Distributed Energy-conserving Routing Protocols

Qun Li, Javed Aslam, and Daniela Rus
Department of Computer Science
Dartmouth College
Hanover, NH 03755
{liqun, jaa, rus}@cs.dartmouth.edu

Abstract

This paper discusses several distributed power-aware routing protocols in wireless ad-hoc networks (especially sensor networks). We seek to optimize the lifetime of the network. We have developed three distributed power-aware algorithms and analyzed their efficiency in terms of the number of message broadcasts and the overall network lifetime modeled as the time to the first message that can not be sent. These are: (1) a distributed min Power algorithm (modeled on a distributed version of Dijkstra's algorithm), (2) a distributed max-min algorithm, and (3) the distributed version of our the centralized online $max\text{-}min\ \ zP_{min}$ algorithm presented in [12]. The first two algorithms are used to define the third, although they are very interesting and useful on their own for applications where the optimization criterion is the minimum power, respectively the maximum residual power. The distributed $max\text{-}min\ \ zP_{min}$ algorithm optimizes the overall lifetime of the network by avoiding nodes of low power, while not using too much total power.

1 Introduction

The proliferation of low-power analog and digital electronics has created huge opportunities for the field of wireless computing. It is now possible to deploy hundreds of devices of low computation, communication and battery power. They can create ad hoc networks and be used as distributed sensors to monitor large geographical areas, as communication enables for field operations, or as grids of computation. These applications require great care in the utilization of power. The power level is provided by batteries and thus it is finite. Every message sent and every computation performed drains the battery.

Several metrics can be used to optimize power-routing for a sequence of messages. Minimizing the energy consumed for each message is an obvious solution that locally optimizes the power consumption. Other useful metrics include minimizing the variance in power across computers, minimizing the ratio of cost/packet, and minimizing the maximum node cost. A drawback of these metrics is that they focus on individual nodes in the system instead of the system as a whole. Therefore, routing messages according to them might quickly lead to a system in which nodes have high residual power but the system is not connected because some critical nodes have been depleted of power. We argue that it is advantageous to use a global metric by maximizing the lifetime of the network. This can be modeled as the time to the earliest point at which a message cannot be sent. We can show that for a network that optimizes the residual power of the system, the failure of the first node is equivalent to network partitioning. Our metric is very useful for ad-hoc networks where each message is important and the networks are sparsely deployed.

We assume that the power levels of all the nodes in the system are known and the message sequence is unknown. We also know the topology of the network (we have developed an algorithm for discovering this topology that uses $n-1$ messages for a network with n nodes.) If a host needs power e to transmit a message to another host which is distance d away, the power consumption for sending this message is $e = kd^c + a$, [7] where k and c are constants for the specific wireless system (typically $2 \leq c \leq 4$) and a is the electronics energy. Thus, we can model this problem as a weighted graph, where vertices correspond to hosts and weighted edges correspond to communication costs. We seek to find the best way to route each message as it arrives, so as the maximize the lifetime of the network.

This problem is different from the maximal network flow problem although there are similarities. The classical network flow problem constrains the capacity of the edges instead of limiting the capacity of the nodes. If the capacity of a node does not depend on the distances to neighboring nodes, our problem can also be reduced to maximal network flow. The maximal number of messages sustained by a network from the source nodes to the sink nodes can be formulated as linear programming.

In [12] we prove that no online algorithm for power-aware message routing has a constant competitive ratio in terms of the lifetime of the network or the number of messages sent. Guided by this theoretical result, we develop, analyze, and implement an approximation algorithm we call $max\text{-}min \quad zP_{min}$ and show that in practice this algorithm has a very good competitive ratio [12]. Our algorithm optimizes two criteria: (1) computing a path with minimal power consumption P_{min}; and (2) computing a path that maximizes the minimal residual power in the network. Neither criterion alone is sufficient for a good practical solution. There is a tradeoff between minimizing the total power consumption (which may drain critical nodes) and maximizing the minimal residual power of the network (which may use too much total power). This tradeoff is measured by parameter z which can be computed adaptively as described in [12].

The message paths computed by the $max\text{-}min \quad zP_{min}$ algorithm avoid the nodes with low residual power while choosing a low power consumption path. The algorithm discards all the routes that have high power consumption (greater than z times of the minimal power consumption to the base), and finds the route with the maximal minimum residual power in the remaining graph. This algorithm is centralized in the sense that every node must know the remaining powers of all nodes and the power consumption to transmit a packet along any two nodes in the network. We have shown empirically that this algorithm has good performance.

The algorithm $max\text{-}min \quad zP_{min}$ has the great advantage of not relying on the message sequence but the disadvantage of being centralized and requiring knowledge of the power level of each node in the system. These are unrealistic assumptions for field applications, for example involving sensor networks, where the computation is distributed and information localized. Our distributed version of the $max\text{-}min \quad zP_{min}$ algorithm has the flavor of the distributed Bellman-Ford algorithm. The protocol requires n^2 message broadcasts to find all the max-min zPmin paths from each sensor to the base station. In order to reduce communication, we add a waiting time prior to each broadcast. In this method, some of the messages that travel along suboptimal paths are suppressed. Only the messages that travel along the best paths end up being broadcast. The number of message broadcasts required to find the best paths to the base station for all the nodes are reduced to n.

The running time of our distributed algorithm can be improved in several ways by using approximations. We present the theoretic analysis that leads to these algorithms and experimental simulation results.

The structure of the paper is as follows. We first present how to use the waiting time to reduce the message broadcasts in minimal power consumption routing protocol and max-min routing protocol in section 2 and 3. Then we describe the distributed $max\text{-}min \quad zP_{min}$ path algorithm in section 4 and 5. Section 6 concludes the paper.

2 Related Work

We are inspired by exciting recent results in ad-hoc networks and in sensor networks. Most previous research on ad-hoc network routing [10, 15, 11] focused on the protocol design and performance evaluation in terms of the message overhead and loss rate. This previous work focused on how to find the correct route efficiently, but did not consider optimizing power while sending messages.

Singh et al. [16] proposed power-aware routing and discussed different metrics in power-aware routing. Some of the ideas in this paper are extensions of what that paper proposed. Minimal energy consumption was used in [14]. Stojmenovic and Lin proposed the first localized power-aware algorithm in their paper series [17]. Their algorithm is novel in combining the power and cost into one metric and running only based on the local information. Chang and Tassiulas [2] also used the combined metric to direct the routing. Their algorithm is proposed to maximize the lifetime of a network when the message rate is known. Their main idea, namely to avoid using low power nodes and choose the short path at the beginning, has inspired the approach described in this paper. We also use the same formula to describe the residual power fraction. The work presented in this paper is different from these previous results in that we develop online, hierarchical, and scalable algorithms that do not rely on knowing the message rate and optimize the lifetime of the network. Energy efficient MAC layer protocols can be found in [5, 4, 22]. Wu et al.[19] proposed the power-aware approach in dominating set based routing. Their idea is to use rules based on energy level to prolong the lifetime of a node in the refining process of reducing the the number of nodes in the dominating set

Another branch of the related work concerns optimizing power consumption during idle time rather than during the time of communicating messages [21, 3].Their protocols put some nodes in the network into sleep mode to conserve energy, while at the same time maintain the connectivity of the network to ensure communication. In a related work [19, 20], Wu and Stojmenovic give an elegant solution by using connecting dominating sets, which generalize the idea of maintaining a connected network while keeping most of the nodes in sleeping mode. This work is complementary to the results of the idle time power conservation optimizing methods. Combined, efficient ways for dealing with idle time and with communication can lead to powerful power management solutions.

Work on reducing the communication overhead in broadcasting task [18] bears similarity with our approach to reducing the message broadcasting in routing application. In Stojmenovic et al.'s paper, a node will rebroadcast a message only if there are neighbors who are not covered by the previous broadcasts. In contrast, our distributed algorithms eliminate the message broadcasts that are useless by discerning them with the message delay. As a result, in some algorithms we proposed, we can get a constant message broadcasts for each node.

Related results in sensor networks include [13, 9, 6, 8]. The high-level vision of wireless sensor networks was introduced in [13, 1]. Achieving energy-efficient communication is an important issue in sensor network design. Using directed diffusion for sensor coordination is described in [9, 6]. In [8] a low-energy adaptive protocol that uses data fusion is proposed for sensor networks. Our approach is different from the previous work in that we consider message routing in sensor networks and our solution does not require to know or aggregate the data transmitted.

3 Power Consumption Model

Power consumption in ad-hoc networks can be divided into two parts: (1) the idle mode and (2) the transmit/receive mode. The nodes in the network are either in idle mode or in transmit/receive mode at all time. The idle mode corresponds to a baseline power consumption. We instead focus on studying and optimizing the transmit/receive mode. When a message is routed through the system, all the nodes with the exception of the source and destination receives a message and then immediately relay it. Because of this, we can view the power consumption at each node as an aggregate between transit and receive powers which we will model as one parameter as described below.

More specifically, we assume an ad-hoc network that can be represented by a weighted graph $G(V, E)$. The vertices of the graph correspond to computers in the network. They have weights that correspond to the computer's power level. The edges in the graph correspond to pairs of computers that are in communication range. Each weight between nodes is the power cost of sending a unit message[1] between the two nodes.

Suppose a host needs power e to transmit a message to another host who is d distance away. We use the model of [7, 8] to compute the power consumption for sending this message:

$$e = kd^c + a,$$

where k, c and a are constants for the specific wireless system (usually $2 \leq c \leq 4$). We focus on networks where power is a finite resource. Only a finite number of messages can be transmitted between any two hosts. The algorithms proposed in this paper below use this formula to characterize the power consumption of sending a message.

4 Distributed Power-aware Routing with $max\text{-}min\ \ zP_{min}$

In this section we introduce three new algorithms: a distributed minimal power algorithm, a distributed

[1] Without loss of generality, we assume that all the messages are unit messages. Longer messages can be expressed as sequences of unit messages.

max-min power algorithm, and the distributed max-$min \quad zP_{min}$ power-aware algorithm. The first two algorithms are used to define the third, although they are very interesting and useful on their own for applications where the optimization criterion is the minimum power, respectively the maximum residual power.

4.1 A Distributed Minimal Power Algorithm

We can develop a distributed version of Dijkstra's algorithm that is guaranteed to be a minimal-power routing path algorithm by giving messages variable propagation delays. The idea is to have messages traveling along short paths move faster than messages traveling along long paths. Thus, messages traveling along longer paths will arrive faster than messages traveling along longer paths—that is, the algorithm will select the shortest paths. In this case, the Dijkstra distance corresponds to power-consumption.

We can implement this idea by augmenting each message with a a record of how far it traveled from the base to the current node. This information is represented by a variable attached to the message that measures the cost (distance representing power consumption). Algorithm 1 the resulting minimal power path algorithm, which represents a distributed version of Dijkstra's algorithm.

We continue this section by arguing that Algorithm 1 produces the minimal power-consumption path for each node. Furthermore, the running time of the algorithm is proportional to the longest shortest distance from the base node to any node.

We first examine a special case—when messages are time-sorted in the following sense. Suppose two messages carrying "distance" values v_1 and v_2 arrive at the same node at time t_1 and t_2. If for any two messages with $v_1 < v_2$, we have $t_1 < t_2$, the messages are *time-sorted.* Let n be the number of nodes in the network. In order to keep our proof simple, we assume that message transmission is instantaneous—this restriction can be relaxed.

Theorem 1 *If the messages are time-sorted, then Algorithm 1 requires $O(n)$ broadcasting messages (O(1) for each node).*

Proof: Let the message value of a message be the distance from the base station to the current node. Since the messages are time-sorted, the earliest message must carry the shortest distance from the base station to the current node. By line 9 of the algorithm, this message will be broadcast only once after the t_B waiting period has been completed. ♠

Algorithm 1 *Minimal Power Path.* The input consists of a network system in which each node can determine its location and its power level. The output is the minimal-power routing table at each node (with respect to communicating to the base.) The algorithm uses the following parameters: η is the unit power for transforming the power level into waiting time; P_A is the total power consumption of the optimal path found so far from A to the base node; $e(A, B)$: the power consumption of sending one message from A to B directly; t_B: the earliest time for B to broadcast the routing message; N_B: the route of node B.

1: **Initialization**; may not be necessary
2: Handshaking among neighbors; each node broadcasts its id, its position, and its current power level
3: $P_B = \infty, t_B = \infty$
4: **if** I am base station **then**
5:     initiate the message broadcasting
6: **else if** I am not base, say my id is B **then**
7:     Receive message (A, P_A); get the sender id A and P_A from the message
8:     Compute $P_B = min(P_A + e(A, B), P_B)$ and $t_B = min(t_B, \eta P_B)$
9:     **if** $P_B == P_A + e(A, B)$ **then**
10:         $N_B = A$
11:     **end if**
12:     Wait till the current time is t_B, broadcast the message (B, P_B) to its neighbors, and stop
13: **end if**

In Algorithm 1, the messages are not time-sorted. However, the messages become time-sorted if we consider the broadcast time of a node as the message arrival time (because of the delays enforced by the algorithm) and by Theorem 1, Algorithm 1 gives the shortest path within $O(n)$ broadcasts.

Note that the performance of our algorithm depends on the granularity at which we can measure power. Let the smallest measurement unit of the power consumption or the tolerable measurement unit be s. The parameter η, which can be chosen as the smallest time unit a node can distinguish, is the waiting time that corresponds to the distance s. The running time of Algorithm 1 is proportional to $1/s$ and to the size of the largest minimal power path. A large value for s results in a fast running time, but at the expense of precision. Say two messages that travel along paths with power consumption of P and $P + s_1$ (where $s_1 < s$) arrive at the same node in an interval less than η. The node may not distinguish them because the time difference is too small. Therefore, the running time is dependent on the precision of the required power consumption measurement. A better running time, can be obtained by allowing a low measurement precision, that is, a large unit power consumption η.

Algorithm 2 summarizes our ideas for improving the performance.

We assume the maximal minimal power consumption from the base station to any node in the network P. Let's divide $[0, P)$ into m slots, $[0, P/m)$, $[P/m, 2P/m)$, $\cdots$, $[iP/m, (i + 1)P/m)$, $\cdots$, $[(m - 1)P/m, P)$. When a node receives a message with value v, it first finds the i^{th} slot such that $iP/m \leq v < (i + 1)P/m$, waits till time $i\delta$, and then broadcasts the message to its neighbors. The running time of the algorithm ($m\delta$) is proportional to m and the parameter δ, which is the time interval corresponding to P/m.

We can choose δ to be large enough that any message traveling from the base station to any node in the network along a minimal power path with total message processing time $\epsilon < \delta$. (That is, the sum of the message processing time at each node on the minimal power path is less than δ).

Theorem 2 *For Algorithm 2, the number of messages broadcast by each node is no greater than the maximal number of paths from the base to a node with the power consumption in the same slot as that of the minimal power path (that is, $[iP/m, (i + 1)P/m)$ in which the minimal power consumption lies).*

Proof:

Consider a message arriving at node A and scheduled to be broadcast in the slot $[i\delta, (i + 1)\delta)$.

1. The message traveling along the minimal power path arrives at A at some time point before $i\delta + \epsilon$ since we assume the total message handling time (including message buffering, queuing, and propagation) is less than ϵ.

2. A message traveling along a path with power no less than $(i + 1) \cdot \frac{P}{m}$ will not be scheduled to be broadcast because the node stops broadcasting at time $(i + 1)\delta$.

3. There is no path with power consumption less than $i \cdot \frac{P}{m}$ to that node, so no message can be broadcast before $i\delta$ by that node.

4. Thus, only the messages traveling along the paths with power in the range of $[P_{min}, (i + 1)\delta)$ can be scheduled to broadcast.

♠

Theorem 3 *Algorithm 2 gives the minimal power consumption route for each node.*

Proof:

The message traveling along the minimal power path arrives at A at some time point before $i\delta + \epsilon < (i+1)\delta$ since we assume the total message handling time (including message buffering, queuing, and propagation) is less than ϵ. There is no path with power consumption less than $i \cdot \frac{P}{m}$ to that node, so no message cannot be broadcast before $i\delta$ by that node.

Thus, the message traveling along the minimal power path will be broadcast at each node. Then each node can look at the power consumption value carried by the message and set the node who broadcast the message as its route. ♠

4.2 A Distributed Max-Min algorithm

Minimal power path algorithm does not consider the residual powers of nodes when compute the route. Although a packet is routed along the minimal power path, some nodes on that path may be saturated very quickly. An alternative is to use the nodes with high power and avoid the nodes that are almost saturated, which leads to the max-min path for packet routing.

The max-min path is defined as the route from a node to the base on which the minimal residual power of the nodes is maximized among all the routes. The minimal residual power of a path $p(c, d)$ is $c = a_1, a_2, \cdots, a_k = d$, as $m_{p(c,d)} = min_{i=1}^{n-1} \frac{P(a_i) - e(a_i, e_{i+1})}{P(a_i)}$, and the max-min value $F_{(c,d)} = max_{all\ p(c,d)} m_{p(c,d)}$. If there may be multiple routes with the same max-min residual power, we can resolve ties arbitrarily.

Max-min paths can be found by using a modified version of the distributed Bellman-Ford algorithm. Upon computing a new max-min value, each node broadcasts it. The neighbors compute their max-min value according to the new incoming value, and broadcast the result only if the value is changed. The number of message broadcasts may be $O(n^3)$ as in the case of distributed Bellman-Ford algorithm.

To reduce the message broadcasts, we employ the same method as in Section 4.1 and add a variable waiting time on each node, which controls when the node broadcasts. Algorithm 3 summarizes the resulting protocol. We assume all the nodes are synchronized well, so they can decide locally the global time. Thus, a global clock is not needed to make this protocol work.

The max-min approximation, Algorithm 3 considers the maximal residual power fraction of all nodes in the network F_{max} split into m slots $([0, F_{max}/m), [F_{max}/m, 2F_{max}/m), \cdots, [iF_{max}/m, (i+1)F_{max}/m), \cdots, [(m-1)F_{max}/m, F_{max}).)$. The m slots are mapped to consecutive Δ long time slots $(s_1, s_2, \cdots, s_m.)$ In s_i the algorithm will find all the nodes whose max-min values are in slot $[(i-1)F_{max}/m, iF_{max}/m]$. The nodes found in the earlier slots have higher max-min values in the later slots.

We assume that the base has the maximal max-min value in the beginning of the algorithm. Thus, the base initiates the algorithm in the first slot s_1. Upon receiving the max-min values from the neighbors, nodes update their max-min value. Nodes wait until the time slot corresponding to the current max-min value, and the broadcast the value to its neighbors. If the node receives a new incoming value in some slot, say s_i, and finds that its max-min value should also be broadcast in this time slot, the broadcast is immediate. Thus, the nodes with max-min values in $[(i-1)F_{max}/m, iF_{max}/m)$ will be found as the messages go around the whole network.

If all the nodes have synchronized clocks, this algorithm performs $O(1)$ message broadcasts for each node. Otherwise, the base must initiate a synchronized broadcast to all the nodes to start a new slot and the number of broadcasts per node becomes $O(m)$.

Since each node broadcasts at most m messages, the running time of the algorithm is $m\delta$ where δ is the time for each round, which is at most n times the per message handling time. Furthermore, we can prove the following result using induction.

Theorem 4 *For each node, the algorithm gives a route with the minimal residual power fraction F, such that F and F^m are in the same slot where F^m is the max-min power fraction of the route from the base to that node. Then we have $|F - F^m| \leq F_{max}/m$.*

Proof: We use induction. In the first round, the maximal max-min value is broadcast by the base node. Each node that has the max-min value in the slot will broadcast the message.

For any node B with max-min value F_B^m in slot i, it is impossible for B to broadcast its value in slots before i. That is, F_B must be no greater than F_B^m, the actual max-min value of node B. This can be derived by examining the computation of F_B.

Suppose each node who finishes broadcast has F and F^m in the same slot. For any node B whose max-min value is in slot i, let A be the upstream node on the max-min path from the base to B. If B broadcasts its max-min value before A, then B can determine A's slot. Otherwise, A must broadcast its max-min value before B and B will hear the max-min value of A. Thus, from the algorithm, we have (see Algorithm 3) $min(F_A^m, \frac{P(A)-e(A,B)}{P(A)}) = F_B^m \geq F_B \geq min(F_A, \frac{P(A)-e(A,B)}{P(A)})$. From step (3), we know $min(F_A^m, \frac{P(A)-e(A,B)}{P(A)})$ and $min(F_A, \frac{P(A)-e(A,B)}{P(A)})$ are in the same slot, so we know F_B and F_B^m are in the same slot. ♠

We can improve Algorithm 3 using binary search. The running time can be reduced to $\delta \log m$, but the number of total messages sent is $n \log m$. The key idea is to split the range $[0, F_{max})$ in two, $[0, F_{max}/2)$ and $[F_{max}/2, F_{max})$. In the first epoch, the algorithm tries to find all the nodes whose max-min values are in the higher half. In the second epoch, we split each range into two halves to get four ranges. The algorithm finds in parallel all the nodes whose max-min values are in the higher half of each range, etc.

4.3 Distributed max-min zP_{min}

We now derive the distributed version of the centralized online max-min zP_{min} algorithm. Like in the centralized case, our motivation is to define a routing algorithm that optimizes the overall lifetime of the network by avoiding nodes of low power, while not using too much total power. There is a tradeoff between minimizing the total power consumption and maximizing the minimal residual power of the network. We propose to enhance a max-min path by limiting its total power consumption.

Recall that the network is described as a graph in which each vertex corresponds to a node in the network, and only two nodes within the transmission ranges of each other have an edge connecting them in the graph. The power level of a node a is denoted as $P(a)$, and the power consumption to send a message unit to one of its neighbors b is denoted as $e(a, b)$. Let $s(a)$ be the power consumption for sending a unit message from a to the base station along the least power consumption path. Let $r(a)$ be the minimum residual power fraction of the nodes on a's mmz path. Let P_a be the power consumption along the mmz path.

An mmz path has the following properties:

1. it consists of two parts: the edge connecting a to one of its neighbors and the mmz path of that neighbor;
2. its total power consumption is less then or equal to $z \cdot s(a)$; and
3. among all those paths defined by (1) and (2), the max-min value of the mmz path is maximized.

More precisely, $p(a)$, the mmz path of node a, is: (1) a simple path from a to the base station; (2) $P_a < z \cdot s(a)$; and (3) $p(a) = (a, b) \cup p(b)$, where b is a's neighbor such that for any other neighbor c $r(a) = min(r(b), \frac{P(a)-e(a,b)}{P(a)}) \geq min(r(c), \frac{P(a)-e(a,c)}{P(a)})$.

Theorem 5 *There is one node b_j such as $e(a, b_j) + P_{b_j} \leq z \cdot s(a)$.*

Proof: Use induction. The case for base is obvious. Let b_j be the node on the shortest path from a to the base. $P_{b_j} \leq z \cdot s(b_j)$ and $e(a, b_j) + s(b_j) = s(a)$. So $e(a, b_j) + P_{b_j} \leq e(a, b_j) + z \cdot s(b_j) \leq z \cdot (e(a, b_j) + s(b_j)) = z \cdot s(a)$ ♠

Note that $s(a)$ can be computed easily by using $s(a) = min\ \{s(b) + e(a, b)\}$ where b is a's neighbor.

The definition of the mmz path actually gives a constructive method for computing incrementally the mmz path by keeping track of $s(node), r(node), p(node)$ of each node n, because the computation only depends on these values at v's neighbors. Let $n(node)$ be the next node on the path $p(node)$. The resulting algorithm is shown as Algorithm 4. In the algorithm, the base station initiates the route exploration by broadcasting its route information ($s(base)$, $r(base)$, and $n(base)$ to its neighbors). When a node's route information changes, it broadcasts its updated information. This broadcast triggers its neighbor nodes to check if their route information changes. Every time the route information of a node changes the information is broadcast until the system achieves equilibrium.

In our distributed version of the Max-min zP_{min} algorithm, we expect $O(n^3)$ messages broadcast totally in the worst case.

It is possible to improve the number of message broadcasts by using timing variables to suppress some of the messages. Two specific approaches are

- In the max-min part, let the message carry the total power consumption on the path, and use the power consumption to decide if the max-min value should be accepted.
- In the minimal power path part, incorporate the max-min value in the waiting time.

4.4 Experiments in simulation

We have implemented the distributed algorithms outlined in this section and compared the performance of the distributed *max-min* zP_{min} algorithm. Furthermore, we compared this algorithm against a Greedy-style distributed algorithm.

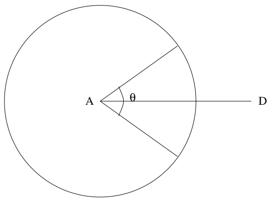

Figure 1. Search range in the greedy routing

In greedy routing implementation, nodes exchange power information with their neighbors periodically. When there is a message at A for destination D, A finds the node B with the highest power level within its transmission range, in a cone centered at A with angle θ, which is bisected by line AD, and sends message to B.

Figure 2 shows the performance comparison of the distributed *max-min* zP_{min} algorithm and the distributed greedy algorithm. We conclude that *max-min* zP_{min} outperforms a simple greedy algorithm for all values of z, and for some values of z the distributed *max-min* zP_{min} doubles the performance. More specifically, peak of the *max-min* zP_{min} algorithm is obtained when z=1.2, and the number of messages sent is 26912. When z=2, the number message sent is 18935. The distributed greedy algorithm sent

14278 messages in total. The performance improvement is 88.4% in the best case when z=1.2 and 32.61% in the worst case.

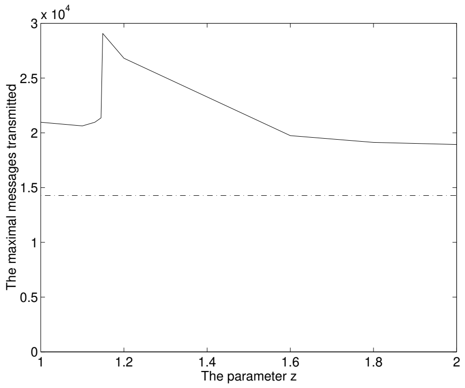

Figure 2. The performance comparison of distributed max-min zP_{min} algorithm and greedy algorithm. The dashed line shows the performance of the greedy algorithm and the solid line shows the performance of the max-min zP_{min} algorithm. The network includes 100 nodes. The network space is $100 * 100$, the transmission range is 20, the power consumption formula is $E = 2 * 10^{-6} * d^3$. The greedy algorithm uses a $\theta = \pi/3$. The routing protocol is run after every 100 messages. The neighbor information update in the greedy algorithm is updated every 100 messages.

We are currently collecting empirical data on the tradeoffs between the various parameters we introduced to describe our algorithms.

5 Conclusion

We have described several localized distributed algorithms for power-aware routing of messages in large networks dispersed over large geographical areas. Our algorithms do not know ahead of time the message sequence and are thus online. In most applications that involve ad-hoc networks made out of small hand-held computers, mobile computers, robots, or smart sensors, battery level is a real issue in the duration of the network. Power management can be done at two complementary levels (1) during communication and (2) during idle time. We believe that optimizing the performance of communication algorithms for power consumption and for the lifetime of the network is a very important problem.

It is hard to analyze the performance of online distributed algorithms that do not rely on knowledge about the message arrival and distribution. This assumption is very important as in most real applications the message patterns are not known ahead of time. In this paper we have presented and analyzed three types of distributed algorithms and shown empirically that our new algorithm called distributed max-min zP_{min} obtains significant performance improvement on the network lifetime relative to a simple distributed greedy-style algorithm. Much more evaluation needs to be fully understand the applicability of distributed max-min zP_{min}. We are currently undergoing these experimental studies which are focused on parameter tradeoffs in these algorithms.

References

[1] Jon Agre and Loren Clare. An integrated architeture for cooperative sensing networks. *Computer*, pages 106 – 108, May 2000.

[2] Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000.

[3] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *7th Annual Int. Conf. Mobile Computing and Networking 2001*, Rome, Italy, July 2001.

[4] I. Chlamtac, C. Petrioli, and J. Redi. Energy-conserving access protocols for indetification networks. *IEEE/ACM Transactions on Networking*, 7(1):51–9, Feb. 1999.

[5] A. Chockalingam and M. Zorzi. Energy efficiency of media access protocols for mobile data networks. *IEEE Transactions on Communications*, 46(11):1418–21, Nov. 1998.

[6] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In *ACM MobiCom 99*, Seattle, USA, August 1999.

[7] Laura Maria Feeney and Martin Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *INFOCOM 2001*, April 2001.

[8] W. Rabiner Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient routing protocols for wireless microsensor networks. In *Hawaii International Conference on System Sciences (HICSS '00)*, Jan. 2000.

[9] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. of the Sixth Annual International Conference on Mobile Computing and Networks (MobiCOM 2000)*, Boston, Massachusetts, August 2000.

[10] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad-hoc wireless networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, pages 153 –181. Kluwer Academic Publishers, 1996.

[11] Y. B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of ACM/IEEE MOBICOM'98*, pages 66 – 75, 1998.

[12] Qun Li, Javed Aslam, and Daniela Rus. Online power-aware routing in wireless ad-hoc networks. In *MOBICOM*, pages 97–107, Rome, July 2001.

[13] G. J. Pottie and W. J. Kaiser. Wireless integrated newtork sensors. *Communications of the ACM*, 43(5):51–58, May 2000.

[14] Volkan Rodoplu and Teresa H. Meng. Minimum energy mobile wireless networks. In *Proc. of the 1998 IEEE International Conference on Communications, ICC'98*, volume 3, pages 1633–1639, Atlanda, GA, June 1998.

[15] Elizabeth Royer and C-K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. In *IEEE Personal Communication*, volume 6, pages 46 – 55, April 1999.

[16] S. Singh, M. Woo, and C. S. Raghavendra. Power-aware routing in mobile ad-hoc networks. In *Proc. of Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 181–190, Dallas, TX, Oct. 1998.

[17] Ivan Stojmenovic and Xu Lin. Power aware localized routing in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(11):1122–1133, November 2001.

[18] Ivan Stojmenovic, Mahtab Seddigh, and Jovisa Zunic. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(1):14–25, January 2002.

[19] J. Wu, F. Dai, M. Gao, and I. Stojmenovic. On calculating power-aware connected dominating set for efficient routing in ad hoc wireless networks. *IEEE/KICS Journal of Communications and Networks*, 4(1):59–70, March 2002.

[20] J. Wu, B. Wu, and I. Stojmenovic. Power-aware broadcasting and activity scheduling in ad hoc wireless networks using connected dominating sets. In *IASTED International Conference on Wireless and Optical Communication*, Banff, Canada, July 2002.

[21] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *7th Annual Int. Conf. Mobile Computing and Networking 2001*, Rome, Italy, July 2001.

[22] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *INFOCOM*, New York, NY, June 2002.

Algorithm 2 The second minimal power path algorithm. The input is a network in which each node can determine its location and its power level. The output is a routing table for each node. The parameters are P_A, the total power consumption of the optimal path found so far from A to the base node; $e(A, B)$, the power consumption of sending a message from A to B directly; and δ, the unit time corresponding to each power slot (P/m), used to transform the power level into waiting time; N_B: the route of node B.

1: **Initialization**; may not be necessary
2: Handshaking among neighbors: each node broadcasts its id, its position, and its current power level
3: The base initiates the message broadcasting
4: **if** I am not the base **then**
5:     Let my id be B
6:     $P_B = \infty$. Initial time is 0.
7:     Receive message (A, P_A); get the sender id A and the power P_A from the message
8:     Compute the new power $P_B = min(P_B, P_A + e(A, B))$, and find the proper slot $i = \lfloor m \cdot P_B/P \rfloor$
9:     **if** $P_B == P_A + e(A, B)$ **then**
10:         $N_B = A$
11:     **end if**
12:     Set waiting timer to $i\delta$ (i.e. the time point when a broadcast happens)
13:     **if** the current time is no less than the waiting time point **then**
14:         broadcast the message (B, P_B) to its neighbors, and clear the timer. ; We do that because there are may be several paths being broadcast to the node. But their time must be between $i\delta$ and $(i + 1)\delta$
15:     **end if**
16:     **if** the current time is $(i + 1)\delta$ **then**
17:         stop
18:     **end if**
19: **end if**

Algorithm 3 Distributed Max-min Approximation. The input is a network in which each node can determine its location and its power level. The output is a routing table at each node. The parameters are: $P(A)$, the current power level of node A; $e(A, B)$, the power consumption of sending one message from A to B directly; and δ, the unit time corresponding to each power slot (P/m) used to transform the power level into waiting time.

1: **Initialization**; may not be necessary
2: Handshaking among neighbors: each node broadcasts its id, its position, and its current power level
3: For each node B, let $F_B = 0$, B does the following for $i = m - 1, m - 2, \cdots, 1, 0$.
4: The base node initiates the search and broadcasts the maximal max-min value
5: **if** Node B receive a message $(A, P(A), F_A)$ from its neighbor A **then**
6:     According to the power level of A and the distance between A and B, compute $F_B = max(F_B, min(F_A, \frac{P(A)-e(A,B)}{P(A)}))$
7:     **if** $F_B == min(F_A, \frac{P(A)-e(A,B)}{P(A)})$ **then**
8:         $N_B = A$
9:     **end if**
10: **end if**
11: **if** $(i + 1)F_{max}/m > F_B \geq iF_{max}/m$ **then**
12:     the max-min value of B is found
13:     B broadcasts the message $(B, P(B), F(B))$, the next node in the routing table is A, stop
14: **end if**
15: After time δ, i=i-1; go to 5

Algorithm 4 Distributed $max\text{-}min\ zP_{min}$. The parameters are P^B_{min}, the minimal power consumption for node B to send a message to the base; P_B, the power consumption of the path discovered so far from the node to the base; $P(B)$, node B's current power level; F_B, the maximal min residual power level of the found route to base from node B; and N_B: the next node on B's found route. δ is an algorithm-dependent parameter; different implementations may have difference choices.

1: Find the minimal power consumption path for each node
2: The base node 0 initiates the route discovery
3: $P_0 = 0; F_0 = \infty; N_0 = 0$
4: Node 0 sends route discovery request to its neighbors
5: Each node B receives message from its neighbors $A_1, A_2, \cdots, A_k$
6: It waits for time δ, then compute: $P_B = min(P_{A_1} + e(B, A_1), P_{A_2} + e(B, A_2), \cdots, P_{A_k} + e(B, A_k))$ Find all the neighboring nodes such that $P_{A_i} + e(B, A_i) <= zP^{A_i}_{min}$ Among all those found neighbors, find the node with maximal $min(F_{A_k}, (P(B) - e(B, A_k))/P(B))$ Let the node be N_B and the min value be F_B
7: Broadcast the P_B and F_B to its neighbors
8: Repeat 3, 4 until the routing table gets to equilibrium