Mobile Networks and Applications 11, 317–326, 2006
© 2006 Springer Science + Business Media, LLC. Manufactured in The Netherlands.
DOI: 10.1007/s11036-006-5185-x

An Energy-Optimal Algorithm for Neighbor Discovery in Wireless Sensor Networks

R. MADAN*
Department of Electrical Engineering, Stanford University, Stanford CA 94305-9505, U.S.A.

S. LALL
Department of Aeronautics and Astronautics 4035, Stanford University, Stanford, CA 94305-9505, U.S.A

Published online: 4 April 2006

Abstract. We consider sensor networks in which individual nodes with on-board sensing and low-power transmitters and receivers establish connections with neighboring nodes. The overall objective is to enable energy-efficient data communication, relayed between arbitrary nodes on the network. We develop a distributed algorithm which minimizes the power required for neighbor discovery.

Initially nodes do not have deterministic knowledge of the location of their neighbors, and we model the distribution of the nodes as a two-dimensional Poisson process with known intensity. This corresponds to a situation in which a large number of nodes are randomly distributed over a given area. The process of neighbor discovery is modeled as a Markov decision process, and the resulting control policy is a finite automaton, driven by the underlying probability distribution, that minimizes the average power consumed. This policy can be computed offline and stored in each node with very low requirements for online memory and processor capability.

Keywords: stochastic processes, control theory, mathematical programming, optimization

1. Introduction

This paper describes an algorithm for network discovery in wireless sensor networks which uses minimum energy. We are motivated by a scenario in which a large number of small, low-cost sensors with on-board processing, wireless transceivers, and limited power sources are distributed in a region [14,5]. Applications of such networks include environmental monitoring and factory automation, as well as military applications such as surveillance.

We assume that the sensor nodes are *randomly* distributed. The sensors are not previously configured with knowledge of their location and must transmit wireless queries to discover surrounding nodes and establish a communication network. If the topology changes, say as a result of attrition of sensor nodes, then this process may need to be repeated more than once during the lifetime of the network. We focus on the problem of *network discovery* where nodes must find other nodes and determine the minimum energy paths by which to route data to them. This network discovery phase consumes energy, and our objective is to develop an algorithm for establishing the network with minimum energy consumption, and thereby increase the network lifetime.

We focus on the problem where the nodes are in unknown and random locations in a region in the plane. Given the positions of the nodes, one can determine which pairs of nodes can communicate directly with each other, based upon the maximum communication range available to the nodes. This range is in turn determined by the maximum transmission power and the path loss model. As pointed out in [17], if the received power falls as $1/d^n$, where d is distance from the transmitter, then if $n \geq 2$ relaying information between nodes may be more energy-efficient than direct communication over large distances.

1.1. Prior work

The underlying topology of the network defines a graph $G_{\max}$, called the *maximum connectivity graph,* in which each node represents a sensor, and two nodes are joined by an edge if they are within communication range of each other. All possible routes are then paths within the graph $G_{\max}$. We separate the next stage, that of finding desirable routes, into the following two parts.

1. *Topology Formation*: In this phase, one forms a subgraph $G_{\min}$ of $G_{\max}$, called the *minimum-energy graph* that contains all nodes of $G_{\max}$. The graph $G_{\min}$ is defined in the following way. For any two neighboring nodes u and v in $G_{\max}$, an edge is included in $G_{\min}$ if there is no other path between u and v which uses less energy for communication. Thus $G_{\min}$ contains every energy efficient path

Both authors were partially supported by the stanford URI *Architectures for Secure and Robust Distributed Infrestures*, AFOSR DoD award number 49620-01-1-0365. Ritesh Madan is also supported by a Sequoia Capital Standford Graduate Fellowship.

* Corresponding author

in the network. Edges of $G_{\max}$ which are not in $G_{\min}$ are therefore ruled out as routing candidates, thus reducing the work that the routing algorithm itself must perform.

2. *Packet Routing*: There are several possible protocols which may be used to route packets over the graph $G_{\min}$. Examples of routing protocols for ad hoc networks are *ad hoc on-demand distance vector routing* (AODV) [13], *location-aided routing* (LAR) [9], and *greedy perimeter stateless routing* (GPSR) [8].

In this paper, we focus on the first problem of topology formation. In principle, the minimum-energy graph $G_{\min}$ can be constructed from $G_{\max}$ using the Bellman-Ford algorithm. However, the Bellman-Ford algorithm is expensive in terms of energy consumption, due to the broadcasts used for exchange of routing tables. Instead, several alternative distributed algorithms using local information have been proposed to form topologies that are sparse and contain energy-efficient routes [15,17,10,20]. By first ruling out inefficient routes, it is possible to decrease the energy used, and hence increase the network lifetime, for many different routing algorithms, as discussed in [10].

The wireless sensor nodes are modeled as having knowledge of their positions, which is typically implemented using the Global Positioning System (GPS) [12]. The algorithm proposed by Rodoplu and Meng [17], and further optimized by Li and Halpern [10] forms a graph G_{enc}, called the *enclosure graph* that contains the minimum energy graph. The construction of this graph is based on the notion of the *enclosure* of a node, defined in the following way. The enclosure of node i contains all the neighbors of i to which direct transmission is more energy efficient than transmission over any path with exactly two edges.

An algorithm to approximate the minimum-energy graph is proposed in [20], which we refer to as the *cone algorithm*. Suppose a node is at the origin in the plane $\mathbb{R}^2$, and let $m > 0$ be an integer. We divide the plane into the m cones

$$C_k = \left\{ (r\cos\theta, r\sin\theta) \,\middle|\, r \geq 0, \frac{2\pi(k-1)}{m} \leq \theta \leq \frac{2\pi k}{m} \right\}$$

for $k = 1, \ldots, m$. The node at the origin is then joined by an edge to the closest node within each cone C_k. A graph G_m, called the *cone graph*, is constructed by applying this procedure to every node in $G_{\max}$. In [20], a bound for the energy cost of an optimal path between a pair of nodes u and v in the cone graph is derived in terms of the corresponding energy consumption of the optimal path between u and v in the minimum-energy graph. In [10] it is further shown that if $G_{\max}$ is connected and $m \geq 5$, then the cone-based graph is also connected.

Figure 1 shows the graphs constructed by the different topology formation algorithms for 200 nodes distributed uniformly in a square of side 10. The maximum transmission radius of each node is 2. The graph $G_{\max}$ is significantly more dense, containing many redundant edges compared to the minimum-energy graph $G_{\min}$. The topology constructed by the enclosure-based graph contains more edges than that in the minimum-energy graph, but is also significantly more sparse than $G_{\max}$. The cone graph G_6 constructed with $m = 6$ is a subgraph of $G_{\max}$ which approximates the minimum-energy graph.

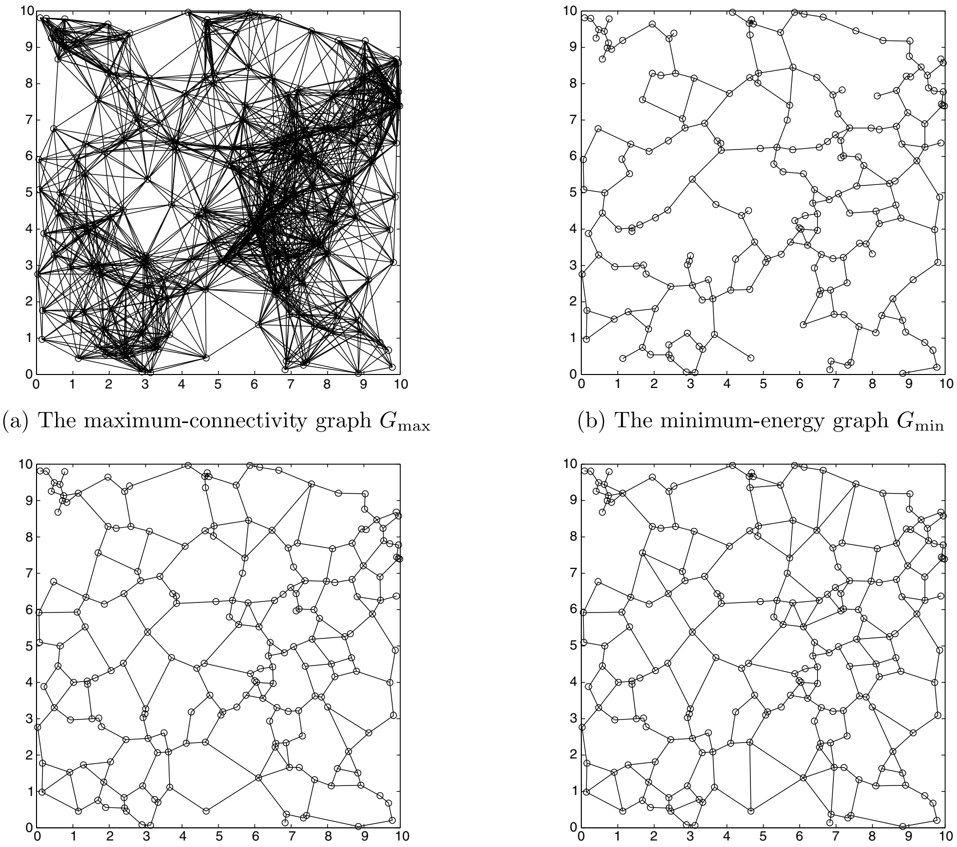
1.2. Approach

In all of the above algorithms, each node repeatedly broadcasts with increasing power to discover its neighbors, stopping either when it is *enclosed* [10] in the case of the enclosure-based algorithm, or, in the case of the cone algorithm, when a node has been found in each cone. These broadcasts used to discover neighbors are a significant source of energy consumption during topology formation. If the nodes are mobile, or the network topology changes due to node attrition and obstruction, then this topology formation process must be repeated, and the amount of energy spent in topology formation will have a significant impact on the network lifetime. We note that *neighbor discovery* in wireless ad hoc networks is a much more general problem. We use a specific notion of a neighbor (as in [20]), and then minimize the average energy consumption during neighbor discovery.

This paper focusses on minimizing the energy consumption during topology formation of an ad hoc wireless sensor network. In [10], it is pointed out that one reasonable suboptimal choice for increasing broadcast power is to repeatedly double the transmitted power until either the enclosure is found or the maximum power is reached. In order to discretize the problem, we consider minimizing the energy consumption use by a modified cone algorithm for neighbor discovery. We model the nodes as being able to detect the direction from which signals arrive. We construct a Markov decision process which models the relationship between the energy used and the probabilities of finding nodes, and show that the minimum-energy solution can be found by solving a *stochastic shortest path* problem [1]. These problems are also referred to as *pursuit* [4] or *first passage* [3] problems. They can be solved offline using both linear programming and dynamic programming. In this case we arrive at an optimal search strategy for the nodes which can be implemented as a simple finite-state automaton with very low memory and processor requirements. Such probabilistic automata are considered in [19].

1.3. Outline of the paper

The remainder of the paper is organized as follows. Section 2 describes the system model and the formulation of our objective. Section 3 models the neighbor location process as a Markov decision process and Section 4 describes a method to compute an optimal network discovery strategy. Section 5 gives simulation results for a simple scenario, and Section 6 summarizes the results and gives some directions for future work.



(a) The maximum-connectivity graph $G_{\max}$

(b) The minimum-energy graph $G_{\min}$

(c) The enclosure graph G_{enc}

(d) The cone graph G_6

Figure 1. Network topology formed by different topology control algorithms.

2. System model

2.1. Problem definition and approach

Motivated by the results in [20,11], our objective is to locate the closest node within each of the m cones $C_1, \ldots ,C_m$, if there is one that can be reached with maximum transmission power. We would like to find a strategy that minimizes the expected energy used, given the probability distribution of the nodes in the plane. We assume that the nodes are identically and independently distributed (IID) over a given region $\mathrm{D} \subset \mathbb{R}^2$.

More specifically, we model the distribution of nodes as a *two dimensional Poisson process* [18] with known intensity ρ. In practice we may have a fixed number of nodes, each uniformly distributed over D. We use a two dimensional Poisson distribution to approximate this scenario, as this simplifies some of the computations later in the paper. Note that for a Poisson distribution, the nodes in any finite region D are identically and independently distributed in a uniform manner over the region D.

We refer to the *coverage region* of a node i; this is the region in the plane in which other nodes can receive the transmissions from node i. We assume that the region D is sufficiently large that the coverage region of the node under consideration is entirely within D. This assumption does not hold in the case when nodes are close to the boundary of D, and our analysis would need modification in that case. If the percentage of nodes close to the boundary of D is small, then the overall inefficiency in power consumption due to this approximation maybe neglected.

A node may transmit at only a discrete-set of allowable powers, given by

$$Q = \{q_1, q_2, \ldots, q_l\}$$

where $q_1 < q_2 < \cdots < q_l$. We also refer to the *coverage region with power* q. This is the circular region with radius $r(q)$ within which transmissions of power q are received. We assume r is an increasing function. The cones and coverage regions for $m = 6$ and $l = 3$ is shown in Figure 2. We define $R_k(q_i, q_j) \subset \mathbb{R}^2$ to be the interior of the region between radii $r(q_i)$ and $r(q_j)$ in cone C_k, as shown in Figure 3.

2.2. Properties of the model

We assume that the nodes are roughly at the same height, and hence we can model the process in two dimensions. The IID distribution of nodes implies that the location of each node is independent of that of the others. An example when this would be a reasonable assumption is environmental

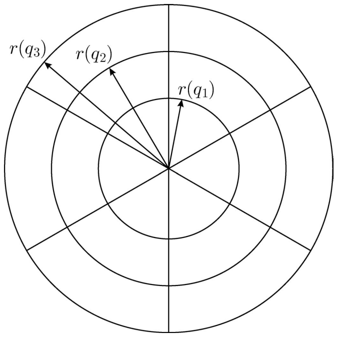


Figure 2. Circular coverage regions and cones for $m = 6$, $Q = \{q_1, q_2, q_3\}$.

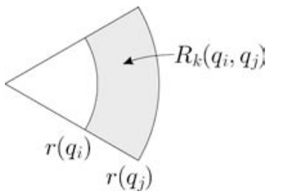


Figure 3. Region $R_k(q_i, q_j)$ in cone k.

monitoring and reconnaissance, where for example the nodes may be airdropped.

Each node can detect the *direction* of arrival of the received signal. This requires an antenna configuration with this capability. A node can communicate with all the nodes in a circular region centered on it, whose radius is determined by the transmitted power and the path loss model. Further the path loss model is *deterministic* in the sense that if a node is within this region it is guaranteed to receive the transmission. We also assume that there are no obstructions within the coverage region of any node, and each node has the same threshold power required to decode a received signal.

3. Markov decision process

3.1. State space

We denote random variables with upper case letters and their realizations in lower case. We focus on a single node, and model its state at time t by a random variable $X(t)$. Intuitively, one expects that the state should have two parts; one the maximum power transmitted so far and the second characterizing how many nodes have been found in which cones. We make a specific choice in the following, and show that it has the desired properties of a state; that is, that the resulting process is Markov.

Let $X_{\text{power}}(t)$ denote the maximum power transmitted until time t, and let $X_{\text{found}}(t)$ be the number of cones in which at least one neighbor has been found. The state $X(t)$ of the node is then

$$X(t) = (X_{\text{found}}(t), X_{\text{power}}(t))$$

Since the number of cones is m, we have

$$X(t) \in \{(0, 0)\} \bigcup \{Q \times \{0, 1, \ldots, m\}\}$$

The total number of states is $n = l\,(m + 1) + 1$. We will call a state x a *terminating state* if

$$x_{\text{found}} = m \quad \text{or } x_{\text{power}} = q_l.$$

That is, a state is a terminating state if at least one node has been found in each cone and/or the maximum available transmission power has been used. We partition the set of states into the set of terminating states S_{term} and the remaining states $S_{\text{non-term}}$.

3.2. Action space

In each state the action space is the set of available transmission powers in that state. In terminating states, only zero transmission power is considered. For state $x \in S_{\text{non-term}}$, the action space is

$$A_x = \{\, q \in Q \mid q > x_{\text{power}} \}$$

Since the path-loss model is deterministic, only a strict increase in power can lead to discovery of new neighbors.

Define the *policy space* as the product of the action spaces of non-terminating states

$$A = \prod_{x \in S_{\text{non-term}}} A_x$$

At time t, let a *policy* $\delta(t) \in A$. A policy defines a map which assigns to each state $x \in S_{\text{non-term}}$ an action in the set A_x, denoted by $\delta_{\text{x}}(t)$. The policy $\delta(t)$ is called *stationary* if it is constant over time. In this case, we have for some $\delta \in A$,

$$\delta(t) = \delta \quad \text{for all } t.$$

3.3. State evolution

For a region $R \subset D$ let $N(R)$ be the number of nodes in R, and define the indicator function $I(R)$ such that

$$I(R) = \begin{cases} 1 & \text{if } N(R) \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

Note that both $I(R)$ and $N(R)$ are random variables. Then the stochastic process X is defined by the state evolution equations

$$\begin{aligned} X_{\text{power}}(t+1) &= \delta_{X(t)}(t) \\ X_{\text{found}}(t+1) &= \sum_{k=1}^{m} I(R_k(0, X_{\text{power}}(t+1))) \qquad (1) \\ &= \sum_{k=1}^{m} I(R_k(0, \delta_{X(t)}(t))). \end{aligned}$$

3.4. Markov property

The process X is a Markov decision process, i.e.

$$\begin{aligned}\Pr\{X(t+1) = {} & x \mid X(t), \delta(t), \\ & X(t-1), \ldots, X(0), \\ & \delta(t-1), \ldots, \delta(0)\,\} \\ = \Pr\{\, X(t+1) = {} & x \mid X(t), \delta(t)\,\}\end{aligned}$$

The proof of the following theorem is given in the Appendix.

Theorem 1. *For a stationary policy δ, the process X is Markov.*

For simplicity of notation, we assume that the policy δ is stationary and prove that the process X is Markov. The proof requires only minor changes if this assumption is removed. Intuitively, one can see that X is a Markov decision process, since the random variables corresponding to the locations of nodes in a ring between circles of radii r and $r + \delta r$, are independent of the random variables corresponding to the locations of nodes within a circle of radius r. This is because we have assumed the nodes to have a Poisson distribution.

3.5. Transition matrix

Since the process X is Markov, it has a transition matrix, which we denote by $P(\delta)$ for policy δ. Denote the elements of this transition matrix by

$$P_{yw}(\delta) = \Pr\{\, X(t+1) = w \mid X(t) = y, \delta(t) = \delta\} \qquad (2)$$

Let B be the parametrized probability mass function of a binomial random variable, given by

$$B(p, n, l) = \binom{n}{l}(1-p)^{n-l}p^{l}$$

Now the area of $R_k(y_{\text{power}}$,$\delta_y))$ is

$$\text{Area}(R_k(y_{\text{power}}, \delta_y)) = \frac{2\pi}{m}\left(\frac{r(\delta_y)^2 - r(y_{\text{power}})^2}{2}\right)$$

Define the function θ by

$$\theta(y, \delta) = \Pr\{I(R_k(y_{\text{power}}, \delta_y)) = 1\}.$$

Since the distribution of nodes is two-dimensional Poisson, we have

$$\theta(y, \delta) = 1 - \exp\left(-\rho\frac{2\pi}{m}\left(\frac{r(\delta_y)^2 - r(y_{\text{power}})^2}{2}\right)\right)$$

If $\mathrm{y} \in \mathrm{S}_{\text{non-term}}$, then equations (1) and (7), together with non-negativity of $\mathrm{Z(t+1)}$, imply that

$$P_{yw}(\delta) = 0 \quad \text{if } w_{\text{power}} \neq \delta_y \text{ or } w_{\text{found}} < y_{\text{found}}$$

For $w_{\text{power}} = \delta_y$ and $w_{\text{found}} \geq y_{\text{found}}$,

$$\begin{aligned}P_{yw}(\delta) &= \Pr\{N_{\text{new-cones}}(y_{\text{power}}, w_{\text{power}}) = w_{\text{found}} - y_{\text{found}}\} \\ &= B(\theta(y, \delta), m - y_{\text{found}}, w_{\text{found}} - y_{\text{found}})\end{aligned}$$

For $y \in S_{\text{term}}$, we have

$$P_{yw}(\delta) = \begin{cases} 1 & \text{for } y = w \\ 0 & \text{otherwise}\end{cases} \qquad (3)$$

since once a node reaches a terminating state, it stays in that state. Ordering the states such that the non-terminating states come before the terminating states, we have

$$P(\delta) = \begin{bmatrix} P_1(\delta) & P_2(\delta) \\ 0 & I \end{bmatrix} \qquad (3)$$

which is a block decomposition of $P(\delta)$. We will use this to compute an optimal policy.

3.6. An equivalent substochastic system

We would like to minimize the expected total energy used in finding a node in each cone. This corresponds to a stochastic shortest path problem where we would like to minimize the expected cost of reaching a terminating state, starting in the initial state (0,0).

The cost function in state x with policy δ is

$$c_x(\delta) = \begin{cases} \delta_x & \text{for } x \in S_{\text{non-term}} \\ 0 & \text{for } x \in S_{\text{term}}\end{cases}$$

Let $c(\delta)$ denote the vector of costs for a policy δ, with same ordering as for the transition matrix. Since non-terminating states come before terminating states, we have

$$c(\delta) = \begin{bmatrix} \delta \\ 0 \end{bmatrix}$$

Since Q is a set of l transmit powers, after l time steps termination is guaranteed either because the maximum power has been used or because a neighbor has been found in each of the m cones. Hence the structure of $P(\delta)$ for all $\delta \in A$ is such that after time $t \geq l$, $\Pr\{X(t) \in S_{\text{term}}\} = 1$. Hence for all δ, we have the matrix product

$$P(\delta(1))P(\delta(2))\cdots P(\delta(l)) = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}$$

Define the t step transition matrix to be

$$P_{\text{step}}(t) = P(\delta(1))\cdots P(\delta(t))$$

Hence the total cost of the neighbor location process is

$$\sum_{t=1}^{l} P_{\text{step}}(t)c(\delta(t)) = \begin{bmatrix} \sum_{t=1}^{l} P_{\text{blockstep}}(t)\delta(t) \\ 0 \end{bmatrix}$$

where

$$P_{\text{blockstep}}(t) = P_1(\delta(1))\cdots P_1(\delta(t)) \qquad (4)$$

Hence we need to consider only the substochastic system[1] $x \in S_{\text{non-term}}$ to determine the cost for neighbor location for a particular policy δ.

4. Computation of an optimal policy

4.1. Dynamic programming recursion

In this section we make use of standard results on dynamic programming for stochastic shortest path problems. In particular, we follow the conventions in [7]; see also [6].

Define the vector V_δ whose i'th component is the total cost when starting in state i and using policy δ. We have

$$V_\delta = \sum_{t=1}^{l} P_{\text{blockstep}}(t)\delta(t)$$

Since $P_1(\delta)$ is a substochastic matrix for every δ, every policy δ is transient[2] (see for example, Theorem 2 in [7]). Hence there exists a stationary policy δ^* such that

$$V_{\delta^*} \leq V_\delta \quad \text{for all } \delta \in A$$

where the inequality holds elementwise. Moreover V_{δ^*} is the unique solution of

$$V = \min_{\delta \in A} (\delta + P_1(\delta)V) \tag{5}$$

where again the minimum holds elementwise.

It is standard that the dynamic programming recursion

$$V(k+1) = \min_{\delta \in A} (\delta + P_1(\delta)V(k))$$

converges to the optimal cost. For the Markov decision process constructed in this paper, this iteration converges *exactly* to V_{δ^*} in a finite number of steps, since there is an upper bound on the number of time-steps before the Markov decision process terminates, and there is no cost associated with terminating states. Once the optimal V is found, an optimal policy δ^* can be found as a solution to equation (5).

4.2. Linear programming approach

An alternative to applying the dynamic programming recursion is to compute the solution to the stochastic shortest path problem using linear programming, as in for example [3]. The vector of optimal costs V_{δ^*} is the unique solution of the following linear program

$$\begin{aligned} &\text{maximize } 1^T V \\ &\text{subjectto } V \leq \delta + P_1(\delta)V, \quad \delta \in A \end{aligned}$$

Here 1 denotes the vector of 1's. Let V_{opt} denote the optimal solution to this linear program. Any policy δ satisfying

$$V_{\text{opt}} = \delta + P_1(\delta)V_{\text{opt}}$$

is an optimal policy. This policy may also be obtained directly through the solution of the dual linear program.

4.3. The automaton for an optimal policy

In this section we explicitly construct an optimal strategy for an example, and represent it as a finite automaton. We use 4 cones, a node density of 10 per unit area, and transmission powers

$$Q = \{1, 2, 3, 4, 5\}$$

The function r determining the range within which transmissions can be decoded by other nodes is

$$r(q) = q^{\frac{1}{4}}$$

corresponding to a path-loss model where received power decays as $1/\mathrm{d}^4$.

A finite state automaton implementing this strategy is shown in Figure 4. Each circle represents a set of states, and the number contained in the circle is the power that should be transmitted in any of those states. After the power is transmitted, the number of cones in which at least one node has been found so far determines which is the next set of states. Note that we can group states into sets in this way because the policy is the same for many different states. The required number of non-empty cones to make a transition is shown next to the corresponding arrow. It can be seen that for the set of states corresponding to the same transmit power,

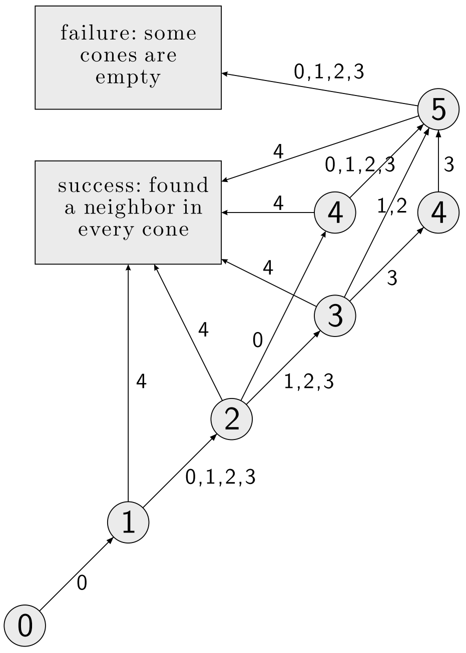

failure: some cones are empty
success: found a neighbor in every cone
0,1,2,3
5
4
0,1,2,3
3
4
4
1,2
4
4
3
3
4
0
4
1,2,3
2
0,1,2,3
1
0
0

Figure 4. State Transition Automation.

[1] We say that a system is substochastic if for all $\delta \in A$, the transition matrix $\mathrm{P}(\delta)$ is such that the sum of each row of $P(\delta)$ is less than one.

[2] A policy $\delta(t)$ is said to be transient if $\sum_{t=1}^{\infty} \tilde{p}_{\text{blockstep}}(t)$ is finite, where $\tilde{p}_{\text{blockstep}}(t) = P(\delta(1)) \cdots P(\delta(t))$.

the optimal policy exhibits a *monotonic* relationship with the number of cones in which at least one node has been found. The next transmit power is a non-increasing function of the number of cones with at least one neighbor. This is what we expect, since if we have found only a small fraction of the desired neighbors with the previous power, we would increase the transmit power by a larger amount compared to when a greater number of neighbors has been found.

The automaton is driven by the underlying probability distribution, since state transitions are determined by the number of nodes detected. This example illustrates the simplicity of the policy. Note that some states are never visited, such as $x_{\text{found}} = 1$ and $x_{\text{power}} = 4$, and so need not be stored on the node.

5. Numerical results

In this section we perform two types of computation and compare the results. We compute, using dynamic programming, the theoretical minimum energy policy and the corresponding energy usage. We also perform simulations by generating random distributions of nodes in a region D and simulating the optimum policy to

1. compare the performance of our scheme with that of an ideal scheme (perfect knowledge), and a simple scheme (doubling the power).
2. study the effect of boundary nodes on the average energy expenditure.

We compute the optimal policy and cost for various different system parameters. The scenario we consider has nodes identically and independently distributed over a region of area 1000 with a uniform distribution. We use the set of powers

$$Q = \{i/5 \,|\, i \in \mathbb{Z}, 1 \leq i \leq 25\}.$$

The path-loss model is

$$q_{\text{rec}} = \frac{q_{\text{trans}}}{d^4}$$

where q_{rec} is the power received, q_{trans} is the power transmitted, and d is the distance between the transmitter and the receiver. This is a good model if the inter-node distance is sufficiently large, and there are only two components in the received signal—*reflection from the ground* and the *line of sight* component. More complicated path-loss models (like in [16]) may also be used. We use a deterministic model in which the receiver is able to decode the signal if the received power q_{rec} is greater than a threshold power q_{th}. This will occur if the distance between the transmitter and receiver is less than $r(q_{\text{trans}})$, where

$$r(q_{\text{trans}}) = \left(\frac{q_{\text{trans}}}{q_{\text{th}}}\right)^{\frac{1}{4}}$$

We choose $q_{\text{th}} = 1$ for this example.

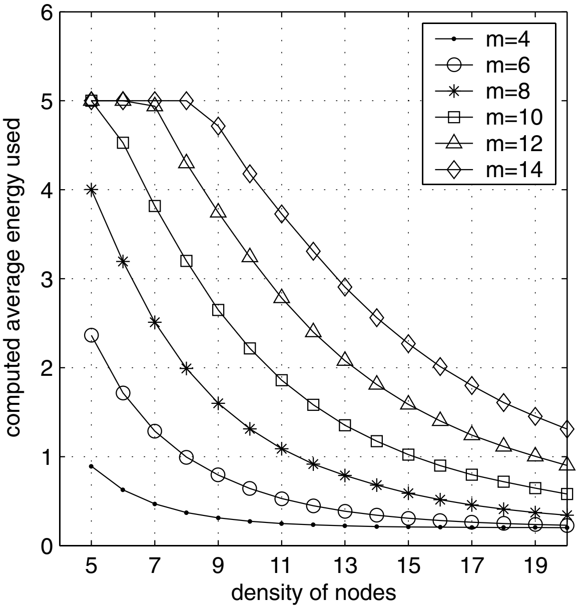


Figure 5. Computed cost as a function of density, for different number of cones.

5.1. *Variation in the optimal cost with density and number of cones*

The average energy consumption of the optimal policy was computed via dynamic programming for a range of densities, and a range of numbers of cones. The results are shown in Figure 5. As the density of nodes increases, the average amount of power required to find a neighbor in a cone decreases, hence the optimal energy used decreases as shown. Similarly as the cone angle is decreased, each node needs to find neighbors in more cones, leading to an increase in the average energy used.

A larger number of cones corresponds to a finer discretization of space, and it has been shown in [20,11] that this enables the establishment of paths that are closer to the minimum energy paths in the overall network. Hence there is a trade-off in energy consumption during the neighbor discovery phase and energy consumption during network operation after the resulting paths are established.

The maximum energy that any of the computed policies will use corresponds to a single transmission with maximum power. When the number of cones is large, and the density of nodes is low, this is the optimal strategy, as is seen in the figure.

5.2. *Simulation results*

The simulations were performed using 6 cones, corresponding to the same scenario as the above computation. In Figure 6, the following values are plotted as a function of density.

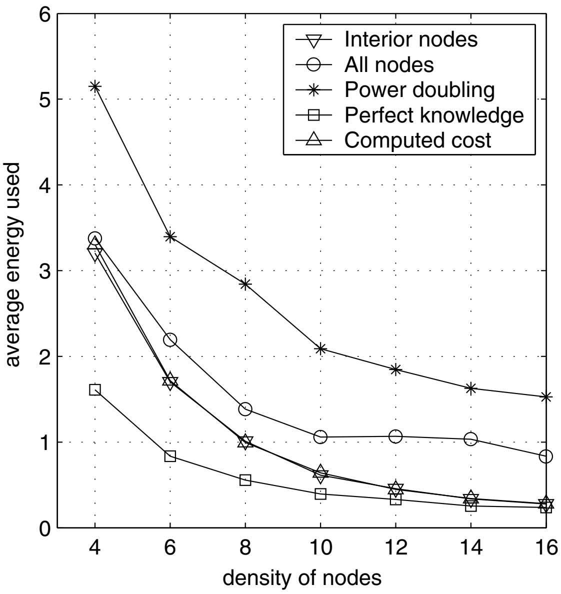


Figure 6. Simulated average energy of search process.

1. *Minimum energy with perfect knowledge:* This is the average energy a node would need to use to reach at least one neighbor in each of its cones, if it had perfect knowledge of the location of its neighbors. This provides a lower bound on the achievable cost by any policy.
2. *Computed average energy:* This is the minimum energy cost for the search process predicted by the dynamic programming computation. As expected, this value is higher than the minimum energy with perfect knowledge.
3. *Simulated average energy for interior nodes:* We call a node an *interior node* if its coverage region lies completely inside the region D. The plot shows the average energy used by these nodes in the simulation; it matches closely that predicted by the computation.
4. *Simulated average energy over all nodes:* This is the average over the energy used by all the nodes in region D, including *boundary nodes* which are those for which the coverage region lies partly outside D. Since all nodes are within D, the boundary nodes may not have nodes in one or more cones of their coverage regions. Hence they will continue transmitting with increasing power until the maximum power is reached. This reduces the efficiency of the computed minimum energy policy in practice, since the computation ignored the effect of the boundary. Hence it leads to an increase in average energy used.
5. *Power Doubling Scheme:* This is the average over energy used by all nodes, when the transmitting power is simply doubled till a neighbor is found in each cone. We can see that the optimal policy has lower energy cost compared to this scheme. Also, note that for a very low density of nodes, power doubling consumes more energy than simply broadcasting at the maximum transmission power. This is because power doubling starts with the minimum transmission power and doubles the power each time until neighbors have been found in all cones. Hence, for low densities many nodes will double their power until they reach the maximum transmission power, and in the process consume more energy than simply transmitting at the maximum power.

6. Conclusions

In this paper we have proposed an approach for minimizing the average energy consumed during the neighbor discovery phase of a wireless ad hoc sensor network. In such networks with a high node density, the energy saved by using this optimal scheme during each instance of neighbor location can significantly increase the lifetime of the network. This approach for neighbor location, combined with the methodology in [20] leads to an efficient way for a wireless sensor network to form low energy routes with small node degree.

There are many remaining open questions and possibilities for refinements of this method. In particular, in this paper each node discovers neighbors only through responses to its own queries, and ignores packets it receives through queries and responses of its neighbors. In this sense one would like a more *cooperative* approach between nodes to further optimize the performance. Other open issues include network rediscovery when nodes fail or move, and the effects of interference. In particular, in this case one may leverage the savings in energy during topology formation to increase the lifetime of the network. These issues are very interesting and we anticipate their investigation in future research.

Acknowledgments

The authors would like to thank the anonymous reviewers for their detailed comments that helped to improve the quality of this paper.

Appendix

Here we show that the process X defined in Section 3 is a Markov process for a stationary policy δ. We will make use of the following preliminary lemmas.

Lemma 1 *Suppose $R, S \subset D$ are disjoint regions; that is $R \cap S = \emptyset$. Then*

$$I(R \cup S) = I(R) + I(S) - I(R)I(S)$$

Proof This holds because

$$I(R \cup S) = \max\{I(R), I(S)\}$$

and the indicator function may take only values one or zero. ■

Lemma 2 *If the regions $R_1, R_2, \ldots, R_k$ are pairwise disjoint, then $I(R_1), I(R_2), \ldots, I(R_k)$ are mutually independent. Moreover if the areas of the regions $R_1, R_2, \ldots, R_k$ are equal, then the random variables $I(R_1), I(R_2), \ldots, I(R_k)$ are IID.*

Proof Since the nodes have a two dimensional Poisson distribution, if regions $R_1, R_2, \ldots, R_k$ are pairwise disjoint, then $N(R_1), N(R_2), \ldots, N(R_k)$ are mutually independent (see for example [18]). *Since $I(R_i)$ is a function of $N(R_i)$,* it follows that $I(R_1), I(R_2), \ldots, I(R_k)$ are mutually independent. The last part of the lemma follows because the distribution of $I(R_i)$ depends only on the area of region R_i. ■

Lemma 3 *Suppose X and Z are stochastic processes, and*

$$X(t+1) = g(X(t), Z(t+1)) \tag{6}$$

for some function g. If $Z(t+1)$ is conditionally independent of Z(t), Z(t − 1), ..., Z(1), X(t − 1), ..., X(0), given X(t) then X is Markov.

Proof This is standard; see for example Theorem 2.2, p. 58 in [2]. ■

Theorem 1 *For a stationary policy δ, the process X is Markov.*

Proof *We have*

$$X_{\text{found}}(t+1) = \sum_{k=1}^{m} I\left(R_k(0, \delta_{X(t)})\right)$$

For each k, partition the region $R_k(0, \delta_{X(t)})$ into two parts, S_k and T_k, defined by

$$\begin{aligned} S_k &= R_k(0, X_{\text{power}}(t)) \\ T_k &= R_k(X_{\text{power}}(t), \delta_{X(t)}) \end{aligned}$$

That is, S_k is the region of cone C_k which has been explored until before action $\delta_{\mathrm{X}(t)}$, and T_k is the additional region explored due to the power increase by action $\delta_{\mathrm{X}(t)}$. We have suppressed the dependence of S_k and T_k on *X(t)* and δ. We have

$$R_k(0, \delta_{X(t)}) = S_k \cup T_k$$

and $\mathrm{S_k} \cap T_k = \emptyset$, so Lemma 1 gives

$$I(S_k \cup T_k) = I(S_k) + I(T_k) - I(S_k)I(T_k)$$

Now, from equation (1),

$$\begin{aligned} X_{\text{found}}(t+1) &= \sum_{k=1}^{m} I\left(R_k(0, \delta_{X(t)})\right) \\ &= \sum_{k=1}^{m} I(S_k \cup T_k) \end{aligned}$$

and hence

$$X_{\text{found}}(t+1) = \sum_{k=1}^{m} I(S_k) + \sum_{k=1}^{m} (I(T_k) - I(S_k)I(T_k))$$

which gives

$$X_{\text{found}}(t+1) = X_{\text{found}}(t) + Z(t+1) \tag{7}$$

where $Z(t{+}1)$ is defined as

$$Z(t+1) = \sum \left\{ I(T_k) \mid I(S_k) = 0 \right\}. \tag{8}$$

We can write this explicitly in terms of the state X as

$$Z(t+1) = N_{\text{new-cones}}\left(X_{\text{power}}(t), \delta_{X(t)}\right)$$

Here, the random variable $N_{\text{new}-\text{cones}}(q_i, q_j)$, defined by

$$\begin{aligned} &N_{\text{new-cones}}(q_i, q_j) \\ &= \sum \left\{ I(R_k(q_i, q_j)) \mid I(R_k(0, q_i)) = 0, k = 1, \ldots, m \right\} \end{aligned}$$

is the number of new cones in which at least one neighbor is found when power is increased from q_i to q_j.

We would now like to apply Lemma 3; to do this, we now show that $Z(t+1)$ is conditionally independent of $Z(t)$, $Z(t-1), \ldots, Z(1), X(t-1), \ldots, X(0)$, given *X(t)*. We have

$$\begin{aligned} Pr\,\{\,Z(t+1) &= i \mid X(t) = x, Z(t) = z, \\ X(t-1) &= x_{t-1}, \ldots, X(0) = x_0, \\ Z(t-1) &= z_{t-1}, \ldots, Z(1) = z_1\} \\ &= Pr\{N_{\text{new}-\text{cones}}(x_{\text{power}}, \delta_x) = i\} \end{aligned} \tag{9}$$

We now explicitly compute this probability. The regions

$$R_1(x_{\text{power}}, \delta_x), \ldots, R_m(x_{\text{power}}, \delta_x)$$

are pairwise disjoint, and all have the same area. Hence from Lemma 2, the random variables $\mathrm{I(R_1(x_{power}}, \delta_\mathrm{x})), \ldots, I(R_m(x_{\text{power}}, \delta_x))$ are IID. Since the distribution of nodes is a two dimensional Poisson process with intensity ρ, we have

$$\begin{aligned} \Pr\{I(R_k(x_{\text{power}}, \delta_x)) = 1\} &= p \\ \Pr\{I(R_k(x_{\text{power}}, \delta_x)) = 0\} &= 1 - p \end{aligned}$$

where

$$p = 1 - \exp\left(-\rho \frac{2\pi}{m}\left(\frac{r(\delta_x)^2 - r(x_{\text{power}})^2}{2}\right)\right)$$

Now $N_{\text{new}-\text{cones}}(x_{\text{power}}, \delta_x)$ is the sum of $\mathrm{m - x_{found}}$ binary IID random variables, and it therefore has a binomial distribution given by

$$\Pr\{N_{\text{new}-\text{cones}}(x_{\text{power}}, \delta_x) = i\} = p^i (1-p)^{m - x_{\text{found}} - i}$$

which is function of only x; in particular it does not depend on $x_{t-1}, \ldots, x_0$ or $z_t, \ldots, z_0$. Hence $Z(t+1)$ is conditionally independent of $Z(t)$, $Z(t-1), \ldots, Z(1)$, $X(t-1), \ldots, X(0)$, given $X(t)$, and Lemma 3 implies that X is Markov, as desired.

References

[1] D.P. Bertsekas and J.N. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3) (1991) 580–595.

[2] P. Brémaud. *Markov Chains, Gibbs Fields, Monte Carlo Simulation, And Queues*. Springer Verlag (1999).

[3] C. Derman. *Finite State Markovian Decision Processes*. Academic Press (1970).

[4] J.H. Eaton and L.A. Zadeh. Optimal pursuit strategies in discrete-state probabilistic systems. *Trans. ASME Ser. D,* J. Basic Eng, 84 (1962) 23–29.

[5] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. ICASSP, 2001.

[6] A.F. Veinott Jr. Discrete dynamic programming with sensitive discount optimality criteria. *Annals of Mathematical Statistics,* 40(5) (1969) 1635–1660.

[7] A.F. Veinott Jr. Markov decision chains. In G.B. Dantzig and B.C. Eaves, editors, Studies in Optimization, pages 124–159. The Mathematical Association of America (1974).

[8] B. Karp and H.T. Kung. Greedy perimeter stateless routing (GPSR) for wireless networks. *Proc. Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, (2000) 243–254.

[9] Y.B. Ko and N.H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. *Proc. Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, (1998) 66–75.

[10] L. Li and J.Y. Halpern. Minimum energy mobile wireless networks revisited. *IEEE International Conference on Communications (ICC)*, 2001.

[11] L. Li, J.Y. Halpern, P. Bahl, Y.M. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithms for wireless multi-hop networks. *ACM Symposium on Principle of Distributed Computing (PODC)*, (2001).

[12] P. Misra and P. Enge. *Global position system: signals, measurement and performance*. Ganga-Jamuna Press, (2001).

[13] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications, (1999) 90–100.

[14] G.J. Pottie and W.J. Kaiser. Wireless sensor networks. *Communications of the ACM*, 43(5) (2000) 51–58.

[15] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. *IEEE INFOCOM*, (2000).

[16] T.S. Rappaport. *Wireless communications: principles and practice*. Prentice Hall (1996).

[17] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. *IEEE J. Selected Areas in Communications,* 17(8) (1999) 1333–1344.

[18] D.L. Snyder and M.I. Miller. *Random Point Processes in Time and Space*. Springer Verlag, 2nd edition, 1991.

[19] M.I.A. Stoelinga. An introduction to probabilistic automata. In G. Rozenberg, editor, *EATCS bulletin*, volume 78, 2002.

[20] R. Wattenhofer, L. Li, P. Bahl, and Y.M. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. *IEEE INFOCOM* (2001).

Ritesh Madan received the B.Tech. degree from the Indian Institute of Technology (IIT), Mumbai, India, in 2001, and the M.S. degree from Stanford University, CA, in 2003, both in Electrical Engineering. He is currently working toward the Ph.D. degree in Electrical Engineering at Stanford University, CA. He is a Sequoia Capital Stanford Graduate Fellow. His research interests include wireless networks, optimization, and distributed algorithms.
E-mail: rkmadan@stanford.edu

Sanjay Lall is Assistant Professor of Aeronautics and Astronautics at Stanford University. Until 2000 he was a Research Fellow at the California Institute of Technology in the Department of Control and Dynamical Systems, and prior to that he was NATO Research Fellow at Massachusetts Institute of Technology, in the Laboratory for Information and Decision Systems. He received the Ph.D. in Engineering from the University of Cambridge, England. His research interests include optimization and distributed control.
E-mail: lall@stanford.edu