# Modeling a Dense Wireless Sensor Network: Complexity, Stability and Robustness

Mingyan Liu
(Joint work with Enrique J. Duarte-Melo and Archan Misra)

Electrical Engineering and Computer Science
University of Michigan, Ann Arbor

January 11, 2007

# Dense Wireless Sensor Networks

## Applications

- Environmental monitoring: large scale data collection
- Surveillance: alarm propagation, data storage and query

## Key Features

- Large in quantity; deployed in bulk
- Close proximity; often duty-cycled
- Locations often random rather than precisely controlled

## Challenges in Modeling

- Scalability, complexity, accuracy, ...

# Outline of the Talk

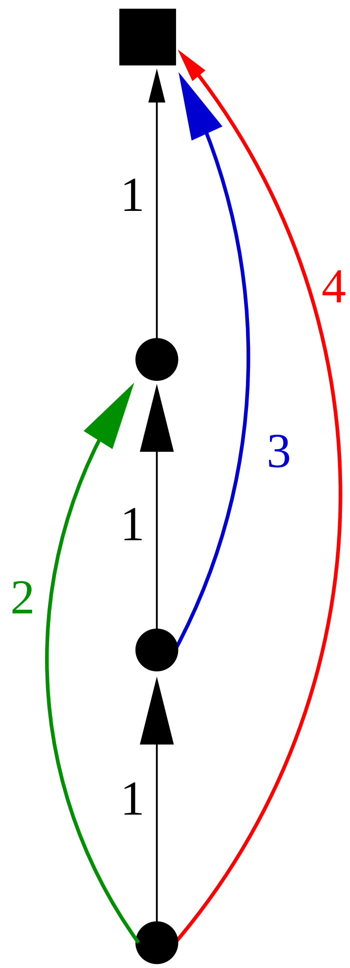
## The case of computing network lifetime

- Fluid-based flow maximization models
- Problems that arise in a dense network
- Our approach and results

## Stability and Robustness

- Robust solutions
- Practical implications

# Network Lifetime




**Key elements**

- Nodes start with fixed initial energy $E_i$ in joule
- Energy model: energy in transmission of a bit
- Data generation rate $r_i$ in bits/unit time
- Lifetime maximized over all possible routing strategies
- Flows and flow distribution $f_{i,j}$, in bits
- Reduced to fluid-based flow maximization

# Basic Assumptions

- Bits divisible
- Transmission range adjustable
- Energy model: transmission, reception and sensing/processing, no idling
- Operational lifetime: time only elapses during active tx and rx
- Communication overhead abstracted into communication energy per bit (may vary with specific physical layer setting)

# A Linear Programming Approach

$$\max \quad t \quad \text{or} \quad \sum_{i\in M} f_{i,C}$$

$$\text{s. t.} \quad \sum_{j\in M} f_{i,j} + f_{i,C} = \sum_{j\in M} f_{j,i} + r_i t$$

$$\sum_{j\in M} f_{i,j} e_{tx}^{i,j} + f_{i,C} e_{tr}^{i,C} + \sum_{j\in M} f_{j,i} e_{rx}$$

$$+ r_i t e_s \leq E_i \quad \forall i \in M$$

$$f_{i,j} \geq 0 \quad \forall i, j \in M$$

$$f_{i,i} = 0 \quad \forall i \in M$$

$$f_{C,i} = 0 \quad \forall i \in M$$

**Observation**

- Produces the optimal flow pattern as a solution
- Constructed using precise knowledge of sensor locations
- Varies from one deployment to another
- Related: [Bhardwaj and Chandrakasan 2002]

# A Continuous Model

## Motivation

- Maximization for node distributions, not particular outcomes or realizations of some distribution
- If possible, we will be able to study things like optimal node distribution

## Key idea

- Extremely densely deployed field: spatially continuous
- Continuous node density $\rho(\sigma)$ in number per unit space
- Continuous information density $i(\sigma)$ in bits per unit time per unit space
- Continuous energy density $e(\sigma)$ in joule per unit space
- Optimize over all flow allocations $f(\sigma, \sigma')$, in bits per unit space-squared

# Problem Formulation

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

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

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

$$f(\sigma, \sigma') \geq 0, \; \forall \sigma, \sigma' \in A \cup C$$
$$f(\sigma, \sigma') = 0, \; \forall \sigma = \sigma'$$
$$f(\sigma, \sigma') = 0, \; \forall \sigma \in C, \forall \sigma' \in A \,.$$

# Some Comments

- Total amount delivered to the collector is

$$\int_{\sigma \in A} \int_{\sigma' \in C} f(\sigma, \sigma') d\sigma d\sigma'.$$

- Maximizing lifetime is equivalent to maximizing total amount of data delivered:

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

- Can completely eliminate $t$ from the formulation: problem (P1)

# Solution Technique: Discretization

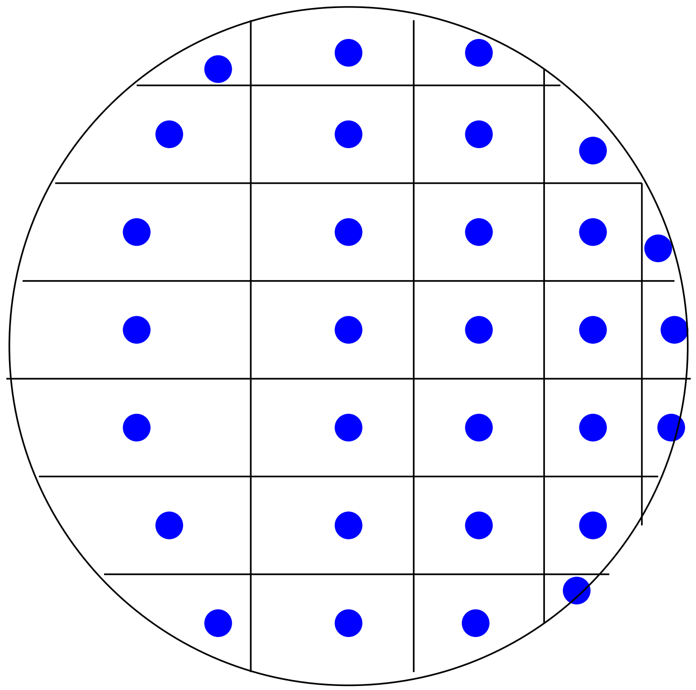
**Consider the objective function:**

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

where $\sigma_m$ ($\sigma_C$) is some location within area $A_m$ ($C$).

# Discretization

- Creating a partition of the field, with regular/irregular cells
- Energy and information concentrated on a single point in each cell
- Computation done for a network of finite points
- Can approximate using a set of grid points



# Choice of Grid Points

**Consider a linear network** $[0, L]$

- $X$: the $n$-element random vector denoting the location of $n$ sensors
- $p_X(x)$: pdf of the deployment
- $C(X)$: the objective function value, or the *capacity*
- We are interested in $E[C(X)]$

**Using linear programs constructed using specific deployment layouts, we can only approximate by averaging over many realizations of the deployment**

# On the other hand...

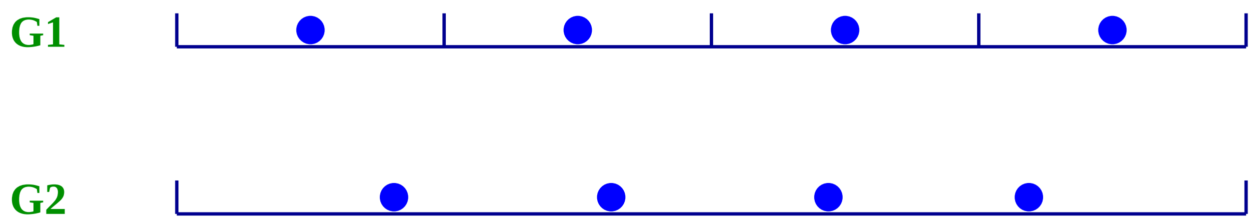
$$E[C(X)] = \int_{[0,L]^n} C(x) p_X(x) dx = C(x_o) p_X(x_o) L^n,$$

- Discretization provides one approximation for $x_o$
- It remains to determine the regions and the points – G1
- G2 – use $E[X]$ to approximate $x_o$: closed form solution in general not available

# Example

**Uniform distribution,** $L = 1$**,** $n = 4$

- G1: Equal size squares/cells $(0.125, 0.375, 0.625, 0.875)$
- G2: Expected position of nodes $(0.2, 0.4, 0.6, 0.8)$



# Discussion and Critique

- A generalization of the original fluid-flow model: continuous functions become impulse functions with known locations
- Our model computes the average capacity of the network for a *distribution* rather than for a particular deployment
- Grid solution approach leads to coarse or fine-grained approximation
- Needs to be modified to take into account in-network processing that violate flow conservation
- May be used jointly with distributed data compression by also optimizing over data rate allocation

# Numerical Experiments

- Accuracy of our method
- How sensitive it is to a range of parameters
- Compare against averages of 100 random instances of random deployment

**Energy model:**

$$e_{tr}(r) = (e_t + e_d r^{\alpha}) \quad \text{J/bit}$$

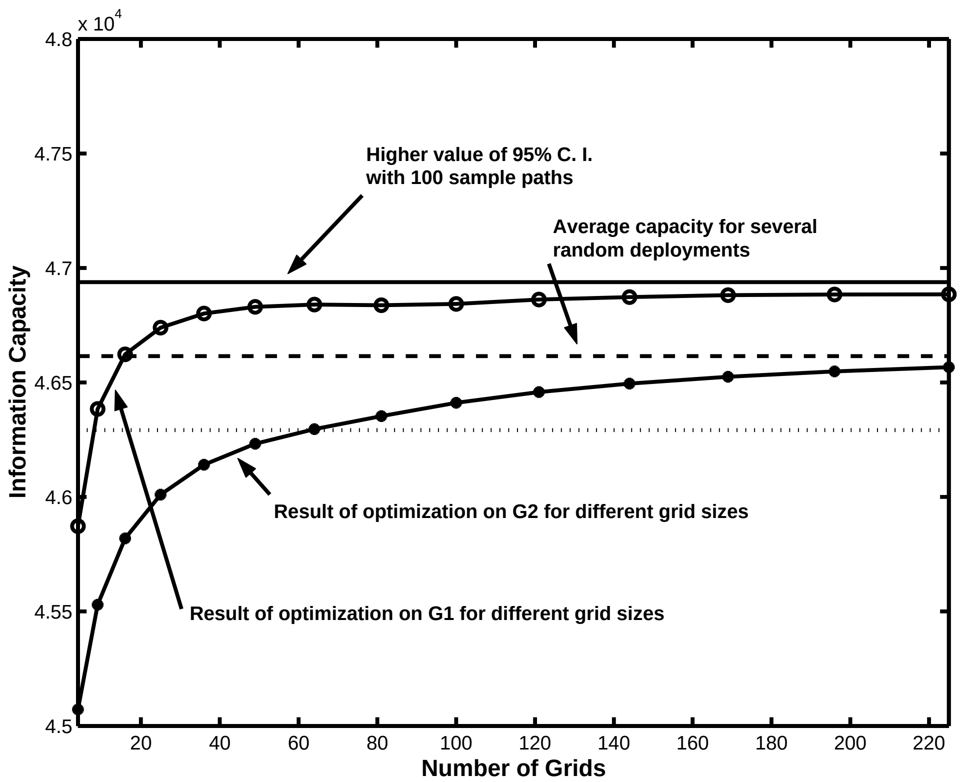
$$e_t = 45 \times 10^{-9}; \qquad e_{rx} = 135 \times 10^{-9}; \qquad e_s = 50 \times 10^{-9} \quad \text{J/bit}$$

$$e_d = 10 \times 10^{-12} \quad \text{J/bit-meter}^{\alpha}$$

# Average over 100 Random Deployment

| $\alpha$ | AVG | 95% C.I. | Min | Max |
|---|---|---|---|---|
| 2 | 46615 | [46292 , 46938] | 43593 | 49577 |

# Varying Grid Size



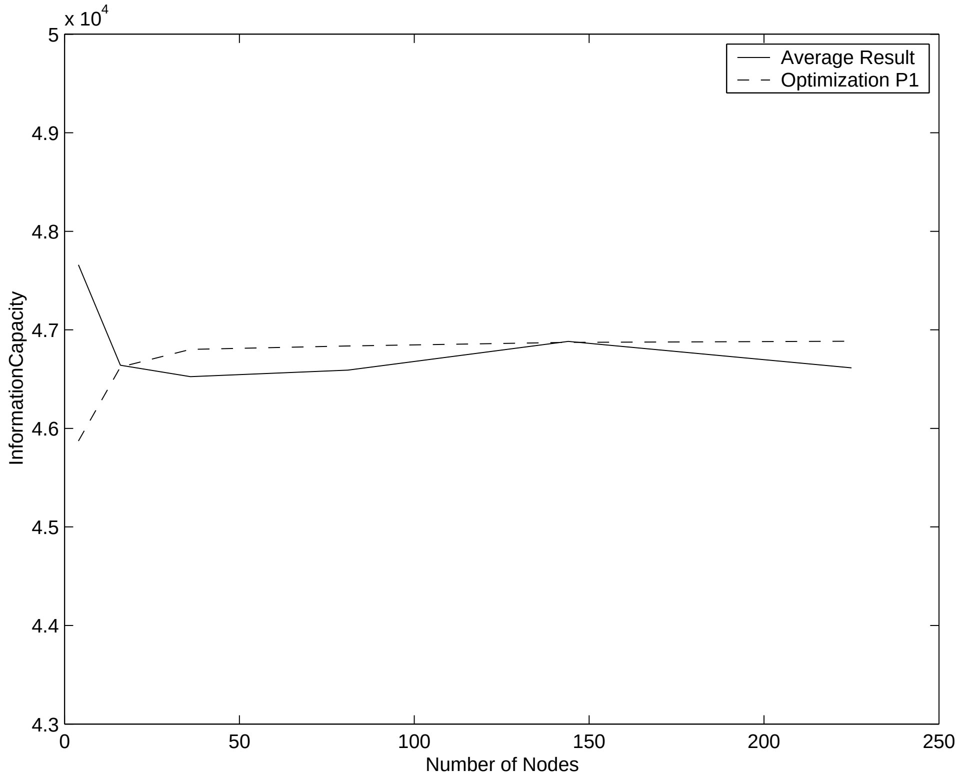

- 225 nodes uniformly distributed over $1000 \times 1000$
- Good accuracy; mostly within 95% C.I.
- Coarser-grained computation remains accurate
- Seconds vs. hours

# Varying Field Size

- 225 nodes
- Grid set to 225

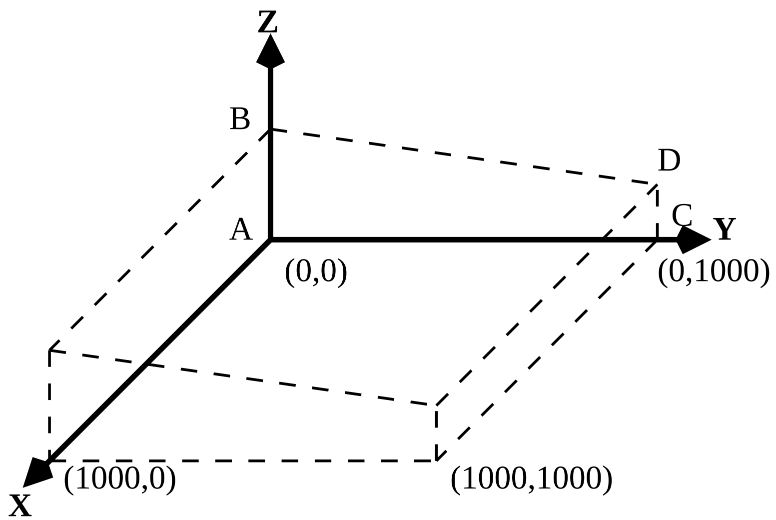
| Field size | AVG | P1 on G1 | %error (G1) | P1 on G2 | % error (G2) |
|---|---|---|---|---|---|
| $10^2$ | 10138000 | 10137000 | -0.01% | 10147000 | 0.08% |
| $1000^2$ | 46615 | 46885 | 0.58% | 46567 | -0.10% |

# Varying Number of Sensor Nodes




- Varying number of nodes does not affect computation based on G1 for a fixed granularity
- Approximation remains good
- Error large when number of nodes very small
- Good for dense networks

# Non-Uniform Node Distribution




- Linear sloped node distribution; 225 nodes
- G1: partitioning the field into differentially-sized rectangles, each with identical energy

| $(\overline{AB}, \overline{CD})$ | 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% |

# Some Applications

## Due to advantage in computation

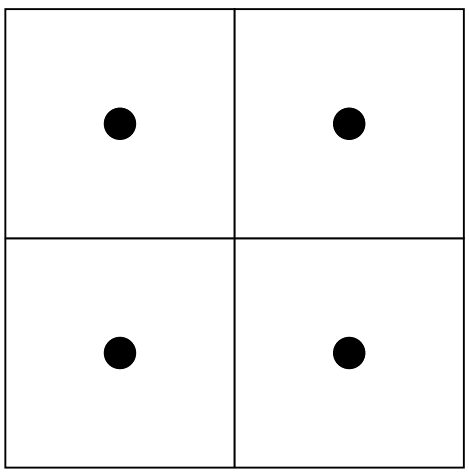
- Optimal routing pattern
- Optimal node distribution
- Joint routing and data compression

# Stability and Robustness

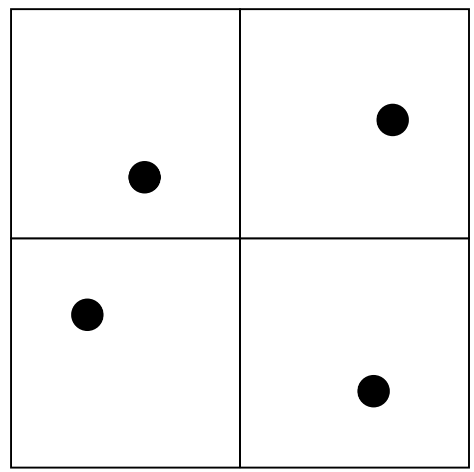
**Consider LP1 and LP2:**

- LP1 (the nominal version): LP using discretization, e.g., G1, for some distribution.
- LP2 (the perturbed version): LP using some random realization of the same distribution

**LP1**



**LP2**



# Stability and Robustness (Cont'd)

**Stability property of LP1:**

- How much does the optimal value of LP2 differ from LP1
- Whether solving LP1 provides good approximation
- Whether we would be able to bound the error

**Robustness property of LP1:**

- Whether optimal solution (flow pattern) to LP1 remain feasible under constraints of LP2
- Whether solutions obtained from LP1 are of practical value

# Average Error in Using LP1

**Using the previous linear network example:**

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

where $C(.)$ is the objective function value.

- The difference $C(x^{'}) - C(x)$ can be bounded using known results [Murty 1983]
- However, the bounds are functions of the solutions to the dual of LP1 and LP2
- We do not yet have better estimates

# Robustness

**A robust solution to LP1:**

- feasible under LP1 and
- only violates any constraint under LP2 by a small tolerance $\delta$ when $x$ is within a bounded range of $x^{'}$

**Why are we interested:**

- Robustness: whether solutions obtained via the grid based computation can be implemented in a random layout
- Because of the uncertainty in the actual node locations, we may be more interested in a robust solution rather than the optimal solution under LP1
- Also interested in the objective value difference between the two

# Robustness (Cont'd)

- Apply robust optimization theory [Ben-Tal and Nemirovski 1997-1999]: location uncertainty (perturbation $\epsilon$) $\longrightarrow$ uncertainty in the coefficients of the constraint matrix of LP1
- Seek a solution $\mathbf{y}$ that will be feasible for LP1 and will violate any constraint in LP2 by at most the tolerance $\delta$
- Obtained by adding extra constraints to LP1
- Example: linear network with 50 nodes; $\delta = 5\%$

| $\epsilon$ | nominal | robust (LP1) | robust (LP2) | diff. |
|---|---|---|---|---|
| 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% |

# Discussion and Conclusion

- Presented a fluid-flow model for dense sensor networks
- Continuous input functions
- Lifetime/Information estimates for a distribution of nodes
- Computational advantage
- Stability and robustness properties

## Discussion

- The legitimacy of studying grid networks?
- Distributed implementation?

# References


- Bhardwaj and Chandrakasan 2002: "Bounding the lifetime of sensor networks via optimal role assignments", *IEEE INFOCOM*.
- Murty 1983: *Linear Programming*, John Wiley and Sons.
- Ben-Tal and Nemirovski 1997: "Robust truss topology design via semidefinite programming", *SIAM J. Optimization*.
- Ben-Tal and Nemirovski 1998: "Robust convex optimization", *Mathematics of Operations Research*.
- Ben-Tal and Nemirovski 1998: "Robust solutions to uncertain linear programs", *Operations Research Letters*.