Minimum-Energy Mobile Wireless Networks Revisited

Li Li
Dept. of Computer Science
Cornell University
Ithaca NY 14853
lili@cs.cornell.edu

Joseph Y. Halpern
Dept. of Computer Science
Cornell University
Ithaca NY 14853
halpern@cs.cornell.edu

Abstract—**We propose a protocol that, given a communication network, computes a subnetwork such that, for every pair (u, v) of nodes connected in the original network, there is a a *minimum-energy path* between u and v in the subnetwork (where a minimum-energy path is one that allows messages to be transmitted with a minimum use of energy). The network computed by our protocol is in general a subnetwork of the one computed by the protocol given in [12]. Moreover, our protocol is computationally simpler. We demonstrate the performance improvements obtained by using the subnetwork computed by our protocol through simulation.**

I. Introduction

Multi-hop wireless networks, especially sensor networks, are expected to be deployed in a wide variety of civil and military applications. Minimizing energy consumption has been a major design goal for wireless networks. As pointed out by Rodoplu and Meng [12], network protocols that minimizes energy consumption are key to low-power wireless sensor networks.

We can characterize a communication network using a graph G' where the nodes in G' represent the nodes in the network, and two nodes u and v are joined by an edge if it is possible for u to transmit a message to v if u transmits at maximum power. Transmitting at maximum power requires a great deal of energy. To minimize energy usage, we would like a subgraph G of G' such that (1) G consists of all the nodes in G' but has fewer edges, (2) if u and v are connected in G', they are still connected in G, and (3) a node u can transmit to all its neighbors in G using less power than is required to transmit to all its neighbors in G'. Indeed, what we would really like is a subnetwork G of G' with these properties where the power for a node to transmit to its neighbors in G' is minimal. Rodoplu and Meng [12] provide a protocol that, given a communication network, computes a subnetwork that is energy-efficient in this sense. We call their protocol MECN (for *minimum-energy communication network*).

The key property of the subnetwork constructed by MECN is what we call the *minimum-energy property*. Given G', it guarantees that between every pair (u, v) of nodes that are connected in G', the subgraph G has a *minimum-energy path* between u and v, one that allows messages to be transmitted with a minimum use of energy among all the paths between u and v in G'. In this paper, we first identify conditions that are necessary and sufficient for a graph to have this minimum-energy property. We use this characterization to construct a protocol called SMECN (for *small* minimum-energy communication network). The subnetwork constructed by SMECN is provably smaller than that constructed by MECN if broadcasts at a given power setting are able to reach all nodes in a circular region around the broadcaster. We conjecture that this property will hold in practice even without this assumption. Our simulations show that by being able to use a smaller network, SMECN has lower link maintenance costs than MECN and can achieve a significant saving in energy usage. SMECN is also computationally simpler than MECN.

The rest of the paper is organized as follows. Section II gives the network model (which is essentially the same as that used in [12]). Section III identifies a condition necessary and sufficient for achieving the minimum-energy property. This characterization is used in Section IV to construct the SMECN protocol and prove that it constructs a network smaller than MECN if the broadcast region is circular. In Section V, we give the results of simulations showing the energy savings obtained by using the network constructed by SMECN. Section VI concludes our paper.

II. The Model

We use essentially the same model as Rodoplu and Meng [12]. We assume that a set V of nodes is deployed in a two-dimensional area, where no two nodes are in the same physical location. Each node has a GPS receiver on board, so knows it own location, to within at least 5 meters of accuracy. It does not necessarily know the location of other nodes. Moreover, the location of nodes will in general change over time.

A transmission between node u and v takes power $p(u, v) = td(u, v)^n$ for some appropriate constant t, where $n \geq 2$ is the path-loss exponent of outdoor radio propagation models [11], and $d(u, v)$ is the distance between u and v. A reception at the receiver takes power c. Computational power consumption is ignored.

Suppose there is some maximum power p_{max} at which the nodes can transmit. Thus, there is a graph $G' = (V, E')$ where $(u, v) \in E'$ if it is possible for u to transmit to v if it transmits at maximum power. Clearly, if $(u, v) \in E$, then $td(u, v)^n \leq p_{max}$. However, we do not assume that a node u can transmit to all nodes v such that $td(u, v)^n \leq p_{max}$. For one thing, there may be obstacles between u and v that prevent transmission. Even without obstacles, if a unit transmits using a directional transmit antenna, then only nodes in the region covered by the antenna (typically a cone-like region) will receive the message. Rodoplu and Meng [12] implicitly assume that every node can transmit to every other node. Here we take a first step in exploring what happens if this is not the case. However, we do assume that the graph G' is connected, so that there is a potential communication path between any pair of nodes in V.

Because the power required to transmit between a pair of nodes increases as the nth power of the distance between them, for some $n \geq 2$, it may require less power to relay information than to transmit directly between two nodes. As usual, a *path* $r = (u_0, \dots, u_k)$ in a graph $G = (V, E)$ is defined to be an ordered list of nodes such that $(u_i, u_{i+1}) \in E$. The *length* of $r = (u_0, \dots, u_k)$, denoted $|r|$, is k. The total power consumption of a path $r = (u_0, u_2, \cdots, u_k)$ in G' is the sum of the transmission and reception power consumed, i.e.,

$$C(r) = \sum_{i=1}^{k-1} (p(u_i, u_{i+1}) + c).$$

A path $r = (u_0, \dots, u_k)$ is a *minimum-energy path* from u_0 to u_k if $C(r) \leq C(r')$ for all paths r' in G' from u_0 to u_k. For simplicity, we assume that $c > 0$. (Our results hold even without this assumption, but it makes the proofs a little easier.)

A subgraph $G = (V, E)$ of G' has the *minimum-energy property* if, for all $(u, v) \in V$, there is a path r in G that is a minimum-energy path in G' from u to v.

III. A Characterization of Minimum-Energy Communication Networks

Our goal is to find a minimal subgraph G of G' that has the minimum-energy property. Note that a graph G with the minimum-energy property must be strongly connected since, by definition, it contains a path between any pair of nodes. Given such a graph, the nodes can communicate using the links in G.

For this to be useful in practice, it must be possible for each of the nodes in the network to construct G (or, at least, the relevant portion of G from their point of view) in a distributed way. In this section, we provide a condition that is necessary and sufficient for a subgraph of G' to be minimal with respect to the minimum-energy property. In the next section, we use this characterization to provide an efficient algorithm for constructing a graph G with the minimum-energy property that, while not necessarily minimal, still has relatively few edges.

Clearly if a subgraph $G = (V, E)$ of G' has the minimum-energy property, an edge $(u, v) \in E$ is *redundant* if there is a path r from u to v in G such that $|r| > 1$ and $C(r) \leq C(u, v)$. Let $G_{\min} = (V, E_{\min})$ be the subgraph of G' such that $(u, v) \in E_{\min}$ iff there is no path r from u to v in G' such that $|r| > 1$ and $C(r) \leq C(u, v)$. As the next result shows, $G_{\min}$ is the smallest subgraph of G' with the minimum-energy property.

Theorem III.1: A subgraph G of G' has the minimum-energy property iff it contains $G_{\min}$ as a subgraph. Thus, $G_{\min}$ is the smallest subgraph of G' with the minimum-energy property.

Proof: We first show that $G_{\min}$ has the minimum-energy property. Suppose, by way of contradiction, that there are nodes $u, v \in V$ and a path r in G' from u to v such that $C(r) < C(r')$ for any path r' from u to v in $G_{\min}$. Suppose that $r = (u_0, \dots, u_k)$, where $u = u_0$ and $v = u_k$. Without loss of generality, we can assume that r is the longest minimal-energy path from u to v. Note that r has no repeated nodes for any cycle can be removed to give a path that requires strictly less power. Since $G_{\min}$ has no redundant edges, for all $i = 0, \dots, k-1$, it follows that $(u_i, u_{i+1}) \in E_{\min}$. For otherwise, there is a path r_i in G' from u_i to u_{i+1} such that $|r_i| > 1$ and $C(r_i) \leq C(u_i, u_{i+1})$. But then it is immediate that there is a path r^* in G' such that $C(r^*) \leq C(r)$ and r^* is longer than r, contradicting the choice of r.

To see that $G_{\min}$ is a subgraph of every subgraph of G' with the minimum-energy property, suppose that there is some subgraph G of G' with the minimum-energy property that does not contain the edge $(u, v) \in E_{\min}$. Thus, there is a minimum-energy path r from u to v in G. It must be the case that $C(r) \leq C(u, v)$. Since (u, v) is not an edge in G, we must have $|r| > 1$. But then $(u, v) \notin E_{\min}$, a contradiction. ■

This result shows that in order to find a subgraph of G with the minimum-energy property, it suffices to ensure that it contains $G_{\min}$ as a subgraph.

IV. A Power-Efficient Protocol for Finding a Minimum-Energy Communication Network

Checking if an edge (u, v) is in $E_{\min}$ may require checking nodes that are located far from u. This may require a great deal of communication, possibly to distant nodes, and thus require a great deal of power. Since power-efficiency is an important consideration in practice, we consider here an algorithm for constructing a communication network that contains $G_{\min}$ and can be constructed in a power-efficient manner rather than trying to construct $G_{\min}$ itself.

Say that an edge $(u, v) \in E'$ is *k-redundant* if there is a path r in G' such that $|r| = k$ and $C(r) \leq C(u, v)$. Notice that $(u, v) \in E_{\min}$ iff it is not k-redundant for all $k > 1$. Let E_2 consist of all and only edges in E' that are not 2-redundant. In our algorithm, we construct a graph $G = (V, E)$ where $E \supseteq E_2$; in fact, under appropriate assumptions, $E = E_2$. Clearly $E_2 \supseteq E_{\min}$, so G has the minimum-energy property.

There is a trivial algorithm for constructing E_2. Each node u starts the process by broadcasting a *neighbor discovery message* (NDM) at maximum power p_{max}, stating its own position. If a node v receives this message, it responds to u with a message stating its location. Let $M(u)$ be the set of nodes that respond to u and let $N_2(u)$ denote u's neighbors in E_2. Clearly $N_2(u) \subseteq M(u)$. Moreover, it is easy to check that $N_2(u)$ consists of all those nodes $v \in M(u)$ other than u such that there is no $w \in M(u)$ such that $C(u, w, v) \leq C(u, v)$. Since u has the location of all nodes in $M(u)$, $N_2(u)$ is easy to compute.

The problem with this algorithm is in the first step, which involves a broadcast using maximum power. While this expenditure of power may be necessary if there are relatively few nodes, so that power close to p_{max} will be required to transmit to some of u's neighbors in E_2, it is unnecessary in denser networks. In this case, it may require much less than p_{max} to find u's neighbors in E_2. We now present a more power-efficient algorithm for finding these neighbors, based on ideas due to Rodoplu and Meng [12]. For this algorithm, we assume that if a node u transmits with power p, it knows the region $F(u, p)$ around u which can be reached with power p. If there are no obstacles and the antenna is omnidirectional, then this region is just a circle of radius d_p such that $td_p^n = p$. We are implicitly assuming that even if there are obstacles or the antenna is not omni-directional, a node u knows the terrain and the antenna characteristics well enough to compute $F(u, p)$. If there are no obstacles, we show that E_2 is a subgraph of what Rodoplu and Meng call the *enclosure graph*. Our algorithm is a variant of their algorithm for constructing the enclosure graph.

Before presenting the algorithm, it is useful to define a few terms.

Definition IV.1: Given a node v, let *Loc*(v) denote the physical location of v. The *relay region* of the transmit-relay node pair (u, v) is the physical region $R_{u \to v}$ such that relaying through v to any point in $R_{u \to v}$ takes less power than direct transmission. Formally,

$$R_{u \to v} = \{(x, y) : C(u, v, (x, y)) \leq C(u, (x, y))\},$$

where we abuse notation and take $C(u, (x, y))$ to be the cost of transmitting a message from u to a virtual node whose location is (x, y). That is, if there were a node v' such that *Loc*$(v') = (x, y)$, then $C(u, (x, y)) = C(u, v')$; similarly, $C(u, v, (x, y)) = C(u, v, v')$. Note that, if a node v is in the relay region $R_{u \to w}$, then the edge (u, v) is 2-redundant. Moreover, since $c > 0$, $R_{u \to u} = \emptyset$.

Given a region F, let

$$N_F = \{v \in V : \mathit{Loc}(v) \in F\};$$

if F contains u, let

$$R_F(u) = \bigcap_{w \in N_F} (F(u, p_{max}) - R_{u \to w}). \qquad (1)$$

The following proposition gives a useful characterization of $N_2(u)$.

Proposition IV.2: Suppose that F is a region containing the node u. If $F \supseteq R_F(u)$, then $N_{R_F(u)} \supseteq N_2(u)$. Moreover, if F is a circular region with center u and $F \supseteq R_F(u)$, then $N_{R_F(u)} = N_2(u)$.

Proof: Suppose that $F \supseteq R_F(u)$. We show that $N_{R_F(u)} \supseteq N_2(u)$. Suppose that $v \in N_2(u)$. Then clearly *Loc*$(v) \notin \cup_{w \in V} R_{u \to w}$ and *Loc*$(v) \in F(u, p_{max})$. Thus, *Loc*$(v) \in R_F(u)$, so $v \in N_{R_F(u)}$.

Now suppose that F is a circular region with center u and $F \supseteq R_F(u)$. The preceding paragraph shows that $N_{R_F(u)} \supseteq N_2(u)$. We now show that $N_{R_F(u)} \subseteq N_2(u)$. Suppose that $v \in N_{R_F(u)}$. If $v \notin N_2(u)$, then there exists some w such that $C(u, w, v) \leq C(u, v)$. Since transmission costs increase with distance, it must be the case that $d(u, w) \leq d(u, v)$. Since $v \in N_{R_F(u)} \subseteq N_F$ and F is a circular region with center u, it follows that $w \in N_F$. Since $C(u, w, v) \leq C(u, v)$, it follows that *Loc*$(v) \in R_{u \to w}$. Thus, $v \notin R_F(u)$, contradicting our original assumption. Thus, $v \in N_2(u)$. ■

The algorithm for node u constructs a set F such that $F \supseteq R_F(u)$, and tries to do so in a power-efficient fashion. By Proposition IV.2, the fact that $F \supseteq R_F(u)$ ensures that $N_{R_F(u)} \supseteq N_2(u)$. Thus, the nodes in $N_{R_F(u)}$ other than u itself are taken to be u's neighbors. By Theorem III.1, the resulting graph has the minimum-energy property.

Essentially, the algorithm for node u starts by broadcasting an NDM with some initial power p_0, getting responses from all nodes in $F(u, p_0)$, and checking if $F(u, p_0) \supseteq R_{F(u,p_0)}(u)$. If not, it transmits with more power. It continues increasing the power p until $F(u, p) \supseteq R_{F(u,p)}(u)$. It is easy to see that $F(u, p_{max}) \supseteq R_{F(u,p_{max})}(u)$, so that as long as the power increases to p_{max} eventually, then this process is guaranteed to terminate. In this paper, we do not investigate how to choose the initial power p_0, nor do we investigate how to increase the power at each step. We simply assume some function *Increase* such that *Increase*$^k(p_0) = p_{max}$ for sufficiently large k. An obvious choice is to take *Increase*$(p) = 2p$. If the initial choice of p_0 is less than the total power actually needed, then it is easy to see that this guarantees that the total amount of transmission power used by u will be within a factor of 2 of optimal. [1]

Thus, the protocol run by node u is simply

$p = p_0$;
while $F(u, p) \not\supseteq R_{F(u,p)}(u)$ do *Increase*(p);
$N(u) = N_{R_{F(u,p)}}$

A more careful implementation of this algorithm is given in Figure 1. Note that we also compute the minimum power $p(U)$ required to reach all the nodes in $N(u)$. In the algorithm, A is the set of all the nodes that u has found so far in the search and M consists of the new nodes found in the current iteration. In the the computation of η in the second-last line of the algorithm, we take $\cap_{v \in M}(F(u, p_{max}) - R_{u \to v})$ to be $F(u, p_{max})$ if $M = \emptyset$. For future reference, we note that it is easy to show that, after each iteration of the while loop, we have that $\eta = \cap_{v \in A}(F(u, p_{max}) - R_{u \to v})$.

Algorithm SMECN

$p = p_0$;
$A = \emptyset$;
$NonNbrs = \emptyset$;
$\eta = F(u, p_{max})$;
while $F(u, p) \not\supseteq \eta$ do
  $p =$ *Increase*(p);
  Broadcast NDM with power p and gather responses;
  $M = \{v | \mathit{Loc}(v) \in F(u, p), v \notin A, v \neq u\}$;
  $A = A \bigcup M$;
  for each $v \in M$ do
    for each $w \in A$ do
      if *Loc*$(v) \in R_{u \to w}$ then
        $NonNbrs = NonNbrs \bigcup \{v\}$;
      else if *Loc*$(w) \in R_{u \to v}$ then
        $NonNbrs = NonNbrs \bigcup \{w\}$;
  $\eta = \eta \cap \bigcap_{v \in M}(F(u, p_{max}) - R_{u \to v})$;
$N(u) = A - NonNbrs$;
$p(u) = \min\{p : F(u, p) \supseteq \eta\}$

Fig. 1. Algorithm SMECN running at node u.

Define the graph $G = (V, E)$ by taking $(u, v) \in E$ iff $v \in N(u)$, as constructed by the algorithm in Figure 1. It is immediate from the earlier discussion that $E \supseteq E_2$. Thus

Theorem IV.3: G has the minimum-energy property.

We next show that SMECN dominates MECN. MECN is described in Figure 2. For easier comparison, we have made some inessential changes to MECN to make the notation and presentation more like that of SMECN. The main difference between SMECN and MECN is the computation of the region η. As

[1] Note that, in practice, a node may control a number of directional transmit antennae. Our algorithm implicitly assumes that they all transmit at the same power. This was done for ease of exposition. It would be easy to modify the algorithm to allow each antenna to transmit using different power. All that is required is that after sufficiently many iterations, all antennae transmit at maximum power.

Algorithm MECN
$p = p_0$;
$A = \emptyset$;
$NonNbrs = \emptyset$;
$\eta = F(u, p_{max})$;
while $F(u,p) \not\supseteq \eta$ do
  $p = Increase(p)$;
  Broadcast NDM with power p and gather responses;
  $M = \{v | Loc(v) \in F(u,p), v \notin A, v \neq u\}$;
  $A = A \bigcup M$;
  $NonNbrs = NonNbrs \bigcup M$;
  for each $v \in M$ do $Flip(v)$;
  $\eta = \bigcap_{v \in (A - NonNbrs)} (F(u, p_{max}) - R_{u \to v})$;
$N(u) = A - NonNbrs$;
$p(u) = \min\{p : F(u,p) \supseteq \eta\}$

Procedure $Flip(v)$
  if $v \notin NonNbrs$ then
    $NonNbrs = NonNbrs \bigcup \{v\}$;
    for each $w \in A$ such that $Loc(w) \in R_{u \to v}$ do
      $Flip(w)$;
  else if $Loc(v) \notin \cup_{w \in A - NonNbrs} R_{u \to w}$ then
    $NonNbrs = NonNbrs - \{v\}$;
    for each $w \in A$ such that $Loc(w) \in R_{u \to v}$ do
      $Flip(w)$;

Fig. 2. Algorithm MECN running at node u.

we observed, in SMECN, $\eta = \cap_{v \in A}(F(u, p_{max}) - R_{u \to v})$ at the end of every iteration of the loop. On the other hand, in MECN, $\eta = \cap_{v \in A - NonNbrs}(F(u, p_{max}) - R_{u \to v})$. Moreover, in SMECN, a node is never removed from *NonNbrs* once it is in the set, while in MECN, it is possible for a node to be removed from *NonNbrs* by the procedure *Flip*. Roughly speaking, if a node $v \in R_{u \to w}$, then, in the next iteration, if $w \in R_{u \to t}$ for a newly discovered node t, but $v \notin R_{u \to t}$, node v will be removed from *NonNbrs* by $Flip(v)$. In [12], it is shown that MECN is correct (i.e., it computes a graph with the minimum-energy property) and terminates (and, in particular, the procedure *Flip* terminates). Here we show that, at least for circular search regions, SMECN does better than MECN.

Theorem IV.4: If the search regions considered by the algorithm SMECN are circular, then the communication graph constructed by SMECN is a subgraph of the communication graph constructed by MECN.

Proof: For each variable x that appears in SMECN, let x_S^k denote the value of x after the kth iteration of the loop; similarly, for each variable in MECN, let x_M^k denote the value of x after the kth iteration of the loop. It is almost immediate that SMECN maintains the following invariant: $v \in NonNbrs_S^k$ iff $v \in A_S^k$ and $Loc(v) \in \cup_{w \in A_S^k} R_{u \to w}$. Similarly, it is not hard to show that MECN maintains the following invariant: $v \in NonNbrs_M^k$ iff $v \in A_S^k$ and $Loc(v) \in \cup_{w \in A_M^k - NonNbrs_M^k} R_{u \to w}$. (Indeed, the whole point of the *Flip* procedure is to maintain this invariant.) Since it is easy to check that $A_S^k = A_M^k$, it is immediate that $NonNbrs_S^k \supseteq NonNbrs_M^k$. Suppose that SMECN terminates after k_S iterations of the loop and MECN terminates after k_M MECN iterations of the loop. Hence $\eta_S^k \subseteq \eta_M^k$ for all $k \leq \min(k_S, k_M)$. Since both algorithms use the condition $F(u,p) \supseteq \eta$ to determine termination, it follows that SMECN terminates no later than MECN; that is, $k_S \leq k_M$.

Since the search region used by SMECN is assumed to be circular, by Proposition IV.2, $A_S^{k_S} - NonNbrs_S^{k_S} = N_2(u)$. Moreover, even if we continue to iterate the loop of SMECN (ignoring the termination condition), then $F(u,p)$ keeps increasing while η keeps decreasing. Thus, by Proposition IV.2 again, we continue to have $A_S^k - NonNbrs_S^k = N_2(u)$ even if $k \geq k_S$. That means that if we were to continue with the loop after SMECN terminates, none of the new nodes discovered would be neighbors of u. Since the previous argument still applies to show that $NonNbrs_S^{k_M} \supseteq NonNbrs_M^{k_M}$, it follows that $N_2(u) = A_S^{k_M} - NonNbrs_S^{k_M} \subseteq A_M^{k_M} - NonNbrs_M^{k_M}$. That is, the communication graph constructed by SMECN has a subset of the edges of the communication graph constructed by MECN. ∎

In the proof of Theorem IV.4, we implicitly assumed that both SMECN and MECN use the same value of initial value p_0 of p and the same function *Increase*. In fact, this assumption is not necessary, since the neighbors of u in the graph computed by SMECN are given by $N_2(u)$ independent of the choice of p_0 and *Increase*, as long as $F(u, p_0) \not\supseteq F(u, p_{max})$ and $Increase^k(p_0) \geq p_{max}$ for k sufficiently large. Similarly, the proof of Theorem IV.4 shows that the set of neighbors of u computed by MECN is a superset of $N_2(u)$, as long as *Increase* and p_0 satisfy these assumptions.

Theorem IV.4 shows that the neighbor set computed by MECN is a superset of $N_2(u)$. As the following example shows, it may be a strict superset (so that the communication graph computed by SMECN is a strict subgraph of that computed by MECN).

Example IV.5: Consider a network with 4 nodes t, u, v, w, where $Loc(v) \in R_{u \to w}$, $Loc(w) \in R_{u \to t}$, and $Loc(v) \notin R_{u \to t}$. It is not hard to choose power functions and locations for the nodes which have this property. It follows that $N_2(u) = \{t\}$. (It is easy to check that $Loc(t) \notin R_{u \to v} \cup R_{u \to w}$.) On the other hand, suppose that *Increase* is such that t, v, and w are added to A in the same step. Then all of them are added to *NonNbrs* in MECN. Which ones are taken out by *Flip* then depends on the order in which they are considered in the loop [2] For example, if they are considered in the order v, w, t, then the only neighbor of u is again t. However, if they are considered in any other order, then both v and t become neighbors of u. For example, suppose that they are considered in the order t, w, v. Then *Flip* makes t a neighbor, does not make w a neighbor (since $Loc(w) \in R_{u \to t}$), but does make v a neighbor (since $Loc(v) \notin R_{u \to t}$). Although $Loc(v) \notin R_{u \to w}$, this is not taken into account since $w \in NonNbrs$ at the point when v is considered.

V. Simulation Results and Evaluation

How can using the subnetwork computed by (S)MECN help performance? Clearly, sending messages on minimum-energy paths is more efficient than sending messages on arbitrary paths,

[2]Note that the final neighbor set of MECN is claimed to be independent of the ordering in [12]. However, the example here shows that this is not the case.

but the algorithms are all local; that is, they do not actually find the minimum-energy path, they just construct a subnetwork in which it is guaranteed to exist.

There are actually two ways that the subnetwork constructed by (S)MECN helps. First, when sending periodic beaconing messages, it suffices for u to use power $p(u)$, the final power computed by (S)MECN. Second, the routing algorithm is restricted to using the edges $\cup_{u \in V} N(u)$. While this does not guarantee that a minimum-energy path is used, it makes it more likely that the path used is one that requires less energy consumption.

To measure the impact of focusing on energy efficiency, we compared the use of MECN and SMECN in a simulated application setting.

Both SMECN and MECN were implemented in ns-2 [10], using the wireless extension developed at Carnegie Mellon [3]. The simulation was done for a network of 200 nodes, each with a transmission range of 500 meters. The nodes were placed uniformly at random in a rectangular region of 1500 by 1500 meters. (There has been a great deal of work on realistic placement, e.g. [13], [2]. However, this work has the Internet in mind. Since the nodes in a multihop network are often best viewed as being deployed in a somewhat random fashion and move randomly, we believe that the uniform random placement assumption is reasonable in many large multihop wireless networks.)

We assume a $1/d^4$ transmit power roll-off for radio propagation. The carrier frequency is 914 MHz; transmission raw bandwidth is 2 MHz. We further assume that each node has an omni-directional antenna with 0 dB gain, which is placed 1.5 meter above the node. The receive threshold is 94 dBW, the carrier sense threshold is 108 dBW, and the capture threshold is 10 dB. These parameters simulate the 914 MHz Lucent WaveLAN DSSS radio interface. Given these parameters, the t parameter in Section II is 101 dBW. We ignore reception power consumption, i.e. $c = 0$.

Each node in our simulation has an initial energy of 1 Joule. We would like to see how our algorithm affects network performance. To do this, we need to simulate the network's application traffic. We used the following application scenario. All nodes periodically send UDP traffic to a sink node situated at the boundary of the network. The sink node is viewed as the master data collection site. The application traffic is assumed to be CBR (constant bit rate); application packets are all 512 bytes. This application scenario has also been used in [4]. Although this application scenario does not seem appropriate for telephone networks and the Internet (cf. [7], [8]), it does seem reasonable for ad hoc networks, for example, in environment-monitoring sensor applications. In this setting, sensors periodically transmit data to a data collection site, where the data is analyzed.

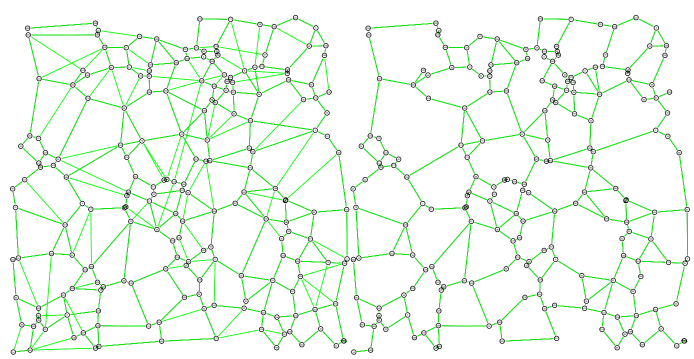
To find routes along which to send messages, we use AODV [9]. However, as mentioned above, we restrict AODV to finding routes that use only edges in $\cup_v N(u)$. There are other routing protocols, such as LAR [6], GSPR [5], and DREAM [1], that take advantage of GPS hardware. We used AODV because it is readily available in our simulator and it is well studied. We do not believe that using a different routing protocol would significantly affect the results we present here.

We assumed that each node in our simulation had an initial energy of 1 Joule and then ran the simulation for 1200 simulation seconds, using both SMECN and MECN. We did not actually simulate the execution of SMECN and MECN. Rather, we assumed the neighbor set $N(u)$ and power $p(u)$ computed by (S)MECN each time it is run were given by an oracle. (Of course, it is easy to compute these the neighbor set and power in the simulation, since we have a global picture of the network.) Thus, in our simulation, we did not take into account one of the benefits of SMECN over MECN, that it stops earlier in the neighbor-search process. Since a node's available energy is decreased after each packet reception or transmission, nodes in the simulation die over time. After a node dies, the network must be reconfigured. In [12], this is done by runinng MECN periodically. In the full paper, we present a protocol that does this more efficiently. In our simulations, we have used this protocol (and implemented an analogous protocol for MECN).

For simplicity, we simulated only a static network (that is, we assumed that nodes did not move), although some of the effects of mobility—that is, the triggering of the reconfiguration protocol—can already be observed with node deaths.

In this setting, we were interested in network lifetime, as measured by two metrics: (1) the number of nodes still alive over time and (2) the number of nodes still connected to the sink.

Before describing the performance, we consider some features of the subnetworks computed by MECN and SMECN. Since the search regions will be circular with an omnidirectional antenna, Theorem IV.4 assures us that the network used by SMECN will be a subnetwork of that used by MECN, but it does not say how much smaller it will be. The initial network in a typical execution of the MECN and SMECN is shown in Figure 3. The average number of neighbors of MECN and SMECN are 3.64 and 2.80 respectively. Thus, each node running MECN has roughly 30% more links than the same node running SMECN. This makes it likely that the final power setting computed will be higher for MECN than for SMECN. In fact, our experiments show that it is roughly 49% higher, so more power will be used by nodes running MECN when sending messages. Moreover, AODV is unlikely to find routes that are as energy efficient with MECN.



(a) MECN (b) SMECN

Fig. 3. Initial network computed by MECN and SMECN.

As nodes die (due to running out of power), the network topology changes due to reconfiguration. Nevertheless, as shown in

Figure 4, the average number of neighbors stays roughly the same over time, thanks to the reconfiguration protocol.

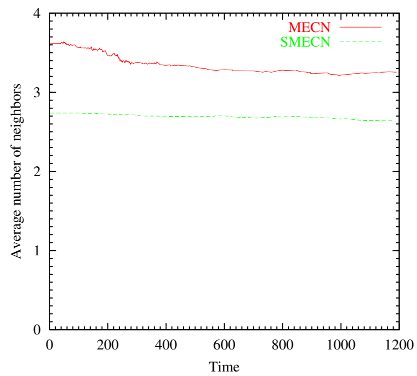
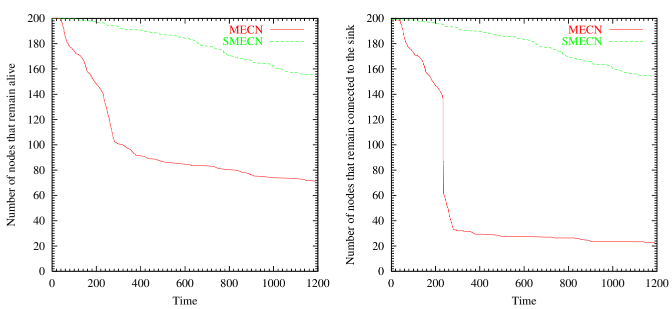


Fig. 4. Average number of neighbors over time.

Turning to the network-lifetime metrics discussed above, as shown in Figure 5, SMECN performs consistently better than MECN for both. The number of nodes still alive and the number of nodes still connected to the sink decrease much more slowly in SMECN than in MECN. For example, in Figure 5(a), at time 1200, 64% of the nodes have died for MECN while only 22% of the nodes have died for SMECN.



(a) Number of nodes that remain alive

(b) Number of nodes that remain connected to the sink

Fig. 5. Network life time with respect to two different metrics.

Finally, we collected data on average energy consumption per node at the end of the simulation, on throughput, and on end-to-end delay. MECN uses 63.4% more energy per node than SMECN. SMECN delivers more than 127% more packets than MECN by the end of the simulation, MECN's delivered packets have an average end-to-end delay that is 21% higher than SMECN. Overall, it is clear that the performance of SMECN is significantly better than MECN. We did not simulate the performance of the network in the absence of an algorithm designed to conserve power (This is partly because it was not clear what power to choose for broadcasts. If the maximum power is used, performance will much worse. If less power is used, the network may get disconnected.) However, these results clearly show the advantages of using an algorithm that increases energy efficiency.

VI. Conclusion

We have proposed a protocol SMECN that computes a network with minimum-energy than that computed by the protocol MECN of [12]. We have shown by simulation that SMECN performs significantly better than MECN, while being computationally simpler.

As we showed in Proposition IV.2, in the case of a circular search space, SMECN computes the set E_2 consisting of all edges that are not 2-redundant. In general, we can find a communication network with the minimum-energy property that has fewer edges by discarding edges that are k-redundant for $k > 2$. Unfortunately, for u to compute whether an edge is k-redundant for $k > 2$ will, in general, require information about the location of nodes that are beyond u's broadcast range. Thus, this computation will require more broadcasts and longer messages on the part of the nodes. There is a tradeoff here; it is not clear that the gain in having fewer edges in the communication graph is compensated for by the extra overhead involved. We plan to explore this issue experimentally in future work.

VII. Acknowledgments

We thank Volkan Rodoplu at Stanford University for his kind explanation of his MECN algorithm, his encouragement to publish these results, and for making his code publicly available. We thank Victor Bahl, Yi-Min Wang, and Roger Wattenhofer at Microsoft Research for helpful discussion.

References

[1] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (DREAM). In *Proc. Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 76–84, 1998.

[2] K. Calvert, M. Doar, and E. W. Zegura. Modeling internet topology. *IEEE Communications Magazine*, 35(6):160–163, June 1997.

[3] CMU Monarch Group. Wireless and mobility extensions to ns-2. http://www.monarch.cs.cmu.edu/cmu-ns.html, October 1999.

[4] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless micro-sensor networks. In *Proc. IEEE Hawaii Int. Conf. on System Sciences*, pages 4–7, January 2000.

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

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

[7] V. Paxson and S. Floyd. Wide-area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[8] V. Paxson and S. Floyd. Why we don't know how to simulate the internet. *Proc. 1997 Winter Simulation Conference*, pages 1037–1044, 1997.

[9] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999.

[10] VINT Project. The ucb/lbnl/vint network simulator-ns (version 2). http://www.isi.edu/nsnam/ns.

[11] T. S. Rappaport. *Wireless communications: principles and practice*. Prentice Hall, 1996.

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

[13] E. W. Zegura, K. Calvert, and S. Bhattacharjee. How to model an internetwork. In *Proc. IEEE Infocom*, volume 2, pages 594–602, 1996.