator may then be able to control only the coarse spatial density of the deployment (how many nodes per unit area are deployed in different portions of the field).

As in Bhardwaj and Chandrakasan [4], 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. As the same overall flow rates may be realized by an infinite number of routing strategies, our goal is not to determine *a* precise routing strategy, but to study the optimal relative usage of different paths by *any* well-designed routing strategy. As proved in Bhardwaj and Chandrakasan [4], any solution as a result of this optimization is, however, *always realizable* in practice. 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. Highly dense sensor fields allow us to treat individual sensor nodes as *fungible* and study the aggregate properties of groups of nodes. Of course, 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.

Once the grids are decided, the flow maximization problem becomes similar to that in Bhardwaj and Chandrakasan [4]. However, the use of independent continuous-valued functions for the node or energy densities in our model provides important flexibilities unavailable in Bhardwaj and Chandrakasan [4]. 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 individual nodes is perturbed within a specified tolerance bound. While the objective function value (lifetime or information capacity) under the robust version of the linear programming may be sub-optimal, it serves as an accurate and conservative estimate on the obtainable information capacity for any particular network layout. In contrast, Bhardwaj and Chandrakasan [4] does not provide a way to ascertain if the lifetime bound from a specific deployment is indeed representative of the information capacity achievable under various "slightly modified" deployments.

The remainder of the paper is organized as follows. We present the details of our formulation, the solution technique, and a critique of this modeling framework, in Section 2. In Section 3 we present numerical results of our model under various parameter settings. Section 4 shows how we use this modeling method to obtain an optimal node distribution that maximizes the information capacity in the example of a linear network, and also study properties of the optimal routing strategy. Section 5 discusses the stability and robustness properties of the linear program, and demonstrates how the use of additional constraints allows us to derive maximal flow values that remain feasible under perturbations to the actual nodal layout. Related work is presented in Sections 6 and 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 Bhardwaj and Chandrakasan [4], 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. Note that, in practice, medium access schemes that prevent collisions, such as TDMA, will have to be used in order to approach the lifetime bound obtained here.

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 functions 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 $(x, y) \in A$, where A denotes 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 bits/sec, then $i(x, y) = b \cdot \rho(x, y)$. If the information rate varies with time then the information rate density becomes *i(x,y,t)*. For simplicity, in the rest of this paper we will limit our attention to the case where the information rate does not change over time.

$e(x, y)$: 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 "flow", denoted by $f(\sigma, \sigma')$, 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 also not a sensor simultaneously). This distinction becomes trivial when the density functions are replaced by sampling functions at single points, as we will show later. 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 (**P**) for maximizing the total information transferred from A to C:

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

$$\text{S.t.} \int_{\sigma'\in A} f(\sigma, \sigma')d\sigma' + \int_{\sigma'\in C} f(\sigma, \sigma')d\sigma'$$

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

$$\int_{\sigma'\in A} f(\sigma, \sigma')p_{tx}(\sigma, \sigma')d\sigma' + \int_{\sigma'\in C} f(\sigma, \sigma')p_{tx}(\sigma, \sigma')d\sigma'$$

$$+ \int_{\sigma'\in A} f(\sigma', \sigma)p_{rx}d\sigma' + t\cdot\epsilon_s(\sigma, i(\sigma))$$

$$\leq e(\sigma), \quad \forall\sigma \in A \tag{3}$$

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

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

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

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. This necessarily precludes broadcast. However note that the use of broadcast does not increase the network lifetime for the purpose of data gathering. This is because under our model flow is optimally distributed to different neighboring nodes. Therefore disseminating the same information to multiple nodes does not help; it will only increase energy consumption in packet reception. (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. $\epsilon_{\mathrm{s}}(\cdot)$ is the energy spent per unit time in sensing, and $e(\cdot)$ is the initial energy.

The formulation (**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) 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.[1] 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.

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, $m = 1, 2, \ldots, M$, each of size A_m. That is, $A_m \cap A_n = \phi$ for $m \neq n$, and$A_1 \cup \ldots A_M = A$. Note that if there are multiple collectors/base stations with non-deterministic locations but known distribution, then a similar partition of C can be created and the same framework would apply. For simplicity, in the following we will assume there is a single collector. Again we will let A_m indicate both the size and area itself. Then the original objective function becomes

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

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:

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

where σ_m^i, $i = 1,\ldots,4$ are points within area A_m. Furthermore, if all A_k are small enough, $f(\sigma_m^i,\sigma_m^k)$ can be approximated as constant throughout the areas A_m and A_k, and be simply written as $f(m,k)$. Using similar notations, we obtain the following discretized version of the energy constraints.

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

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 ith element is a random variable representing the location of the ith node on this line segment (nodes are ordered in increasing distance from point 0). 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

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

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

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

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 illustrates one way of selecting such a vector x' (along with its dimensionality) that might lead to an approximation of this average, i.e., $C(x')p_X(x') \approx C(x_o)p_X(x_o)$, 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'). Note that this type of discretization still leaves the choices of regions and points unspecified. In our numerical experiments presented in the next section, we will use this type of grid construction, denoted by **G1**, and specify how we select the regions and points. 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.

One could also use $x'' = E[X]$ 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 *ρ()* and a desired level of granularity (or number of grids) *n*. This type of selection of representative node locations is denoted as **G2** and will be studied in the next section as well.

Not surprisingly, G1 and G2 result in the selection of different points, and the computation of different values of the information capacity. (See Appendix A for a derivation of x'' for a uniform node distribution in a linear network.) For example, if $L = 1$ and $n = 4$, then (G1) over a uniform node distribution selects the vector

$X = [0.125, 0.375, 0.625, 0.875]$, while (G2) selects the location vector $X = [0.2, 0.4, 0.6, 0.8]$. Unfortunately, for a general deployment and distribution, G2 does not have a closed form solution. Thus, except for the simple case of the uniform node distribution, we shall only use G1 for most of our numerical experiments.

2.4. Discussion and critique of the model

The formulation presented above is essentially a generalization of the formulation in Bhardwaj and Chandrakasan [4]. To see this, note that if we know the precise location of each sensor, then the continuous functions become impulse (δ) functions sampling at these particular locations, and the integrals reduce to summation at the sampled locations (identical to Bhardwaj and Chandrakasan [4]). In general, our formulation produces a much more powerful result by computing the average capacity of the network for a *distribution* of the deployment (expressed by the density function) rather than for a particular deployment. By modeling sensor deployment as a continuous function, (**P1**) provides a way to study the variation in the capacity with different node density functions (deployment patterns).

On the other hand, our model also assumes that $f(\cdot,\cdot)$, $p_{tx}(\cdot,\cdot)$, etc. are continuous functions of position (*x*, *y*) (to apply the intermediate value theorem). For real-life scenarios, where parameters such as p_{tx} are typically discontinuous and have a few discrete values, our model will clearly introduce error. In addition, our grid solution approach involves coarse or fine-grained approximation. The utility of this method thus depends on the significance of this approximation error.

It's worth mentioning that this fluid flow based model needs to be modified to take into account in-network data compression/aggregation that violates the flow conservation constraint. It does apply to the scenario when distributed data compression [13], e.g., of the Slepian-Wolf type, is used. This is because by using such distributed data compression each sensor can adopt a rate to encode its data, and encoded data from different sensors may be treated as independent (i.e., no further compression is necessary). In Duarte-Melo et al. [8] 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.

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 is 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 [4]: $e_t = 45 \times 10^{-9}$, $e_r = 135 \times 10^{-9}$, and $e_s = 50 \times 10^{-9}$, all in J/bit, and $e_d = 10 \times 10^{-12}$ in J/bit-meter$^\alpha$. α ranges between 2 and 4. As stated earlier, we will ignore idling energy in this model.

In all the cases where a uniform distribution is used, 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.

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 results obtained by averaging over 100 random deployments (AVG) are shown in Table 1. We included the 95% confidence interval (C.I.) as well as the maximum and minimum values among these random samples, all 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. Table 2 lists selected results from this experiment by varying the size/number of grids, and Figure 1 shows the complete results for G1 and G2, both for $\alpha = 2$. In Table 2, percentage of error is calculated as the ratio of the difference between G1 and the average (given in Table 1) and the average.

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.[2] 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.

Table 1
Average information capacity (in bits) from 100 random deployments.

α	AVG	95% C.I.	Min	Max
2	46615	[46292, 46938]	43593	49577

Table 2
Information capacity (in bits) with varying number of grids ($\alpha = 2$).

# of Grids	P1 on G1	% error (G1)	P1 on G2	% error (G2)
225	46885	0.58	46567	−0.10
196	46884	0.58%	46548	−0.14
144	46873	0.55%	46495	−0.26
100	46843	0.49%	46411	−0.44
64	46840	0.48%	46296	−0.68
36	46801	0.40%	46141	−1.02
16	46623	0.02	45819	−1.71
9	46384	−0.49	45529	−2.33
4	45872	−1.60	45072	−3.31

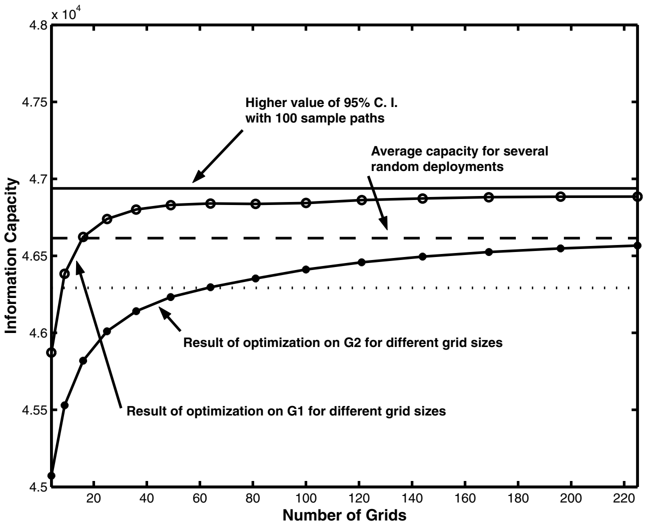

Figure 1. Information capacity (in bits) with varying number of grids ($\alpha = 2$).

All results are obtained in Matlab. Results in Table 2 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, to obtain the results in Table 1 we need hours of computation. Thus our model can indeed serve as a very powerful computational tool.

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

Table 3
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

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 first 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 3. 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 a smaller field 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 now examine the effect of varying 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 $\alpha = 2$. 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 2. In each case the result is compared to the result of (**P1**) using the same number of grids as the number of nodes.

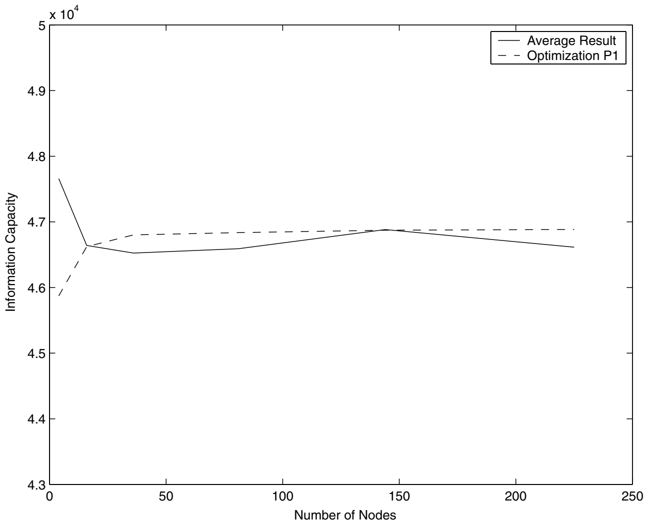

Figure 2. Information capacity (in bits) with varying the number of sensor nodes.

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.

3.3. Non-uniform node distributions

All previous results employ a uniform node distribution across the sensing field. In this subsection we will examine different node distributions while fixing the total number of nodes at 225. Specifically we will consider the linear sloped node distribution shown in Figure 3, where the node density linearly increases over the field as it gets closer to the collector. To obtain a sequence of node distributions, we vary the slope by changing the length of the line segments $\overline{AB}$ and $\overline{CD}$ shown in Figure 3. The uniform distribution is a special case with $\overline{AB} = \overline{CD} = 1 \cdot c$, where c is a normalizing constant. For simplicity we will assume that the per-node energy and information generation rate remain constant. Accordingly, the energy density and information generation functions are non-uniform (linearly scaled versions of the node distribution) as well.

Under such non-uniform distributions, the discretization of (**P1**), G1, is done by partitioning the field into differentially-sized rectangles, such that the total energy in each rectangle grid is identical. The results for G1 are shown in Table 4 by using 225 such rectangle grids. Again our model produces very accurate estimates. It should be noted that the increasing capacity with the increase in the slope is to be expected since in this case more bits are generated at locations closer to the collector.

In general this framework will work with any distribution, as long as the proper grid is selected. In Section 4.1 we will consider a few more examples.

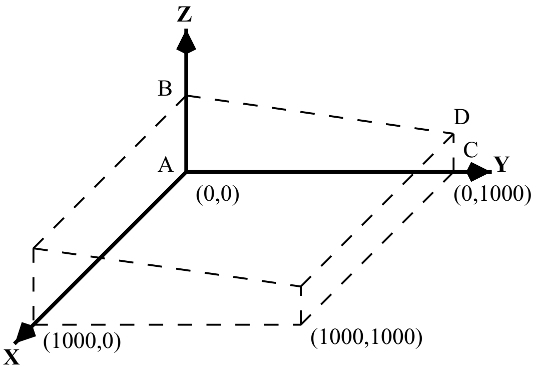

Figure 3. "Sloped" distribution.

Table 4
Information capacity (in bits) with varying node distribution.

$(\overline{AB}, \overline{CD})$	P1 on G1	AVG	% error
($2c$,0)	57162	57322	−0.28
($1.75c$, $0.25c$)	54602	54769	−0.3
($1.5c$, $0.5c$)	52013	52215	−0.38
($1.25c$, $0.75c$)	49431	49424	0.014
($1c$, $1c$)	46885	46615	0.58

3.4. Residual energy

Figure 4 shows the amount of residual energy of every node in the two-dimensional network we have been using, with 225 nodes in a 1000 × 1000 network, using (**P1**). 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, as we can see from this result, the maximal is actually 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. Note that the residual energy values are not exactly zero due to numerical tolerance used when solving the problem in Matlab. This result is easy to understand when we note that, if some of the sensors do indeed have residual energy when the first node dies, then the lifetime of the critical dying node can be increased by shifting its forwarding burden to other nodes, thereby violating the assumption of maximal lifetime. While this observation holds in many practical cases of interest, it is not axiomatically true in general. Consider, for example, a topology where all nodes are located very close to the collector, except for a single node located very far away. If all nodes generate data at the same rate, there is no routing technique that can compensate for the higher energy drain on the far-away node, which will always die before the others.

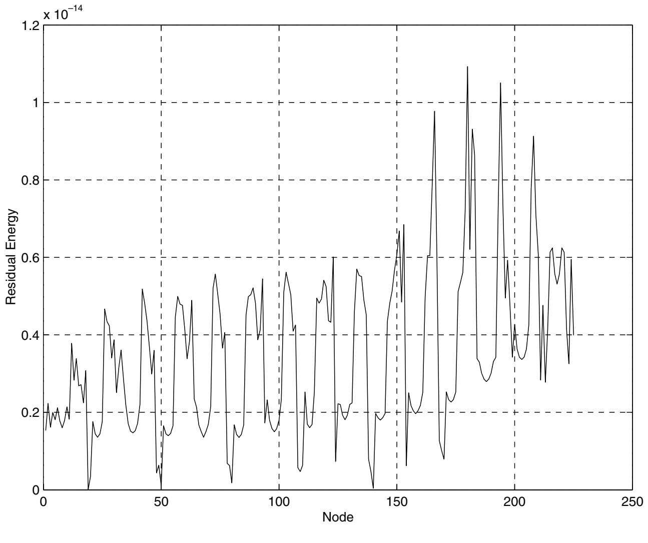

Figure 4. Residual node energy (in J).

4. Optimal node distributions and flow patterns

In this section we use (**P1**) to investigate the effect node distribution has on the network lifetime and capacity. We will also examine the optimal flow allocation patterns to obtain insights into the properties required of a well-designed routing protocol. We emphasize that our model provides a rapid computational tool that facilitates the investigation of alternative node distributions. In this section we will solely focus on a "linear" network where sensor nodes as well as the collector are lined up on a straight line, although our method can be equally applied to two-dimensional or three-dimensional networks. This is because working in one dimension makes the result much easier to represent and interpret. The grid used throughout the section will be G1.

Our network consists of a line segment of length D between 0 and D, on which sensor nodes are distributed. Our computation will be based on dividing the line segment into a constant number (M) of grids. The collector is located at $D + \frac{D}{M}$, a distance D/M away from one end of the line segment.

4.1. Optimal node distribution

One question of obvious interest is what would be an optimal node distribution that could maximize the network lifetime as well as total amount of data deliverable. While most reported work on sensor networks assumes a uniform node distribution, it is intuitively clear that the network should last longer if we place more nodes closer to the collector, the point of traffic concentration. Following this, even though there are infinitely many types of possible distributions, we will only consider the following family of exponential node distributions:

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

where x is the location on the line segment, c is a normalizing constant, and a will be varied to obtain different distributions. Note that $a = 0$ and $c = 1/D$ gives the uniform distribution. As a increases, more nodes are being deployed closer to the collector. For simplicity, below the term "optimal distribution" will refer to the best distribution within this family of distributions.

For each a value, we create $M = 100$ grids and set $D = 1000$. We will assume that the field has a constant information distribution across the network. Therefore if we place more nodes closer to the collector, then nodes closer to the collector will generate less data while nodes far away will generate more. Each node will also contain equal amount of initial energy. *The size of a grid is chosen such that each grid contains an equal amount of probability mass of the node distribution*. In other words, each grid contains an equal number of nodes and possesses equal initial energy.

Figure 5 shows the amount of data delivered by the network for different values of a. Here we have used two different energy models for comparison purposes. "Energy Model A" refers to the model presented and used in the previous section, while "Energy Model B" is the same model with a different

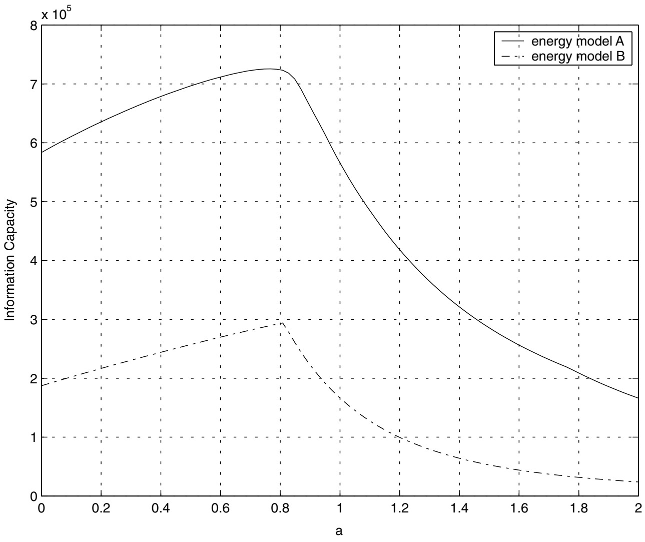

Figure 5. Information capacity (in bits) with varying node distributions.

set of parameters taken from [10]: $e_t = 50 \times 10^{-9}$, $e_r = 50 \times 10^{-9}$ both in J/bit, and $e_d = 100 \times 10^{-12}$ in J/bit-m^2. In this section α is set to 2. We see that the maximum amount of data is delivered when a is approximately 0.7677 in the case of model A and approximately 0.8 in the case of model B. Note that $a = 0$ corresponds to a uniform distribution and $a = 1$ corresponds to a linear density. With a uniform distribution, nodes closer to the collector spend too much energy relaying data for other nodes; for a very high value of a, nodes far away spend too much energy because they are generating the majority of the data. The maximizing node distribution essentially achieves the optimal balance between these two effects.

It's interesting that except for the scale, the two models generate strikingly similar results. They show that the uniform distribution can result in a capacity loss of about 18% and 35% from the optimal value for the two models, respectively. We also see that, once a exceeds 1, causing most nodes to be deployed too close to the collector, the results deteriorate quickly.

4.2. Optimal flow patterns and routing implications

Our linear programming framework essentially computes the optimal flows between any two points σ and σ'. 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. For example, if it turns out that most of the flow mass is concentrated over small-distance links, it follows that the routing algorithm should prefer a larger number of small-distance hops over a small number of larger-distance hops. Since the routing pattern obtained from our computation is specific to the particular grid used, we focus on *extracting* the essential principles of the optimal traffic paths.

We consider two measures of the optimal flow patterns. The first is the fraction of data transmitted over a given distance (as a function of the distance), i.e., a histogram of the flow mass vs. the link distances. The second is the average transmission distance of a node (averaged over the total amount of data transmitted by that node), as a function of its distance to the collector. Both these measures attempt to reveal, in some sense, the (average) size of a transmission hop and the proportionate use of hops of different distances.

We will determine these two characteristics with the optimal node distribution ($a = 0.7677$) and the uniform distribution ($a = 0$), both under energy model A. Figure 6 shows the fraction of data transmitted over a given distance for both cases. For a given distance d, this fraction is defined as $\frac{1}{M}\sum_{i=1}^{M}\frac{\text{flow}_i^d}{\text{flow}_i^T}$, where M is the total number of grids in the partition, $\text{flow}_\text{i}^\text{d}$ is the total flow out of node i that is transmitted over a distance d and $\text{flow}_\text{i}^\text{T}$ is the total flow out of node i. In the uniform distribution case, where the transmission distance has discrete values (multiples of the distance of a single hop), the curve is generated by connecting values at these discrete points. In the non-uniform distribution case, where each node has a different set of possible transmission distances, we combine data transmitted over 10 meter segments, and represent it as a single point. For example, the value at 300 meters represents all transmissions over 290 to 300 meters.

From Figure 6 we see that there is a clear "distance threshold" beyond which data rarely travels in one hop. This threshold seems to be around 180 meters in the optimal case, and 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. In addition, the majority of data is transmitted over distances between 100 and 180 meters in the optimal case, and between 140 and 220 meters in the uniform case. Note that nodes closer to the collector do not have the option of transmitting over large distances. So in that sense this result is somewhat biased towards smaller distances.

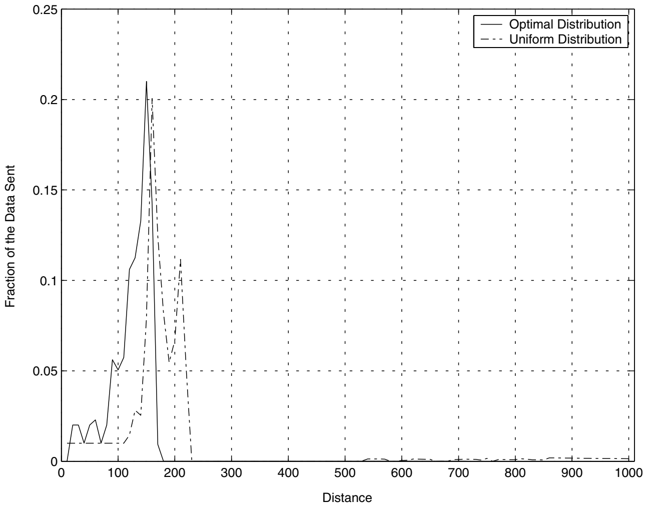

Figure 6. Fraction of data vs. distance.

Perhaps most importantly, this figure shows that a significant portion of the flow mass is transported over medium-sized hops (in the uniform case, the distance between neighboring grids is 10 meters), and not directly to the nearest neighbor. Various proposals for minimum-energy routing in multi-hop wireless networks (e.g., [9]), on the other hand, prefer a large number of small distance hops to minimize the communication energy. Such protocols are often designed for ad-hoc networks, where all nodes are equally likely to be both sources and sinks of data. In contrast, for sensor network environments where most sensor nodes are either sources or relays, it appears that using direct transmissions to more distant neighbors is preferable, since it reduces the forwarding burden on intermediate nodes. We also see that the optimal node distribution has a smaller range than the uniform case. For the optimal node distribution, nodes further away from the collector have more data to transmit (the information generation rate is constant over the entire field) than in the uniform distribution, and accordingly prefer smaller distance hops to reduce their transmission costs.

Figure 7 shows the average distance a node transmits its data as a function of its distance from the collector. The node closest to the collector will always send directly to it and therefore its average distance is always its distance to the collector. As nodes are placed further away, we see that their average distance continuous to increase, although not as fast as their distance to the collector. This shows an increased reliance on having their data relayed by other downstream nodes. Finally, we see that for nodes far away from the collector, their average distance decreases under the optimal distribution but continues to increase under the uniform distribtion. This shows that under the optimal distribution such nodes transmit most of their data to increasingly closer relaying nodes. Since regions farther from the collector have a smaller node density, the corresponding nodes have higher data generation rates. Accordingly, they conserve their transmission energy by choosing shorter hops, which is less of a need under the uniform distribution. Note also that the curve for the optimal case is near perfectly smooth: small changes in node location only create small changes in the average transmission distance. *This suggests that near-optimal routing might be constructed using location information.*

4.3. Limited transmission range

In the previous subsections, we have allowed the maximum transmission range of each node to be unbounded. We now consider the impact of specifying a maximum limit on the transmission range. Since the previous studies show that the bulk of the data flow occurs over intermediate hop distance, we expect that limiting this transmission range to moderate values should not significantly reduce the information capacity.

To consider a bounded transmission range, we add the following extra constraint to (**P1**): $f(\sigma, \sigma') \cdot [d(\sigma, \sigma') - r]^+ = 0$ for all σ, σ', where $[x]^+$ takes value x or 0, whichever is greater. $d(\sigma, \sigma')$ denotes the distance between one location/grid σ and the other σ', and r denotes the maximum range of transmission allowed.

Figure 8 shows the total amount of data that could be delivered to the collector for different maximal transmission ranges. We see in the case of optimal node distribution, relaxing the transmission range constraint to be beyond 150 meters (the "knee") makes virtually no difference. In the uniform case, the total amount of data does continuously increase as the transmission range limit is relaxed. However, this increase significantly slows down beyond 200 meters (the "knee"). In each case, an increase in the transmission range limit results in significant gain in information capacity when the range is below the "knee".

For the optimal distribution, the smallest transmission range shown is 60 meters—any lower value causes the nodes

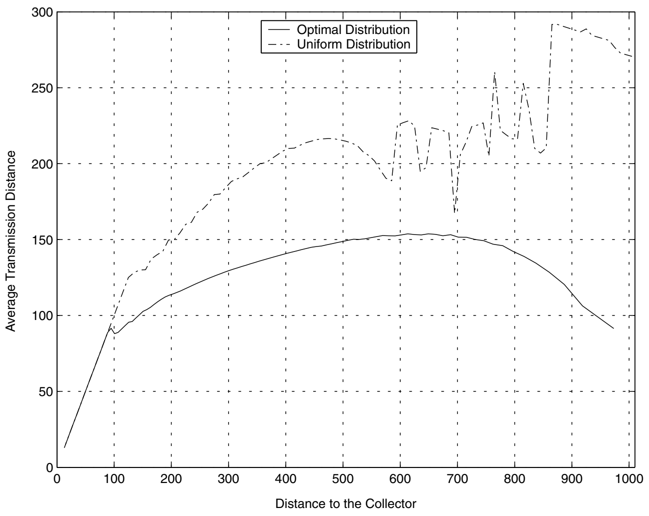

Figure 7. Average transmission distance.

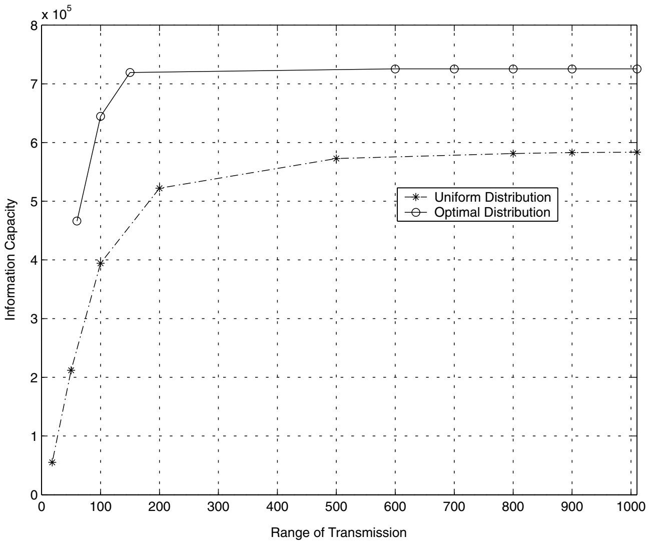

Figure 8. Data delivered vs. limit of transmission range.

farthest from the collector to become disconnected. Similarly, for the uniform distribution, the smallest range shown (18 meters to avoid division by zero in Matlab) is the minimum needed to maintain connectivity. These results indicate that sensor nodes should be equipped with moderately powerful radio interfaces, capable of direct transmission over reasonably large distances, to achieve close-to-optimal information capacity.

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 ([12]), 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 [1–3]. 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 ith 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 5
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%

Table 5 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 sub-optimal, provides an accurate estimate on the information capacity attainable under different realizations of the sensor network layout.

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 Bhardwaj et al. [4,5]) 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, Bhardwaj et al. [4,5] considers lifetime bounds for a specific instance of sensor network deployment. Bhardwaj and Chandrakasan [4] 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 Chang and Tassiulas [6] to determine how routing should be done in order to increase network lifetime.

The power consumption of specific sensor network technologies and deployments has also been studied in Lindsey and Raghavendra [11] and Heinzelman et al. [10]. 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, Heinzelman et al. [10]) 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 [7]. In Coleri et al. [7], 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. There were three main objectives of our modeling methodology. The first goal was to build a rapid computational tool that could be used in many sensor network scenarios, where the exact node locations are not determinable in advance, but are instead a *sample realization* of an underlying probabilistic node distribution strategy. With this framework we are able to derive the expected lifetime and information capacity of any *distribution* of sensor nodes rather than just particular sample paths of the node deployment. Secondly, by treating the node distribution, the sensing rate and the energy capacity as continuous-time spatial distributions that are direct inputs to the linear program, our framework allows designers to easily play around and study how different network topologies and resource allocation strategies affect the maximum realizable information capacity. Finally, by developing robust generalizations of the linear program, and by studying the LP's stability properties, 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 bound.

We conducted various numerical experiments, using 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. To demonstrate the utility of our distribution-oriented framework, we then derived the optimal node distribution (among a specified family of power-law distributions) for a linear network, i.e., the node distribution that maximizes the network lifetime and the total amount of data delivered. Our numerical studies also show that, to maximize the network lifetime, sensor nodes must transmit a significant fraction of their packets directly to longer-distance neighbors (using medium-range radios), rather than simply forwarding it to their nearest downstream neighbor.

In future work, we plan to use our linear programming tool to further study how various parameters of dense sensor networks impact its operational lifetime. In ongoing research, we are studying the tradeoff between the fidelity of sensing (proportional to the actual number of deployed sensors) of the field, and the total functional lifetime of the sensor network.

Appendix A: Expected deployment

In this paper we have described the grid formation G2 as being the case where the location of the points of energy and information concentration matches the expected location for the nodes of a randomly deployed network. For the sake of detail we provide in this appendix the example of a linear network with M nodes.

Consider a network of M nodes to be deployed in a line of length 1. The distance from the beginning of the line to the position of each node, X_i, are assumed to be iid with pdf $f_X(x)$ and cdf $F_X(x)$.

We will order the location of the nodes by their distance to the beginning of the line. That is, $Z_1 = \min(X_1, X_2, \ldots, X_M)$ is the first node in the line, Z_m is the mth node in the line, and Z_{M} is the last node in the line. Then the position of the points where energy and information are concentrated in G2 is $E[Z_1], E[Z_2], \ldots, E[Z_M]$. Thus we proceed to determine these values.

Let us begin with $E[Z_{\mathrm{M}}]$.

$$F_{Z_M}(z) = \Pr(\max(X_1, X_2, \ldots, X_M) \le z)) \tag{14}$$

$$= \Pr(X_1 \le z, X_2 \le z, \ldots, X_M \le z) \tag{15}$$

$$= \Pr(X_1 \le z)\Pr(X_2 \le z)\ldots\Pr(X_M \le z) \tag{16}$$

$$= F_x(z)^M \tag{17}$$

$$f_{Z_M}(z) = MF_X(z)^{M-1} \cdot f_X(z). \tag{18}$$

Suppose that the network is deployed following a uniform distribution, i.e., $f_X(x) = 1$ and $F_X(x) = x$. Then,

$$f_{Z_M}(z) = Mz^{M-1} \tag{19}$$

$$E[Z_M] = \frac{M}{M+1} \tag{20}$$

Now consider $E[Z_1]$. Note that for all i, $f_{X_i}(z) = f_X$ and $F_{X_i}(z) = F_X$.

$$F_{Z_1}(z) = \Pr(\min(X_1, X_2, \ldots, X_M) \le z)) \tag{21}$$

$$= 1 - \Pr(X_1 \ge z, X_2 \ge z, \ldots, X_M \ge z) \tag{22}$$

$$= 1 - \Pr(X_1 \ge z)Pr(X_2 \ge z)\ldots\Pr(X_M \ge z) \tag{23}$$

$$= 1 - (1 - F_X)^M \tag{24}$$

For uniform distribution we get:

$$E[Z_1] = \frac{1}{M+1} \tag{25}$$

Finally for $E[Z_m]$, $m = 1, 2, \ldots, M$, we have:

$$F_{Z_m}(z) = \sum_{j=m}^{M} \binom{M}{j} F_X^j (1 - F_X)^{M-j} \tag{26}$$

For the specific case of a uniform distribution, we get

$$E[Z_m] = \frac{m}{M+1}. \tag{27}$$

References

[1] A. Ben-Tal and A. Nemirovski, Robust truss topology design via semidefinite programming, SIAM Journal on Optimization 7(4)(1997) 991–1016.

[2] A. Ben-Tal and A. Nemirovski, Robust convex optimization. Mathematics of Operations Research 23(4) (1998) 769–805.

[3] A. Ben-Tal and A. Nemirovski, Robust solutions to uncertain linear programs. Operations Research Letters 25 (1999) 1–13.

[4] M. Bhardwaj and A.P. Chandrakasan, Bounding the lifetime of sensor networks via optimal role assignments, in: *Annual Joint Conferences of the IEEE Computer and Communication Societies (INFOCOM)* New York (2002) pp. 1587–1596.

[5] M. Bhardwaj, T. Garnett and A.P. Chandrakasan, Upper bounds on the lifetime of sensor networks, in: IEEE International Conference on Communications (ICC) 3 (2001) 785–790.

[6] J. Chang and L. Tassiulas, Energy conserving routing in wireless adhoc networks, in: *Annual Joint Conferences of the IEEE Computer and Communication Societies (INFOCOM)* (Tel Aviv, Israel), Vol. 1 (2000) pp. 22–31.

[7] 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)* (Atlanta, Georgia, 2002) pp. 98–104.

[8] 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* (2003) pp. 70–79.

[9] J. Gomez and A. Campbell, Power-aware routing optimization for wireless ad hoc networks, in: *High Speed Networks Workshop (HSN)* (Balatonfured, Hungary, 2001).

[10] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, Energy efficient communications protocols for wireless microsensor networks, in: *Hawaii International Conference on System Sciences (HICSS '00)* Vol. 8 (2000) pp. 8020.

[11] S. Lindsey and C. Raghavendra, PEGASIS: Power efficient gathering in sensor information systems, *IEEE Aerospace Conference* 3 (2002) 1125–1130.

[12] K.G. Murty, *Linear Programming* (John Wiley and Sons, 1983).

[13] S.S. Pradhan and K. Ramchandran, Distributed source coding: Symmetric rates and applications to sensor networks, in: *IEEE Data Compression Conference (DCC)* (Snowbird, Utah, 2000) pp. 363–372.

Enrique J. Duarte-Melo was born in Mexico City, Mexico. He received a B.S. degree in Electrical Engineering from Instituto Tecnologico y de Estudios Superiores de Monterrey in December 1996. After his undergraduate career he worked in Schlumberger LTD and later for the National Bureau of Science, Technology and Innovation of Panama. He entered the University of Michigan in August 1999 to pursue his Ph.D. in Electrical Engineering with a major in Communications and a minor in Signal

Processing. He received his Masters Degree in December 2000 and expects to get his PhD in 2004. Research interests include Wireless Sensor Networks, Network Capacity and Energy Consumption.
E-mail: ejd@eecs.umich.edu

Mingyan Liu (M'00 / ACM'01) received her B.Sc. degree in electrical engineering in 1995 from the Nanjing University of Aero. and Astro., Nanjing, China, M.Sc. degree in systems engineering and Ph.D. Degree in electrical engineering from the University of Maryland, College Park, in 1997 and 2000, respectively. She joined the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, in September 2000, where she is currently an Assistant Professor. Her research interests are in performance modeling, analysis, energy-efficiency and resource allocation issues in wireless mobile ad hoc networks, wireless sensor networks, and terrestrial satellite hybrid networks. She is the recipient of the 2003 NSF CAREER Award.
E-mail: mingyan@eecs.umich.edu

Archan Misra (http://www.research.ibm.com/people/a/archan) is a Research Staff Member with the Pervasive Security and Networking Department at the IBM TJ Watson Research Center, Hawthorne, NY. His current research efforts are related to middleware infrastructure for Internet-scale distributed pervasive computing, information-theoretic mobility management in smart wireless environments, and MAC protocols for wireless mesh networks. He is also leading a research effort on algorithms for a middleware supporting adaptive query primitives in sensor and actuator networks. Prior to joining IBM in 2001, Archan spent 3 $\frac{1}{2}$ years at Telcordia Technologies (formerly called Bellcore), working on all-IP cellular networks and Internet QoS. He has served as a technical program committee member on several conferences, and chaired workshops and panels in the areas of wireless and mobile computing. He received the Best Paper awards in ACM WOWMOM 2002 and IEEE MILCOM 2001. Archan is the current Untethered Network Technologies chair of the IEEE Computer Society's Technical Committee on Computer Communications (TCCC). Archan received his Ph.D. in Electrical and Computer Engineering from the University of Maryland at College Park in May, 2000, and his B.Tech in Electronics and Communication Engineering from IIT Kharagpur, India in July 1993.
E-mail: archan@us.ibm.com