

# Routing for Network Capacity Maximization in Energy-constrained Ad-hoc Networks

Koushik Kar<sup>†</sup>

Murali Kodialam\*

T. V. Lakshman\*

Leandros Tassioulas<sup>‡</sup>

<sup>†</sup> ECSE Department  
Rensselaer Polytechnic Institute  
Troy, NY 12180, USA

\* Bell Labs  
Lucent Technologies  
Holmdel, NJ 07733, USA

<sup>‡</sup> ECE Department  
University of Maryland  
College Park, MD 20742, USA

**Abstract**— We present a new algorithm for routing of messages in ad-hoc networks where the nodes are energy-constrained. The routing objective is to maximize the total number of messages that can be successfully sent over the network without knowing any information regarding future message arrivals or message generation rates. From a theoretical perspective, we show that if admission control of messages is permitted, then the worst-case performance of our algorithm is within a factor of  $O(\log(\text{network size}))$  of the best achievable solution. In other words, our algorithm achieves a logarithmic competitive ratio. Our approach provides sound theoretical backing for several observations that have been made by previous researchers. From a practical perspective, we show by extensive simulations that the performance of the algorithm is very good even in the absence of admission control (the admission control being necessary only to prove the competitive ratio result), and that it also performs better than previously proposed algorithms for other suggested metrics such as network lifetime maximization. Our algorithm uses a single shortest path computation, and is amenable to efficient implementation. We also evaluate by simulations the performance impact of inexact knowledge of residual battery energy, and the impact of energy drain due to dissemination of residual energy information.

## I. INTRODUCTION

With the rapid increase in low cost sensor devices, there is an increased interest in the deployment of ad-hoc wireless sensor networks. This is reflected in the recent spurt of research interest in this area [1], [4]. Low cost sensors typically have low battery life and therefore conserving battery energy is a prime consideration in these networks. Since some battery energy is depleted for each message transmission, it is necessary to use energy aware and energy conserving routing algorithms. This paper presents a new distributed algorithm for on-line message routing in energy-constrained wireless ad-hoc networks.

Our routing objective is to maximize the total number of messages successfully carried by the network (hereafter referred to merely as *network capacity*) without making any assumptions on future message arrivals. This is equivalent to minimizing the number of messages that cannot be routed for an (unknown) sequence of message routing requests. This is different from a previously studied objective [1], [4] of maximizing the *network lifetime* (i.e., the time before the first message loss). Despite the differing objective, the algorithm

developed in this paper outperforms the existing algorithms even with respect to optimizing the network lifetime.

The on-line algorithm we develop is a simple algorithm using only one shortest path computation. We show that if admission control of messages is permitted, our algorithm achieves a *competitive ratio* [8] that is logarithmic in the number of network nodes, i.e., its performance without knowledge of future message arrivals is in the worst case within a logarithmic factor of the best performance achievable by an off-line algorithm with complete information about messages to be transmitted. Admission control is necessary only for obtaining the competitiveness result. In practice, the algorithm works very well even without admission control, as can be seen from the simulation results. We use knowledge of residual battery energy at each node, and also consider the general setting where energy consumption for message transmission is dependent on the distance to the neighbor. Though the assumption on the knowledge of residual energies is used for proving the theoretical worst-case performance bounds, simulations show that the algorithm works well even with inexact knowledge of node residual energies.

Our approach uses ideas developed for competitive connection routing in wired networks [5]. However, the routing model for energy aware routing in wireless ad-hoc networks is different from that for connection routing in wired networks. In wired networks the constraint is on the consumption of link-bandwidth. In contrast, the routing problem in wireless ad-hoc networks is constrained by the battery energy of each node. It cannot be transformed to a link capacity constraint problem by node splitting since the energy expended in transmitting on different links can be different. However, as we show in this paper, the ideas used for wired connection routing can still be nicely applied to develop a new algorithm for energy aware capacity maximizing routing in wireless ad-hoc networks and to prove its competitive ratio.

As in [4], our algorithm ideally requires knowledge of the energy level of all other network nodes. This may become a problem in large networks. For this case, we show that periodic, geographically limited broadcasts of the energy level is sufficient to ensure that our algorithm's performance with partial energy information is close to that of the ideal case with complete energy information. The energy consumption for these neighborhood broadcasts is accounted for when

computing the network energy consumption in performance comparisons. An alternate use of the algorithm is to combine it with the zone-based approach developed in [4].

Note that we do not make any assumptions regarding message arrivals, unlike [1], [4] where it is assumed that the message arrival rates between different source-destination pairs have a regular pattern. Moreover, our algorithm is also computationally much simpler than existing algorithms. A detailed comparison of our approach with the previous approaches is provided in Section II.

The paper is organized as follows. The next section discusses related work. In Section III, we describe the energy-constrained routing problem formally, and outline our approach and objectives. In Section IV, we describe a capacity-competitive on-line algorithm for the energy-constrained routing problem. Section VI evaluates the performance of our algorithm on the basis of simulation results. We conclude in Section VII.

## II. RELATED WORK

There has been tremendous research interest in ad-hoc networks in the last few years. The early literature on ad-hoc networking primarily addressed the design of efficient routing algorithms but without optimization of the energy required to send messages. After the work of Singh et al. [10], there has been a growing literature on energy aware routing in ad-hoc wireless networks. Energy aware routing is particularly important for sensor networks where nodes often have a limited amount of battery energy.

Routing with the objective of maximizing the network lifetime has been addressed in [1], [4]. In [1], the authors consider the scenario where messages are generated at the sensor nodes at a known fixed rate, and formulate the lifetime maximization problem as a linear program. The authors propose some distributed heuristics that can be used to solve this linear program approximately. In [2], the authors observe that the structure of the linear program resemble that of a maximum concurrent flow problem, and use this fact to develop an algorithm that can solve the problem to any arbitrary degree of accuracy. The algorithms in [1], [2] can be used to determine how the traffic (generated at a constant rate) should be split amongst the different routes so as to maximize the lifetime of the network. Therefore in this case, the ad-hoc sensor network can solve the lifetime maximizing traffic splitting problem once at the beginning, and use the computed routes thereafter without any further reoptimization. Thus in the scenario considered in [1], [2], since the traffic generation rates are assumed to be constant and known, the network can solve the routing problem in an off-line manner.

Our work differs from [1], [2] in several aspects. Most importantly, we do not assume that the message generation rates are known, or that they are constant. In a vast majority of applications involving ad-hoc networks, the message generation rates can vary with time, and the message generation process is highly unpredictable. In other words, there is no a priori information (deterministic or statistical) on the future message generation (arrival) process. Whereas the approach in

[1], [2] cannot be used in that case, the algorithm proposed in this paper is very well suited for such a scenario. Also note that in this paper we are interested in optimizing the network capacity, and not the network lifetime as considered in [1], [2].

This paper is motivated by the network lifetime competitive ratio results in [4] for the max-min  $zP_{\min}$  algorithm developed by the authors. In [4], the authors propose an on-line message routing algorithm (which does not need to know the message generation rates) for the network lifetime maximization problem, and provide a competitive analysis for their algorithm. However, this approach has two drawbacks. Firstly, in the analysis, the authors assume that the messages are generated at a constant rate. (It is assumed that messages are generated cyclically, or in each interval of time the set of messages are the same.) Secondly, the competitive bound depends on values that are not given as input to the problem, and it is not clear how good the bound is. (For example, the competitive bound depends on the amount of residual energy left over in the network as well as the periodicity of the messages.) Further, the algorithm involves several shortest path computation invocations.

The capacity maximization algorithm developed in this paper uses only one shortest path computation, and the competitive ratio depends only on the input parameters to the problem. Further, from the experimental results, it seems to perform better than max-min  $zP_{\min}$  even for the lifetime maximization objective.

The problem of minimizing power consumption during idle times is addressed in [3]. Topology control of ad-hoc networks by adjustment of node power levels has been considered in [7], [6], [9]. We refer the reader to [4] for an excellent survey of the work in this area.

## III. THE ON-LINE ENERGY-CONSTRAINED ROUTING PROBLEM

In this section, we discuss the on-line routing problem in energy-constrained wireless networks. We first describe the system model, and then outline our routing objectives.

### A. System Model

A node in an ad-hoc sensor network can either be in idle state or in a transmit/receive mode. In this paper, we consider the energy (power) consumed at each node when it is in a transmit/receive mode. The ad-hoc network can be modeled as a graph  $G = (N, A)$  where  $N$  represents the set of nodes in the network and  $A$  the set of edges. We assume that there are  $n$  nodes and  $m$  edges in the graph. The energy consumed for transmitting a unit message along link  $(i, j) \in A$  is represented by  $e_{ij}$ . Node  $i \in N$  is assumed to have an initial energy reserve of  $E_i$ . Each message that has to be carried by the network has a source node that originates the message, and a destination node which the message has to reach through the multi-hop network. Let  $s_k$  and  $d_k$  represent the source and destination nodes for message  $k$ . Let  $l_k$  represent the length of message  $k$ . Therefore if message  $k$  is transmitted at node  $i$  along link  $(i, j)$  then the energy at node  $i$  decreases by the

quantity  $l_k e_{ij}$ . Each message is assumed to represent some unsplitable information and therefore has to be routed along a single path. In other words, a given message cannot be split across multiple paths. We assume that the messages are of different lengths. Though not necessary for the analysis, the energy  $e_{ij}$  that is consumed per message on link  $(i, j) \in A$  is usually proportional to some power of the distance between nodes  $i$  and  $j$ .<sup>1</sup> All messages that enter the system have to be routed at the time they arrive into the system. We do not assume to know any information about future arrivals to the network. We do not assume constant rates or any periodicity of the message pattern.

### B. Routing Objectives

We want to maximize the capacity of the network. In this paper, network capacity is defined to be the total amount of message data that is successfully carried by the network. If the entire message sequence is known ahead of time, then it is possible to develop an off-line routing algorithm to maximize the amount of data that is carried by the network. We use this off-line algorithm only as a benchmark to measure the performance of our routing algorithm for which messages arrive only one at a time. We would like our on-line algorithm (which does not have any knowledge of future message arrivals) to perform well compared to the off-line algorithm for all message arrival patterns. The *competitive ratio* of an on-line algorithm is defined to be the *worst case* ratio of the capacity of the off-line algorithm to the capacity of the on-line algorithm over all sets of inputs. Ideally we would like the competitive ratio to be a small constant. However, as shown in [4] and [5], this is not possible in general. In fact, if there is no control on the size of the messages then the competitive ratio can be as bad as  $O(n)$ . Intuitively, the on-line algorithm can be made to perform arbitrarily bad by an adversary who injects messages into the system if

- The size of the messages can be arbitrary.
- The algorithm is not allowed to perform admission control (i.e., the algorithm is not allowed to reject any message as long as there is a path available to carry it, no matter how much network resources is consumed by the message).

Therefore, to obtain the competitive ratio result, we permit admission control, i.e., the algorithm can occasionally reject messages that are determined to be too detrimental to the network's residual capacity. Fortunately, it turns out that neither of these assumptions are critical in practice. We show that even if the algorithm is run without the admission control option, its performance is excellent. Further from the experiments it can be seen that it outperforms the existing algorithms both with respect to network capacity as well as network lifetime.

## IV. A CAPACITY-COMPETITIVE ALGORITHM

In this section, we describe our on-line routing algorithm (CMAx), and derive its competitive ratio.

<sup>1</sup>Energy consumed in transmitting a message between two nodes that are  $d$  units apart is proportional to  $d^\kappa$ , where  $\kappa$  is a constant between 2 and 4.

### A. The Algorithm

We first define some notation needed to describe the algorithm. Let messages be indexed in the order in which they are generated. Let  $l_k$  denote the length of message  $k$ . Let  $s_k$  and  $d_k$  represent the source and the destination nodes, respectively, of the message  $k$ . Let  $E_i(k)$  denote the residual energy of node  $i$  at the time when message  $k$  is generated (but before it is routed). Note that according to our notation,  $E_i(1) = E_i$ . Let

$$\alpha_i(k) = 1 - \frac{E_i(k)}{E_i}.$$

Therefore  $\alpha_i(k)$  is the fraction of the energy of node  $i$  that is used at the time message  $k$  arrives. We will refer to the quantity  $\alpha_i$  as the *energy utilization* of node  $i$ . In the algorithm stated below,  $\lambda$  and  $\sigma$  are constants, chosen appropriately (we will discuss the choice of these constants later in the paper).

#### Algorithm CMAx

- Step 1. Consider routing message  $k$  on the network  $G$ . Eliminate all links  $(i, j) \in A$  for which  $e_{ij} > \frac{E_i(k)}{l_k}$  to form a reduced network.
- Step 2. Associate weights  $w_{ij}$  with each link  $(i, j)$  in the reduced graph, where  $w_{ij} = e_{ij}(\lambda^{\alpha_i(k)} - 1)$ .
- Step 3. Find the shortest path from  $s_k$  to  $d_k$  in the reduced graph with link weights  $w_{ij}$ , as defined in Step 2.
- Step 4. Let  $\gamma_k$  be the length of the shortest path found in Step 3 ( $\gamma_k = \infty$  if no path was found). If  $\gamma_k \leq \sigma$ , route the message along the shortest path, otherwise reject it.

Note that the computational complexity of the algorithm is determined by the shortest path computation (Step 3) and is  $O(m + n \log n)$ . In contrast, the algorithm in [4] involves performing as many as  $\log m$  shortest path computations.

According to the algorithm stated above, a message is routed along the shortest path, based on the weights  $w_{ij}$  (computed in Step 2), provided the shortest path length is not "too high". Note that the weight of an link  $(i, j)$ ,  $w_{ij}$ , increases with an increase in  $e_{ij}$ , the energy expended in traversing link  $(i, j)$ . Moreover,  $w_{ij}$  increases as the energy utilization of the transmitting node  $i$ ,  $\alpha_i$ , increases. This means that the algorithm tries to avoid links which require very high energy for transmission, and nodes where the residual energy fraction is low. The algorithms in [1] and [4] are also based on a similar intuition. The algorithm in [4] first determines the paths that consume minimum energy, and the path that maximizes minimum residual energy, independently. Then the algorithm, through a series of shortest path computations, determines a path that is good with respect to both these criteria. In our case, the weight function defined above captures both these effects and provides a provably good competitive ratio. The exponential dependence of the weight function on the energy utilization suggests that this plays a dominant role in the routing. Note that if the shortest path length is greater than a specified threshold  $\sigma$ , the message is rejected, even if there is a path with enough energy to accommodate it. Since the path length is an increasing function of time, this implies that for any particular source-destination pair, all messages

will be accepted till a certain instant of time, after which all messages will be rejected. As stated earlier, without this option to reject, an adversary can inject messages that consume too much resources destroying the competitive ratio of the algorithm. However, this is not of practical consequence when messages are generated at random or by an adversary who does not know the routing policy. In practical situations, rejecting a message when sufficient energy is available is usually unacceptable. Experimental results show that although setting  $\sigma$  to its theoretically determined value might result in improved capacity performance, setting the value of  $\sigma$  to infinity (i.e., the only reason for rejecting a message is insufficient energy) results in excellent performance both with respect to the capacity as well as the lifetime of the network.

### B. Competitive Bound

We now outline the competitive bound of CMAX. We define  $e_{\max} = \max_{(i,j) \in A} e_{ij}$ , and  $e_{\min} = \min_{(i,j) \in A} e_{ij}$ . Therefore,  $e_{\max}$  and  $e_{\min}$  are respectively the maximum and minimum energies expended by a message of unit length when traversing some link in the network. Let  $\rho = e_{\max}/e_{\min}$ , and let  $L(k)$  denote the total length of messages successfully routed by our algorithm till the arrival of message  $k$ . Let  $L_{opt}(k)$  be the total length of messages successfully routed by the optimal algorithm till the arrival of message  $k$ . We now formally state the competitiveness result for our algorithm.

*Theorem 1:* Let  $\lambda = 2(n\rho + 1)$  and  $\sigma = ne_{\max}$ . For all messages  $k$ , let

$$l_k \leq \frac{\min_{i \in N} E_i}{e_{\max} \log \lambda} \quad (1)$$

Then

$$\frac{L(k)}{L_{opt}(k)} \geq \frac{1}{1 + 2 \log \lambda} \quad \forall k.$$

The proof of this result is stated in Appendix I. The above result implies that the competitive ratio of CMAX is  $O(\log n\rho)$ . In the derivation of the result, we assume that all messages may not be of the same length. If all messages are of the same length, then the algorithm also provides a competitive ratio for the number of messages accepted.

## V. IMPLEMENTATION ISSUES

For routing messages, the algorithm CMAX has to know the topology of the network and the current energy utilization at all nodes in the network. The topology of the network includes all the nodes and links in the network as well as the energy used to transmit a unit message on each link in the network. The network topology can be learned by a node at the time it joins a network. Typically, the network topology is relatively static, and any changes in the topology (due to joining/leaving of nodes, nodes running out of energy etc.) can be disseminated throughout the network. However, the energy utilization of the nodes changes frequently. In small networks, disseminating this information through the network may not be expensive. However, frequent dissemination of global energy information may not be feasible in larger networks. In such cases, we can use one of the following two approaches.

### A. Zone Based Approach

In [4], the authors use a zone based approach for routing in large networks. This approach is based on aggregating information in zones and routing the message from one zone to another. Note that *the information required by CMAX and  $\max - \min zP_{\min}$  are identical*. Therefore one can use the same zone based approach to route with CMAX. In fact we can use Dijkstra's algorithm to compute the route instead of Bellman-Ford that has to be used in the case of approximate distributed version of  $\max - \min zP_{\min}$ .

In this paper, we use an alternate approach based on limited flooding.

### B. Limited Flooding Approach

Limited flooding approach requires that the algorithm be robust to imprecise energy information. Our experimental results show that this is indeed the case for the CMAX algorithm. The main idea in the limited flooding approach is to transform CMAX into a *hop-by-hop* routing algorithm like OSPF, which we call D-CMAX. In a hop-by-hop implementation, each node computes the shortest path to the destination of the message and forwards the message to the next hop node on this shortest path. This process is repeated until the message reaches the destination. Hop-by-hop routing algorithms are typically robust to errors in link weight information of distant links. We take advantage of this fact in the limited flooding approach. In this approach, each node periodically broadcasts its residual energy information to all nodes that are within a certain distance of that node. This is done by broadcasting with limited power so that it will be "heard" only by all nodes that are within some limited distance of the transmitting node. We define a  $(r, \gamma)$  local broadcast mechanism to be one in which a node  $i \in N$  transmits its energy level information to all nodes within a distance of  $r$  from itself, and the broadcast is done when its current energy level becomes less by a factor of  $\gamma$  from the last local broadcast energy value. While computing the next-hop node using the shortest path algorithm, a node  $i$  assumes that the residual energy at any other node  $j$  is the same as the value last broadcast by  $j$ . If node  $i$  is outside the broadcast radius of node  $j$  (i.e.,  $i$  cannot hear node  $j$ 's broadcast), then  $i$  just assumes that the energy of node  $j$  is at its initial level. For the experimental results presented, we assume that the values of  $r$  and  $\gamma$  are the same for all nodes. We also performed several experiments where the value of  $r$  and  $\gamma$  are node-dependent, and the results were similar to the ones shown below. Theoretically, due to the asymmetry of information at different nodes, there can be loops in the routing paths. Loop avoidance can be done by finding the shortest path subject to the constraint that forward progress is made at each step. This can be achieved in several ways. As an example, consider the case where each node is associated with a GPS device, and in addition to knowing the topology of the network, each node also knows the GPS location of all nodes in the network. The hop-by-hop routing algorithm can then be modified so that the shortest path is determined such that the next hop of the path is closer to the destination. In all

the experiments we assumed that the node energy decreases whenever

- The node forwards a message.
- The node performs a local broadcast of its energy information.

The cost of performing a local broadcast is determined in the same fashion as determining the energy consumed by transmission over a link in the network.

## VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our algorithm through simulations. First we consider the case where each node knows the instantaneous energy level of all nodes in the network. Next we study the case where the nodes have information about the energy levels of only the nodes in its immediate neighborhood.

### A. Effect of $\lambda$ and $\sigma$

In this section, we study the effect of the parameters  $\lambda$  and  $\sigma$  on the performance of CMAX. We consider a network of 20 nodes located randomly in a  $10 \times 10$  region. The initial energy of each node is 30 units. We assume that all messages are of unit length, and are generated randomly between all source-destination pairs. The energy required for transmitting a message along an edge  $(i, j)$  is  $\max(0.001, 0.001 * d_{ij}^3)$ , where  $d_{ij}$  is the distance between nodes  $i$  and  $j$ . This is similar to the kind of networks studied in [4]. We assume that each node can directly send a message to every other node, and so the underlying graph is complete. We also assume that the instantaneous energy level of every node is known to all nodes (the scenario where the energy levels of nodes are not known precisely is addressed later).

Based on Theorem 1, the following (loose) upper bounds on  $\lambda$  and  $\sigma$  can be easily computed:  $\sigma \leq 70$ ,  $\lambda \leq 140,000$ . Setting  $\sigma < \infty$  implies that messages may be rejected even if there is sufficient energy available to route the message. As mentioned earlier, this is needed to obtain the competitive ratio bound. We generate 100,000 messages in the network and Figure 1 shows the number of messages out of this 100,000 that are successfully routed as  $\lambda$  is varied, for two cases:

- $\sigma = 70$ .
- $\sigma = \infty$

The case of  $\sigma = \infty$  represents the case where there is no admission control, which is likely to be the case in practice. (It is unlikely that messages will be dropped if sufficient energy is available.) The figure shows that with  $\sigma = 70$ , the number of accepted messages increase with increasing  $\lambda$ , reaches a peak somewhere between  $\lambda = 10,000$  and  $\lambda = 100,000$ , and then starts decreasing with further increase in  $\lambda$ . With  $\sigma = \infty$ , the network capacity is relatively insensitive to  $\lambda$  as long as  $\lambda$  is large enough. Since this is probably the case that will be used in practice, the rest of the plots are shown for  $\sigma = \infty$ , and a sufficiently large  $\lambda$ . Therefore, messages are rejected only due to insufficient energy.

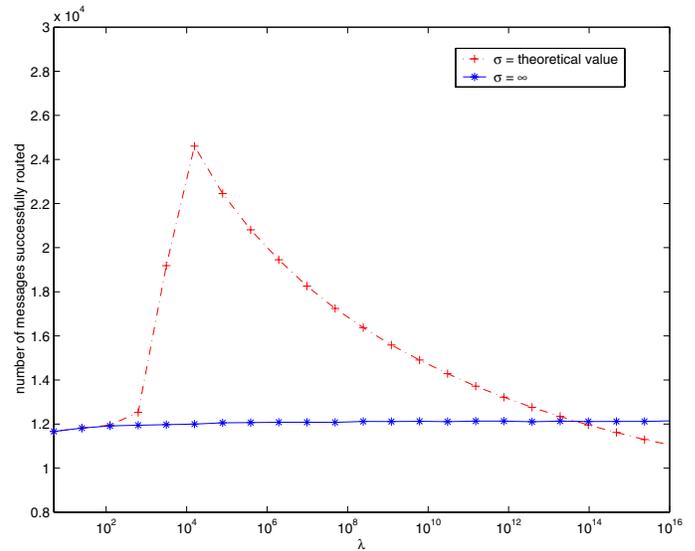


Fig. 1. Number of accepted messages (out of 100000 message arrivals) as a function of  $\lambda$ .

### B. Comparison with existing algorithms

Now we compare our algorithm with some existing algorithms in terms of the achieved network capacity and the achieved network lifetime. We compare the performance of the following algorithms:

- 1) Algorithm pmin, where a message is routed on a path that consumes minimum total energy,
- 2) Algorithm max-min, where a message is routed along a path such that minimum residual energy fraction of nodes on that path is the maximum amongst all paths,
- 3) max-min  $zP_{\min}$ , an algorithm proposed in [4], where a message is routed on the path with the maximum minimum residual energy fraction amongst all paths that consume at most  $z$  times the energy consumed in routing over the minimum energy path.

For the last algorithm, we experimented with different values of  $z$  and observed that the capacity is maximized when  $z = 2$  (the authors in [4] also report similar observations). This is the value we use in the rest of the paper.

First we compare the performance of the algorithms over 20 trials with respect to

- Maximum capacity of the network, i.e., the total number of messages successfully routed till no more messages can be routed. (Figure 2).
- Lifetime of the network, i.e., the number of accepted messages until the first rejection (Figure 3).

Note that in both these measures CMAX outperforms the other algorithms.

Figure 4 shows the number of rejects as the number of messages increases, for a single trial (the results for other trials were similar). Note that the performance of CMAX is better than all other algorithms throughout the trial. This figure demonstrates that our algorithm performs better than the other algorithms in terms of network capacity, measured till any point in time.

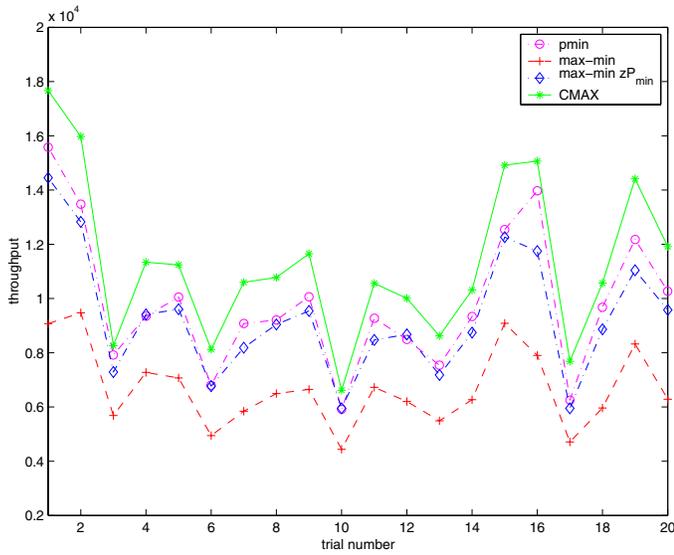


Fig. 2. Maximum network capacity under various algorithms.

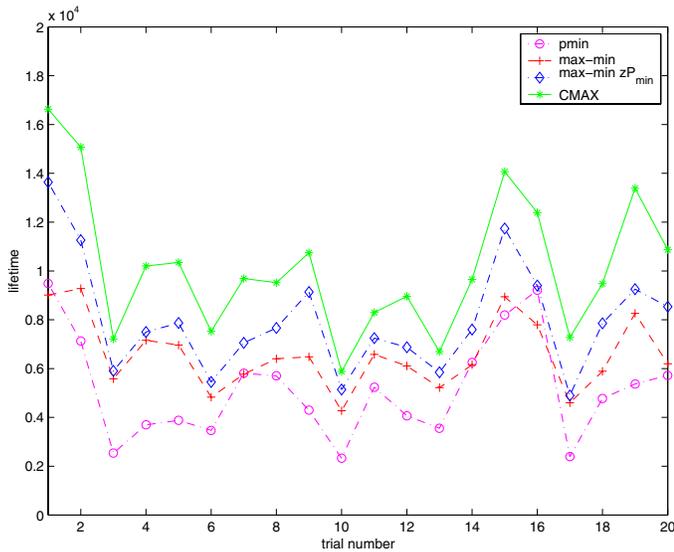


Fig. 3. Lifetime under various algorithms.

Next we compare the performance of CMAX to its distributed version, D-CMAX. Recall that D-CMAX works with imprecise information and local broadcasts. Recall that each local broadcast is characterized by  $(r, \gamma)$  where  $r$  is the local broadcast radius and  $\gamma$  determines the frequency of broadcasting. We assume that all arriving messages are of unit length, and the length the local broadcast message is 0.1 units. The local broadcast message contains the node id and current residual energy of the node. Thus if the local broadcast radius is  $r$ , then the cost of doing local broadcast is given by  $0.1 \times 0.001 \times r^3$ . The first term (0.1) is the length of the local broadcast message. The rest of the formula represents the energy required to send a message over a distance  $r$ .

### C. Effect of broadcast distance

We first study the effect of varying the broadcast distance on the performance of D-CMAX. Figure 5 shows how the

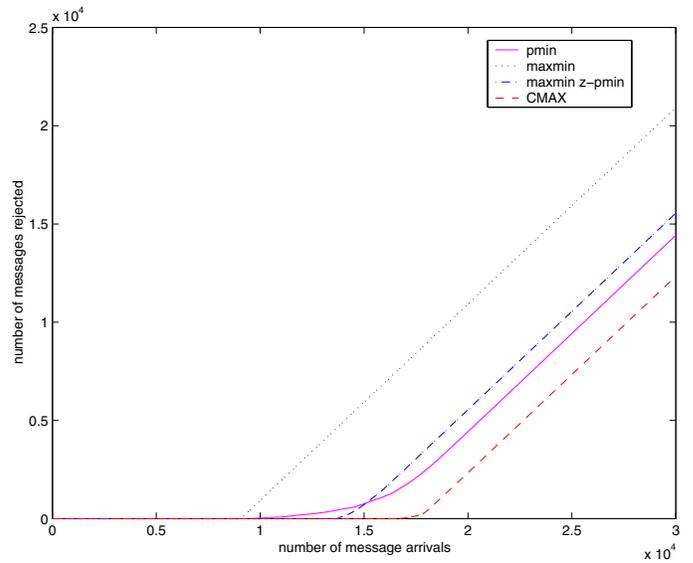


Fig. 4. Number of message rejects as a function of the number of message arrivals.

network capacity varies as a function of the broadcast radius  $r$ . The experiment is conducted for 50 nodes located randomly in  $25 \times 25$  region. The energy model is the same as in Section VI-A. Figure 5 shows the the number of accepted messages (out of 20000 message arrivals) as the broadcast distance of all nodes is varied from 0.2 units to 10 units. In these experiments,  $\lambda = 100$ , and  $\gamma = 1.0$  (i.e., a node broadcasts its energy information whenever its residual energy changes). All nodes have an initial energy of 32 units. The plot shows that the performance improves initially with increase in the broadcast distance. Increasing the broadcast distance allows more nodes to know about the correct energy level of a node, and therefore the algorithm usually takes better routing decisions. However, each broadcast consumes some energy at the broadcasting node which could have been used for transmitting messages arriving in future. Broadcasting over a large distance consumes a lot of energy and could drain the battery at the transmitter considerably, thus adversely affecting performance. Thus, in Figure 5, we see that when the broadcast distance becomes higher than a certain threshold, the achieved capacity diminishes.

### D. Effect of broadcast frequency

Figure 6 studies the effect of broadcast frequency on achieved capacity. The network and the simulation conditions are similar to those described in the previous subsection. The broadcast frequency parameter  $\gamma$  determines when a node broadcasts its energy level information. For example, if  $\gamma = 0.8$ , then a node broadcasts its energy level information only when its energy level drops to 0.8 times or lower compared to the last time it sent out a broadcast message. Increasing  $\gamma$  from 0 to 1 clearly increases the number of broadcast messages. As the figure shows, as the broadcast frequency increases, the achieved capacity (measured by the number of messages successfully routed over a sequence of 20000 message arrivals) increases initially, and then drops very sharply as  $\gamma$  approaches

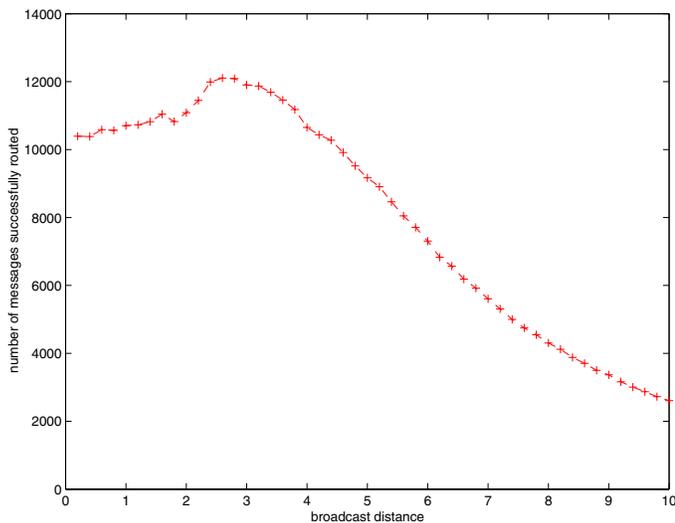


Fig. 5. Number of messages successfully routed as a function of the broadcast distance.

1. In the plots, the broadcast radius,  $r$ , is set to 5 units. For low values of  $\gamma$ , the broadcasts happen very infrequently, and so the nodes have to do the routing based on out-of-date information. Therefore, the routing decisions become better with increasing  $\gamma$ . However, as  $\gamma$  approaches 1, the number of broadcast messages become very high, which consume a lot of energy, thus causing degradation in performance.

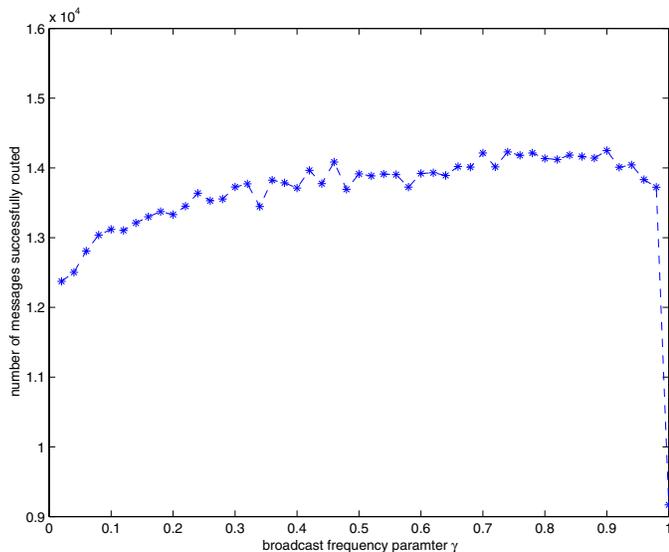


Fig. 6. Number of messages successfully routed as a function of  $\gamma$ , the broadcast frequency parameter.

### E. Comparison of CMAX and D-CMAX

In this section we show that by an appropriate choice of the broadcast distance and broadcast frequency, D-CMAX, the distributed version of the algorithm CMAX, can achieve performance close to CMAX. Figure 7 shows the number of messages routed successfully by the two algorithms D-CMAX and CMAX, for 20 different trials. Each different trial

corresponds to a different network and a different sequence of message arrivals. The networks are randomly generated with parameters similar to those described in previous subsections. The parameter  $\lambda$  is set to 100, the broadcast distance is set to 5 units, and the broadcast frequency parameter ( $\gamma$ ) is set to 0.5. From the plot we see that as expected, CMAX in general performs better than D-CMAX. However, on an average, D-CMAX performs fairly closely. In fact, averaging over the 20 different trials shows that D-CMAX does only about 8% worse than CMAX. We also performed experiments on large networks with 1000 nodes in a  $10 \times 10$  square. In all our experiments, we observed that the performance of D-CMAX goes down by only 8% to 15% compared to CMAX. The performance of the algorithm appears to scale well as the number of nodes in the network increases.

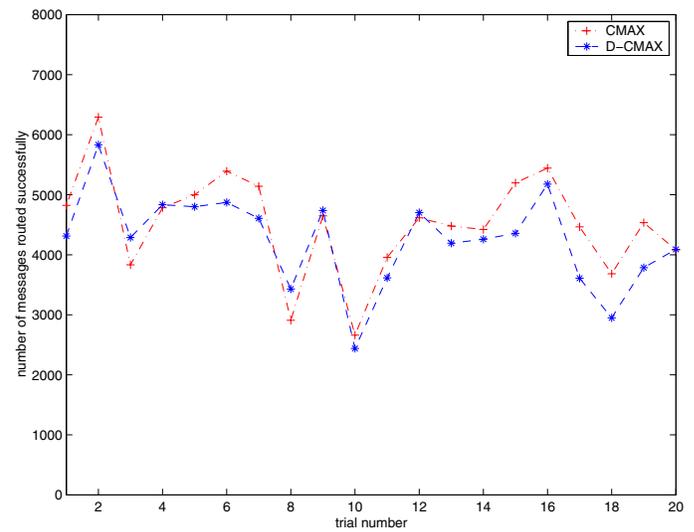


Fig. 7. Number of messages successfully routed by CMAX and D-CMAX, for 20 different trials.

## VII. CONCLUDING REMARKS

We presented a new algorithm for energy-aware on-line routing of messages in wireless ad-hoc networks. The algorithm uses only a single shortest path computation, and can be implemented in a distributed manner. Without making any assumptions regarding future message arrivals, we proved a logarithmic competitive ratio for the case where the routing objective is network capacity maximization. Obtaining this competitive ratio result needs the network to allow admission control. However, we showed by simulations that admission control is not needed to ensure good performance in practice. We also showed by simulations that the algorithm performs significantly better than previously proposed algorithms for both capacity maximization and lifetime maximization routing objectives. Furthermore, the algorithm is robust to imprecision in energy information and works well in a limited information flooding scenario.

### APPENDIX I: PROOF OF THEOREM 1

The competitive analysis is along the same lines as the competitive analysis presented in [5].

Associate a *cost*  $c_i$  with each node  $i \in N$ . The cost  $c_i$  before the arrival of the message  $k$  defined as

$$c_i(k) = E_i(\lambda^{\alpha_i(k)} - 1) \quad (2)$$

In the following, let  $S(k)$  be the set of messages successfully routed by our algorithm till the arrival of message  $k$ .

The proof is done in three steps. First we lower-bound the total length of messages routed by CMAX in terms of the node costs defined above.

*Lemma 2:*

$$\sum_{i \in N} c_i(k) \leq 2ne_{\max}L(k) \log \lambda.$$

**Proof:**

Consider any  $k' \in S(k)$ . Then, by (2), for any  $i \in N$ ,

$$\begin{aligned} & c_i(k' + 1) - c_i(k') \\ & \leq E_i \left( \lambda^{\alpha_i(k'+1)} - \lambda^{\alpha_i(k')} \right) \\ & = E_i \lambda^{\alpha_i(k')} \left( \lambda^{((E_i(k') - E_i(k'+1))/E_i)} - 1 \right) \\ & = E_i \lambda^{\alpha_i(k')} \left( \lambda^{(l_{k'} e_{ij}/E_i)} - 1 \right) \\ & = E_i \lambda^{\alpha_i(k')} \left( 2^{(l_{k'} e_{ij} \log(\lambda)/E_i)} - 1 \right) \end{aligned} \quad (3)$$

Note the base of the logarithm in (3) is 2. From assumption in (1),  $(l_{k'} e_{ij} \log \lambda / E_i) \leq 1$ . Also note that  $2^x - 1 \leq x$  for  $0 \leq x \leq 1$ . Therefore, from (3) we have

$$c_i(k' + 1) - c_i(k') \leq \lambda^{\alpha_i(k')} l_{k'} e_{ij} \log(\lambda) \quad (4)$$

Let  $P(k')$  be the path over which the message  $k'$  was routed. Note that since message  $k'$  was accepted by CMAX therefore,

$$\sum_{(i,j) \in P(k')} e_{ij} (\lambda^{\alpha_i(k')} - 1) \leq \sigma = ne_{\max} \quad (5)$$

Then summing the terms in (4) over all the nodes  $i \in N$ , we get

$$\begin{aligned} & \sum_{i \in N} (c_i(k' + 1) - c_i(k')) \\ & = \sum_{(i,j) \in P(k')} (c_i(k' + 1) - c_i(k')) \\ & \leq \sum_{(i,j) \in P(k')} \lambda^{\alpha_i(k')} l_{k'} e_{ij} \log \lambda \\ & = l_{k'} \log \lambda \sum_{(i,j) \in P(k')} e_{ij} (\lambda^{\alpha_i(k')} - 1) \\ & \quad + l_{k'} \log \lambda \sum_{(i,j) \in P(k')} e_{ij} \\ & \leq l_{k'} ne_{\max} \log \lambda + l_{k'} ne_{\max} \log \lambda \end{aligned} \quad (6)$$

$$\leq 2l_{k'} ne_{\max} \log \lambda \quad (7)$$

Relation (6) follows from (5) and the fact  $|P(k')| < n$ . Note that if  $k' \notin S(k)$ ,  $c_i(k' + 1) - c_i(k') = 0$ . Also note that

$c_i(1) = 0$ . Using these facts, and (7), we obtain

$$\begin{aligned} \sum_{i \in N} c_i(k) & = \sum_{k'=1}^{k-1} \sum_{i \in N} (c_i(k'+1) - c_i(k')) \\ & = \sum_{k' \in S(k)} (c_i(k'+1) - c_i(k')) \\ & \leq \sum_{k' \in S(k)} 2l_{k'} ne_{\max} \log \lambda \\ & = 2ne_{\max}L(k) \log \lambda \end{aligned}$$

■

In the next lemma we show that the sum of the lengths of the shortest paths of the rejected messages is lower bounded by  $ne_{\max}$ . This is needed to prove Lemma 4. Let  $T(k)$  be the set of messages successfully routed by the optimal off-line algorithm but rejected by CMAX, until the arrival of message  $k$ .

*Lemma 3:* For all  $k' \in T(k)$ ,

$$\sum_{(i,j) \in P(k')} e_{ij} (\lambda^{\alpha_i(k')} - 1) > ne_{\max}$$

**Proof:**

Note that a message can be rejected by our algorithm because of two reasons:

- 1) If there is no path with sufficient energy to route the message (see Step 1 of our algorithm).
- 2) If the shortest path length is too high (Step 4 of our algorithm).

Clearly, the above lemma holds if message  $k'$  is rejected in Step 4 of CMAX due to the choice of  $\sigma$ . Now let us assume that message  $k'$  is rejected in Step 1 of CMAX. Let  $P_{opt}(k')$  be the path over which the optimal algorithm routes message  $k'$ . Since the message is rejected by our algorithm, there exists an arc  $(i', j') \in P_{opt}(k')$  such that  $E_{i'}(k') < l_{k'} e_{i'j'}$ . Therefore,  $\alpha_i(k') > 1 - (l_{k'} e_{ij} / E_i) \geq 1 - (1 / \log \lambda)$  (from (1)), and therefore,

$$\begin{aligned} & \sum_{(i,j) \in P_{opt}(k')} e_{ij} (\lambda^{\alpha_i(k')} - 1) \\ & \geq e_{i'j'} (\lambda^{\alpha_{i'}(k')} - 1) \\ & > e_{i'j'} (\lambda^{(1 - (1 / \log \lambda))} - 1) \\ & = e_{i'j'} \left( \frac{\lambda}{2} - 1 \right) \\ & = e_{\min} \left( \frac{\lambda}{2} - 1 \right) \\ & = ne_{\max} \end{aligned} \quad (8)$$

Relation (8) follows from the choice of  $\lambda$  stated in Theorem 1.

We now bound the difference between the sum of the costs of the on-line and optimum off-line algorithms.

*Lemma 4:*

$$ne_{\max}(L_{opt}(k) - L(k)) \leq \sum_{i \in N} c_i(k)$$

**Proof:**

Using Lemma 3,

$$\begin{aligned}
& ne_{\max}(L_{opt}(k) - L(k)) \\
& \leq \sum_{k' \in T(k)} ne_{\max} l_{k'} \\
& < \sum_{k' \in T(k)} \sum_{(i,j) \in P(k')} l_{k'} e_{ij} (\lambda^{\alpha_i(k')} - 1) \\
& = \sum_{k' \in T(k)} \sum_{(i,j) \in P(k')} l_{k'} e_{ij} c_i(k') / E_i \quad (9)
\end{aligned}$$

$$= \sum_{k' \in T(k)} \sum_{(i,j) \in P(k')} l_{k'} e_{ij} c_i(k) / E_i \quad (10)$$

$$\begin{aligned}
& = \sum_{i \in N} c_i(k) \sum_{k' \in T(k), (i,j) \in P(k')} (l_{k'} e_{ij} / E_i) \\
& \leq \sum_{i \in N} c_i(k) \quad (11)
\end{aligned}$$

Relation (9) follows from (2), relation (10) holds because the node costs are non-decreasing, and (11) follows from the fact that the total energy expended (in transmitting messages) at a node cannot exceed the initial energy of the node. ■

From Lemmas 2 and 4, it follows that

$$L(k)/L_{opt}(k) \geq \frac{1}{1 + 2 \log \lambda} ,$$

thus proving Theorem 1.

## REFERENCES

- [1] J-H. Chang, L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks", *Proc. of Infocom 2000*, March 2000.
- [2] J-H. Chang, L. Tassiulas, "Fast Approximate Algorithms for Maximum Lifetime Routing in Wireless Ad-hoc Networks", *Proc. of Networking 2000*, May 2000.
- [3] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad-hoc wireless networks", *Proc. of Mobicom 2001*, July 2001.
- [4] Q. Li, J. Aslam, D. Rus, "On-line power-aware routing in wireless ad-hoc networks", *Proc. of Mobicom 2001*, July 2001.
- [5] S. Plotkin, "Competitive routing of virtual Circuits in ATM networks", *IEEE Journal on Selected Areas in Communications*, 13(6):1128-1136, 1995.
- [6] R. Ramanathan, R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment", *Proc. of Infocom 2000*, March 2000.
- [7] V. Rodoplu, T. H. Meng, "Minimum energy mobile wireless networks", *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, August 1999.
- [8] D. D. Sleator, R. E. Tarjan, "Amortized efficiency of list update and paging rules", *Communications of the ACM*, 28(2):202-208, 1985.
- [9] R. Wattenhofer, L. Li, P. Bahl, Y-M. Wang, "Distributed Topology Control for Wireless Multihop Ad-hoc Networks", *Proc. of Infocom 2001*, April 2001.
- [10] S. Singh, M. Wu, C. S. Raghavendra, "Power-aware routing in mobile ad-hoc networks", *Proc. of Mobicom 1998*, October 1998.