Maximum Lifetime Routing In Wireless Sensor Networks *

Joongseok Park Sartaj Sahni
Computer & Information Science & Engineering
University of Florida
{jpark, sahni}@cise.ufl.edu

June 2, 2005

Abstract

We show that the problem of routing messages in a wireless sensor network so as to maximize network lifetime is NP-hard. In our model, the online model, each message has to be routed without knowledge of future route requests. We develop also an online heuristic to maximize network lifetime. Our heuristic, which performs two shortest path computations to route each message, is superior to previously published heuristics for lifetime maximization– our heuristic results in greater lifetime and its performance is less sensitive to the selection of heuristic parameters. Additionally, our heuristic is superior on the capacity metric.

Keywords: Wireless sensor networks, lifetime maximization, online power-aware routing

1 Introduction

We consider message routing in wireless sensor networks in which each sensor is battery operated. We assume that the batteries are neither replaceable nor rechargeable. This, for example, is the case when the sensors are deployed in hostile (e.g., battlefield) or otherwise hard to reach (e.g., the bottom of the ocean) environments. Hence, in the wireless sensor networks we consider, energy conservation is paramount.

The energy required by a sensor to transmit, under ideal conditions, a unit length message a distance r is proportional to r^d for some d in the range $[2, 4]$ [3, 12]. (More energy is required if there are obstacles in the transmission path.) Hence, energy is conserved by using multihop routing. That is, nodes between the source and destination are used as relays. For example,

*This research was supported, in part, by the National Science Foundation under grant ITR-0326155

suppose that sensors A and C are a distance 4 apart and sensor B is halfway between A and C. Assume further that $d = 2$ and the constant of proportionality is 1. It takes A 16 units of energy to transmit a unit message directly (i.e., via a single hop) to C. However, by using B as a relay, the message may be transmitted from A to B using 4 units of energy and then from B to C using an additional 4 units. The total energy consumed by this multihop (more precisely, two hop) transmission is only 8 units[1]!

Following the deployment of a sensor network, pairs of sensors exchange messages in an unpredictable sequence. We assume that for proper network operation, it is critical that every attempt to transmit a message succeed. Hence, the *lifetime* of a sensor network is defined as the number of messages successfully routed before the first failed message route.

Although several researchers have proposed heuristics to maximize network lifetime, none has actually shown that maximizing network lifetime is NP-hard. In Section 4, we show that maximizing network lifetime is NP-hard. In Section 5, we develop a new heuristic, the online maximum lifetime (OML) heuristic, to maximize lifetime. Zone-based and distributed versions of OML are described in Section 6. In Section 7, the performance of our OML heuristic is compared with previously proposed heuristics for lifetime maximization. Our experiments show that our OML heuristic is superior to the previously proposed heuristics. We begin, in Section 2, by introducing our terminology. Section 3 describes related work.

2 Terminology

The sensor network is modeled as a directed graph $G = (V, E)$. V is the set of sensors in the network and $n = |V|$ is the number of sensors. E is the edge set. There is a directed edge $(u, v) \in E$ from sensor u to sensor v iff a single-hop transmission from u to v is possible. Let $ie(i) > 0$ be the initial energy in sensor i, $ce(i) \geq 0$ denotes the current energy in sensor i, and for each $(u, v) \in E$, $w(u, v) > 0$ denotes the energy required to do a single-hop transmission from sensor u to sensor v. Following a single-hop message tranmission from u to v, the current energy in sensor u becomes $ce(u) - w(u, v)$. Note that this single-hop transmission is possible only if $ce(u) \geq w(u, v)$. Since we assume no energy is consumed during message reception, the current

[1]In reality, slightly more than 8 units are used as both B and C have to expend some energy receiving the message. In this paper, as is done in other papers on routing, we ignore the energy needed to receive a message.

energy in sensor v is unaffected by a transmission from u to v.

A *routing request* is a pair (s, t), where s is the source sensor for the message that is to be routed and t is the destination sensor. Although, in this paper, we assume that all messages have the same length, our work is easily extended to the case when messages have different length. Let $R = r_1, r_2, \cdots$ be a sequence of routing requests. Note that each r_i is a source-destination pair (s_i, t_i). The *lifetime* of a network for request sequence R is the maximum j such that routing requests r_1, $\cdots$, r_j are successfully routed. An *online* routing algorithm routes r_i without knowledge of any r_j, $j > i$. An *offline* routing algorithm, on the other hand, determines the routes for each r_i with full knowledge of all succeeding routing requests. Clearly, the lifetime obtainable by the best offline algorithm is at least as much as that obtainable by any online algorithm. The *competitive ratio* of an online routing algorithm is the maximum value of the lifetime obtained using the online algorithm divided by the lifetime obtained by the best offline algorithm; the maximum is taken over all routing sequences R.

3 Related Work

Several authors have developed energy-efficient algorithms for point-to-point communication [4, 5, 6, 10, 11, 13, 17, 18, 19, 20, 22]. The overall objective of these algorithms is to either maximize the lifetime (time at which a communication fails first) or the capacity of the network (amount of data traffic carried by the network over some fixed period of time).

Aslam, Li and Rus [3] have shown that there is no online routing algorithm with $o(n)$ competitve ratio for the lifetime maximization problem. Singh, Woo and Raghavendra [17] propose five metrics that may be used in the selection of the routing path for energy efficient routing. The first of these is to use a minimum-energy path (i.e., a path in G for which the sum of the edge weights is minimum) from s to t. Such a path may be computed using Dijkstra's shortest path algorithm [16]. Although this metric tends to minimize the total (or averge) energy consumed over a sequence of routes, it doesn't focus on the primary objective of maximizing lifetime. This is because using a minimum-energy path for the current route request may prevent the successful routing of future messages.

The remaining four metrics proposed in [17] are maximize time to network partition, minimize

variance in node energy levels, minimize the node cost of each transmission (the cost of a node is some function of the amount of energy used so far by that node), and minimize maximum node cost. Of the proposed five metrics, only the first (minimum-energy path) and fourth (minimize node cost) have been implemented by Singh, Woo and Raghavendra [17]. They raise concerns about the diffculty of implementing the remaining three in a routing protocol.

Toh et al. [20] propose the MMBCR and CMMBCR (conditional MMBCR) online algorithms to select a source-to-destination path. The MMBCR algorithm selects a path P for which the minimum of the residual energies (i.e., energy remaining following a route) of the sensors on P is maximum. Recognizing that to maximize lifetime we need to achieve some balance between the energy consumed by a route and the minimum residual energy at the nodes along the chosen route, Toh et al. [20] propose also a conditional MMBCR algorithm, CMMBCR. In CMMBCR we look for a minimum energy source-to-destination path in which no sensor has residual energy below a threshold γ. If there is no source-to-destination path with this property, then the MMBCR path is used.

In the MRPC lifetime-maximization heuristic of Misra and Banerjee [15], the capacity, $c(u, v)$ of edge (u, v) is defined to be $ce(u)/w(u, v)$. Note that $c(u, v)$ is the number of unit-length messages that may be transmitted along (u, v) before node u runs out of energy. The lifetime of path P, $life(P)$ is defined to be the the minimum edge capacity on the path. In MRPC, routing is done along a path P with maximum lifetime. Figure 1 gives the MRPC algorithm. A decentralized implementation as well as a conditional MRPC, CMRPC, also are described in [15]. CMRCP attempts to route on a minimum-energy path P with $life(P) \geq \gamma$, where γ is a specified threshold value. If there is no source-to-destination path P with $life(P) \geq \gamma$, the MRPC path is used.

Aslam, Li and Rus [3] propose the max-min zP_{min}-path algorithm to select routes that attempt to make this balance. This algorithm selects a path that uses at most $z * P_{min}$ energy, where z is a parameter to the algorithm and P_{min} is the energy required by the minimum-energy path. The selected path maximizes the minimum residual energy fraction (energy remaining after route/initial energy) for nodes on the route path. Notice that the possible values for the residual energy fraction of node u may be obtained by computing $(ce(u) - w(u, v))/ie(u)$, where $ce(u)$ is

Step 1: [Initialize]
Eliminate from G every edge (u, v) for which $ce(u) < w(u, v)$.
For every remaining edge (u, v) let $c(u, v) = ce(u)/w(u, v)$.
Let L be the list of distinct $c(u, v)$ values.

Step 2: [Binary Search]
Do a binary search in L to find the maximum value max for which there is a path P from source to destination that uses no edge with $c(u, v) < max$.
For this, when testing a value q from L, we perform a depth- or breadth-first search beginning at the source. The search is not permitted to use edges with $c(u, v) < q$.
Let P be the source-to-destination path with lifetime max.

Step 3: [Wrap Up]
If no path is found in Step 2, the route isn't possible.
Otherwise, use P for the route.

Figure 1: MRPC algorithm of [15]

the (current) energy at node u just before the route. This computation is done for all vertices v adjacent from u. Hence the union, L, of these values taken over all u gives the possible values for the minimum residual-energy fraction along any path.

Figure 2 gives the max-min zP_{min} algorithm.

Several adaptations to the basic max-min zP_{min} algorithm, including a distributed version are described in [3]. Kar et al. [12] develop an online capacity-competitive (the capacity is the number of messages routed over some time period) algorithm, CMAX, with logarithmic competitive ratio. On the surface, this would appear to violate the $\Omega(n)$ bound of [3]. However, to achieve this logarithmic competitive ratio, the algorithm CMAX does admission control. That is, it rejects some routes that are possible. The bound of [3] applies only for online algorithms that perform every route that is possible.

Let $\alpha(u) = 1 - ce(u)/ie(u)$ be the fraction of u's initial energy that has been used so far. Let λ and σ be two constants. In the CMAX algorithm, the weight of every edge (u, v) is changed from $w(u, v)$ to $w(u, v) * (\lambda^{\alpha(u)} - 1)$. The shortest source-to-destination path P in the resulting graph is determined. If the length of this path is more than σ, the routing request is rejected (admission control); otherwise, the route is done using path P. Figure 3 gives the algorithm.

Step 1: [Initialize]
Eliminate from G every edge (u, v) for which $ce(u) < w(u, v)$.
Let L be the list of possible values for the minimum residual-energy fraction.

Step 2: [Binary Search]
Do a binary search in L to find the maximum value max of the minimum residual-energy fraction for which there is a path P from source to destination that uses at most $z * P_{min}$ energy.
For this, when testing a value q from L, we find a shortest source to destination path that does not use edges (u, v) that make the residual-energy fraction at u less than q.

Step 3: [Wrap Up]
If no path is found in Step 2, the route isn't possible.
Otherwise, use the path P corresponding to max.

Figure 2: The max-min zP_{min} lifetime algorithm of [3]

Step 1: [Initialize]
Eliminate from G every edge (u, v) for which $ce(u) < w(u, v)$.
Change the weight of every remaining edge (u, v) to $w(u, v) * (\lambda^{\alpha(u)} - 1)$.

Step 2: [Shortest Path]
Let P be the shortest source-to-destination path in the modified graph.

Step 3: [Wrap Up]
If no path is found in Step 2, the route isn't possible.
If the the length of P is more than σ do not do the route.
Otherwise, use P for the route.

Figure 3: CMAX algorithm of [12]

The CMAX algorithm of Figure 3 has a complexity advantage over the max-min zP_{min} algorithm of Figure 2. The former does only 1 shortest-path computation per routing request while the latter does $O(\log n)$, where n is the number of sensor nodes. Although admission control is necessary to establish the logarithmic competitive-ratio bound for CMAX, we may use CMAX without admission control (i.e., route every request that is feasible) by setting $\sigma = \infty$. Experimental results reported in [12] suggest that CMAX with no admission control outperforms max-min

zP_{min} on both the lifetime and capacity metrics.

For the performance evaluation of CMAX, Kar et al. [12] introduce another route selection algorithm, which they can max min. We refer to this algorithm as maxRE here. In maxRE, the routing path is selected so as to maximize the minimum residual energy fraction of sensors on the path. This algorithm is equivalent to max-min zP_{min} with $z = \infty$.

Chang and Tassiulas [4, 5] develop a linear-programming formulation for lifetime maximization. This formulation requires knowledge of the rate at which each node generates messages. Wu, Gao and Stojmenovic [21] propose routing based on connected dominating sets to maximize network lifetime. Stojmenovic and Lin [19] and Melodia et al. [14] develop localized algorithms to maximize lifetime and Heinzelman, Chandrakasan and Balakrishnan [9] develop a clustering-based routing algorithm (LEACH) for sensor networks.

4 NP-hardness of Maximum Lifetime Problem

In this section we show that the Maximum Lifetime Problem (ML) is NP-hard. Our proof employs the disjoint connecting paths (DCP) problem, which is known to be NP-complete [8]. The input to the DCP problem is a graph G (either directed or undirected) and a set of k disjoint source and destination vertex pairs (s_i, t_i), $1 \leq i \leq k$. The output is "yes" iff G has k vertex-disjoint paths; the ith path connects s_i and t_i, $1 \leq i \leq k$.

Theorem 1 *The maximum lifetime problem ML is NP-hard.*

Proof We show how to construct, for any given instance of the DCP problem, an instance of ML whose lifetime is k iff the answer to the given DCP instance is "yes". Since this construction takes polynomial time, a polynomial time algorithm for ML would imply a polynomial time algorithm for DCP. Hence, ML is NP-hard.

Let G, (s_i, t_i), $1 \leq i \leq k$ define the DCP instance. The network for the corresponding ML instance is obtained by introducing k new vertices t'_i and k new edges (t_i, t'_i) to G. $ie(u) = 1$ for each vertex of G, $ie(t'_i) = 0$, and $w(u, v) = 1$ for every edge in the constructed sensor network. The request sequence R has $r_i = (s_i, t'_i)$, $1 \leq i \leq k$.

Suppose that the answer to the given DCP instance is "yes". So, the instance has k vertex-disjoint paths P_i with P_i connecting s_i and t_i, $1 \leq i \leq k$. We may route r_i along the path P'_i

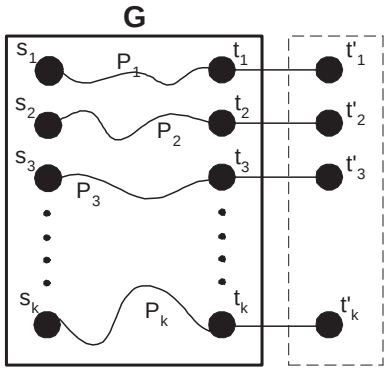


Figure 4: ML instance corresponding to DCP instance given by G and (s_i, t_i), $1 \leq i \leq k$

obtained by concatenating t'_i to P_i. The k paths P'_i, $1 \leq i \leq k$ are vertex disjoint and so use at most 1 unit of energy from each vertex of G. Since t'_i can only be a terminal vertex of a P'_i, no t'_i expends energy when these P'_i paths are used. Hence, the network lifetime is k.

Next, suppose that the lifetime is k. Let P'_i be the route used for r_i, $1 \leq i \leq k$. Let P_i be obtained from P'_i by removing the last edge of P'_i. Since $ie(t'_i) = 0$, no t'_i may be used as a relay on any path. Hence t'_i is on no P_j. Further, the first vertex of P_i is s_i and the last is t_i (as the only way to get to t'_i is to use the edge (t_i, t'_i). The paths P'_i, $1 \leq i \leq k$, and so also the paths P_i, $1 \leq i \leq k$, must be vertex disjoint as $ie(u) = 1$ for every vertex u of G and once one of these vertices is used in a P'_i, it is depleted of all energy. Hence, the P_is define vertex disjoint s_i to t_i paths in G and the answer to the DCP instance is "yes". ■

Note that the proof of Theorem 1 applies to both directed and undirected sensor networks as well as to the offline and online versions of the maximum lifetime problem.

5 The Online Maximum Lifetime Heuristic (OML)

To maximize lifetime, we need to delay as much as possible the depletion of a sensor's energy to a level below that needed to transmit to its closest neighbor. We attempt to accomplish this objective using a two-step algorithm to find a path for each routing request $r_i = (s_i, t_i)$. In the first step, we remove from G all edges (u, v) such that $ce(u) < w(u, v)$ as these edges require more energy than available for a transmit. Let the resulting graph be $G' = (V, E')$. Next, we determine the minimum energy path, P'_i, from s_i to t_i in the pruned graph G'. This may be done using

Dijkstra's shortest path algorithm [16]. In case there is no s_i to t_i path in the pruned graph G', the routing request r_i fails. So, assume such a P'_i exists. Using P'_i, we compute the residual energy, $re(u) = ce(u) - w(u,v)$ for (u,v) an edge on P'_i. Let $minRE = \min\{re(u) | u \in P'_i \&\& u \neq t_i\}$. Let $G'' = (V, E'')$ be obtained from G' by removing all edges $(u,v) \in E'$ with $ce(u) - w(u,v) < minRE$. That is, all edges whose use would result in a residual energy below $minRE$ are pruned from E'. This pruning is an attempt to prevent the depletion of energy from sensors that are low on energy.

In the second step, we find the path to be used to route the request r_i. For this, we begin with G'' as above and assign weights to each $(u,v) \in E''$. The weight assignment is done so as to balance the desire to minimize total energy consumption as well as the desire to prevent the depletion of a sensor's energy. Let $eMin(u) = \min\{w(u,v) | (u,v) \in E''\}$ be the energy needed by sensor u to transmit a message to its nearest neighbor in G''. Let $\rho(u,v)$ be defined as below.

$$\rho(u,v) = \begin{cases} 0 & \text{if } ce(u) - w(u,v) > eMin(u) \\ c & \text{otherwise} \end{cases}$$

where c is a non-negative constant and is an algorithm parameter. For each $u \in V$, define

$$\alpha(u) = \frac{minRE}{ce(u)}$$

The weight $w''(u,v)$ assigned to edge $(u,v) \in E''$ is

$$w''(u,v) = (w(u,v) + \rho(u,v))(\lambda^{\alpha(u)} - 1)$$

where λ is another non-negative constant and an algorithm parameter. As can be seen, this weighting function, through ρ, assigns a high weight to edges whose use on a routing path causes a sensor's residual energy to become low. Also, all edges emanating from a sensor whose current energy is small relative to $minRE$ are assigned a high weight because of the λ term. Thus the weighting function discourages the use of edges whose use on a routing path is likely to result in the failure of a future route.

Figure 5 gives our OML algorithm to select a path for request r_i. This algorithm may be used as a heuristic for lifetime maximization. The resulting heuristic is the OML heuristic. Note that our OML algorithm performs two shortest-path computations for each route request. By contrast, the CMAX algorithm of [12] does only one shortest-path computation per route request,

Step 1: [Compute G'']
$G' = (V, E')$ where $E' = E - \{(u, v) | ce(u) < w(u, v)\}$.
Let P'_i be a shortest s_i to t_i path in G'.
If there is no such P'_i, the route request fails, stop.
Compute the minimum residual energy $minRE$ for sensors other than t_i on P'_i.
Let $G'' = (V, E'')$ where $E'' = E' - \{(u, v) | ce(u) - w(u, v) < minRE\}$.

Step 2: [Find route path]
Compute the weight $w''(u, v)$ for each edge of E''.
Let P''_i be a shortest s_i to t_i path in G''.
Use P''_i to route from s_i to t_i.

Figure 5: Our OML algorithm for route request $r_i = (s_i, t_i)$

the max-min zP_{min} algorithm of [3] does $O(\log n)$ shortest-path computations per route request and the MRPC algorithm of [15] does $O(\log n)$ depth- or breadth-first searches per route request.

Note that although both the CMAX and OML algorithms have a $\lambda^{\alpha(u)} - 1$ term in the edge weighting function, the two algorithms use a different $\alpha(u)$ function. In the case of CMAX, $\alpha(u) = 1 - ce(u)/ie(u)$ is the fraction of u's initial energy that has been used so far. So, CMAX discourages the use, as relays, of sensors that have depleted a large fraction of their initial energy (even though such sensors may a large amount of energy remaining). In OML, $\alpha(u) = minRE/ce(u)$. Hence OML discourages the use, as relays, of sensors whose current energy is much less than $minRE$.

6 Distributed OML

OML, as presented in Section 5 is a centralized algorithm that requires a computational node with complete information regarding the topology of the sensor network and current energy levels of all sensors. In real sensor networks, these requirements often are impractical. We may develop a hierarchical zone-based version of OML using the same strategy as used in [3] to arrive at a zone-based max-min zP_min algorithm. Such a zone-based algorithm divides the sensors into zones, each zone containing a set of geographically close sensors. Each zone has a host sensor that does local routing (the host sensor in a zone may change with time and may, for example, be the sensor with maximum current energy). There is also a sensor or node responsible for global routing among sensor hosts. This global routing sensor treats each zone as a single network vertex

and uses an estimate of the current energy in the zone. Kar et al. [12] have pointed out the same zone-based strategy may be employed to derive a zone-based CMAX algorithm.

Aslam et al. [3] describe also a strategy to arrive at a distributed version of their min-max zP_{min} algorithm. The strategy emploed in [3] also may be employed to arrive at a distributed version of OML. Kar et al. [12] use a limited flooding approach to develop a distributed version of their CMAX algorithm. In this limited flooding approach, each node computes the shortest path (using edge weights as described for CMAX) to the destination and forwards the message to the next hop on this shortest path. Wtih some periodicity, each sensor broadcasts its current energy level to sensors within some distant r from it. Hence when computing a shortest path to the destination, a sensor has the global topology, a reasonably recent measure of the energy of near by sensors and the initial (or much less recent) energy of distant sensors. The distributed algorithm is augmented with loop avoidance defenses. Clearly, this limited flooding strategy may be employed to arrive at a distributed OML algorithm as well.

7 Evaluation

Simulations reported in the literature already have established the superiority of CMAX over max-min zP_{min}, ME and maxRE [12] as well as the superiority of MRPC and CPRCP over MMBCR, CMMBCR, and ME [15]. Misra and Banerjee [15] report that for a sensor transmission radius, r_T, (i.e., the maximum distance that a sensor can transmit in a single hop) of 2.9 units (a unit being the smallest permissible separation between 2 sensors) and larger, MRPC performs slightly better than does CMRPC whereas for a smaller transmission radius such as 1.9 units, CMRPC has a better performance. Therefore, the focus of our simulations is the relative performance of OML, CMAX and MRPC. Since our experiments use $r_T > 5$, we do not include CMRPC in our study.

Our experimental setup is similar to that employed by Kar et al. [12] and Aslam et al. [3]. We randomly populate either a 10×10 or a 25×25 grid with sensors. The sensors are deployed at randomly selected grid points. The energy required by a single-hop transmission between two sensors is $0.001 * d^3$, where d is the Euclidean distance between the two sensors.

7.1 Selecting OML and CMAX Parameters

To determine suitable values for λ and ρ, we experimented with 10×10 grids into which 20 sensors were randomly placed. A total of 10 random placements of 20 sensors were considered. In other words, we used 10 random sensor networks defined over a 10×10 grid. Each network had 20 sensors. The transmission radius and initial energy for each sensor were set to ∞ and 30, respectively. For our first experiment, we set $c = 0.001r_T^3$ in the definition of ρ. Network lifetime was determined for $\lambda = 2^i$, $1 \leq i \leq 12$. For each network and λ combination, the lifetime over 10 random route request sequences was measured. So, for each λ value, 100 lifetime measurements (10 networks with 10 sequences each) were made. The average lifetime for these 100 measurements is reported in Figure 6. Figure 7 gives the average lifetime for 9 of the 10 networks used in the study of Figure 6. However, this time, the average is over the 120 lifetimes for the 10 sequences times 12 λ values used for each network. The standard deviation in the lifetime over these 120 tests also is reported. The rows labeled Avg Imp give the average increase in lifetime obtained by OML over that obtained by CMAX and the rows labeled Std Dev Imp give the standard deviation in this lifetime increase. The results for the 10th network used in our study are similar to those reported in Figure 7. In all 1200 lifetime tests (120 per network; 10 networks), the OML algorithm resulted in a lifetime greater than or equal to that obtained by CMAX!

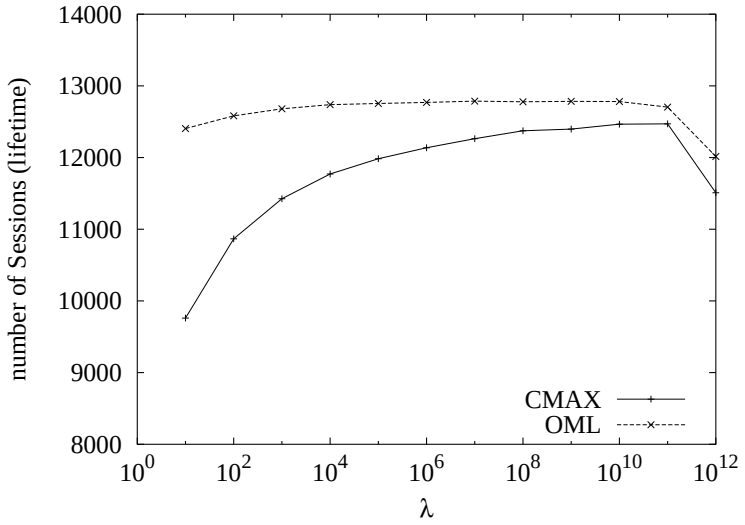


Figure 6: Average lifetime for different λ values

	Graph 1		Graph 2		Graph 3	
	CMAX	OML	CMAX	OML	CMAX	OML
Avg	10651.1	11540.7	10995.3	11750.3	9680.4	10635.3
Std Dev	890.7	148.9	631.7	269.2	857.3	162.9
Avg Imp (%)	8.3		6.8		9.8	
Std Dev Imp (%)	4.9		6.8		10.7	
	Graph 4		Graph 5		Graph 6	
	CMAX	OML	CMAX	OML	CMAX	OML
Avg	10952.2	11988.4	9258.0	10340.8	11729.6	13151.0
Std Dev	756.93	146.67	979.89	170.33	904.38	178.12
Avg Imp (%)	9.5		11.6		12.1	
Std Dev Imp (%)	8.0		13.4		8.3	
	Graph 7		Graph 8		Graph 9	
	CMAX	OML	CMAX	OML	CMAX	OML
Avg	14288.9	15457.5	12399.3	13728.2	11206.2	1200.1
Std Dev	954.59	246.65	1095.74	194.39	776.5	185.0
Avg Imp (%)	8.2		10.7		7.1	
Std Dev Imp (%)	6.1		11.7		6.6	

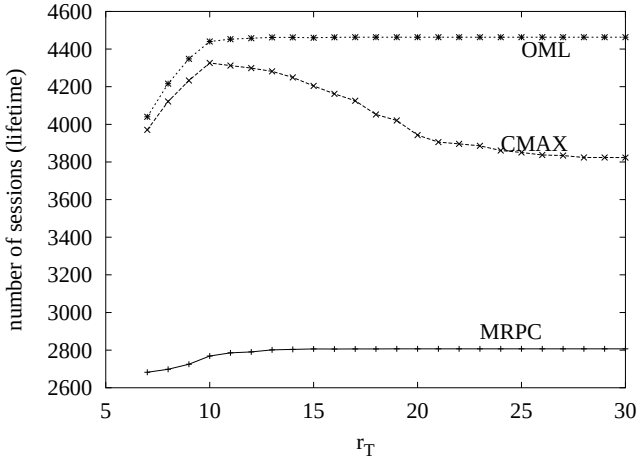
Figure 7: Lifetime statistics for OML and CMAX

As can be seen from Figure 6, the performance of CMAX is quite sensitive to the chosen value of λ whereas OML has a relatively stable performance for λ between 10^4 and 10^{11}. In further experiments, we used $\lambda = 10^{11}$ for both OML and CMAX.

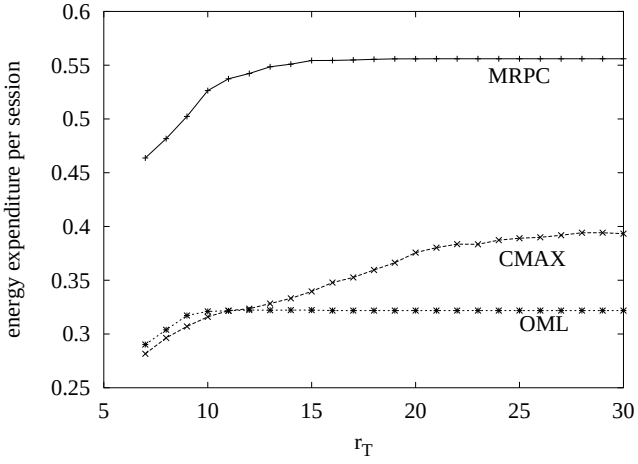
A similar experiment was conducted to determine a suitable value for c in the definition of ρ. This experiment revealed that, for our data sets, lifetime was reltively insensitive to the choice of c. However, lifetime was (marginally) best when $c = 0.001 * r_T^3$ (recall that our single-hop energy model is $0.001 * d^3$ units of energy are needed to transmit a distance d).

7.2 Effect of Transmission Radius

To determine the effect of transmission radius on the performance of OML, CMAX, and MRPC, we used a 25×25 grid. 10 networks, each obtained by placing 50 sensors at randomly chosen grid points were considered. For each network, 10 route request sequences were generated. The initial energy at each sensor was set to 30 and the transmission radius r_T was varied from 7 to 30. Figure 8 shows the average lifetime and average energy consumed per session (i.e., source-



(a) Average lifetime



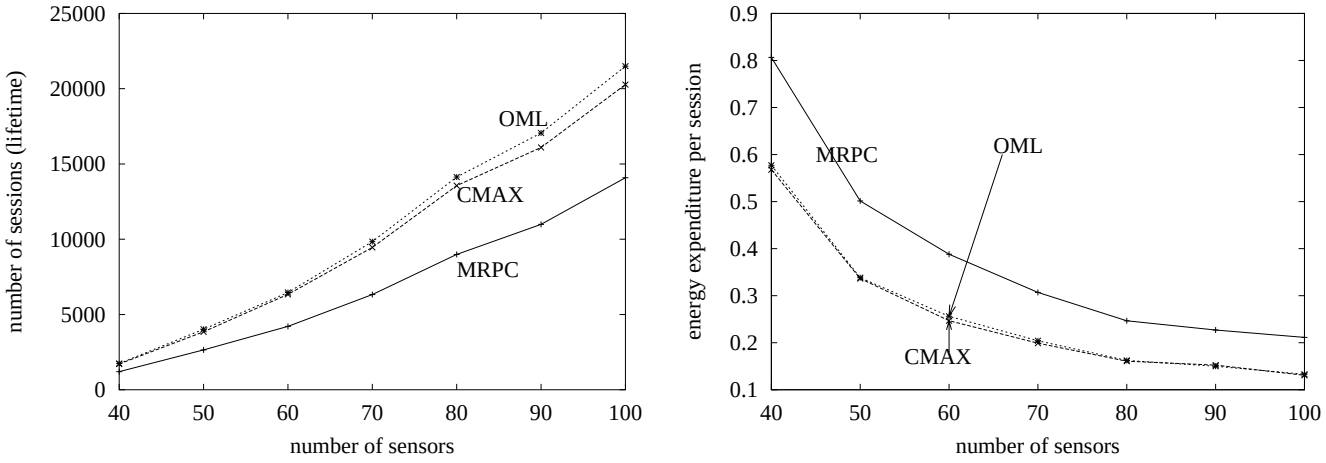
(b) Average energy per session

Figure 8: Average lifetime and energy consumption as a function of transmission radius

to-destination route). The lifetime obtained usinh either OML or CMAX is significantly larger than that obtained using MRPC. For small transmission radii, CMAX and OML have similar performance. As the transmission radius is increased from a low of 7, the lifetime obtained by both CMAX and OML increases. However, while the lifetime obtained by OML is fairly stable for $r_T \geq 10$, the lifetime obtained by CMAX declines rapidly as r_T increases beyond 10. Similarly, the energy consumed per session is much larger when MRPC is used than when either OML or CMAX is used. For $r_T < 10$, CMAX used slightly less energy per session than did OML. However, for larger values of r_T, CMAX was more energy frugal.

7.3 Effect of Sensor Density

Our next study compared the effect of sensor density on performance. For this study, we placed n sensors, $n \in \{40, 50, 60, 70, 80, 90, 100\}$, at random locations in a 25×25 grid. For each choice of n, we experimented with $r_T \in \{10, 20, 30\}$. The initial energy at each sensor was set to 30. For each n and r_T combination, we generated 10 random placements and for each placement, we generated 10 random request sequences. Figures 9 through 11 graph, as a function of n, the average lifetime and average energy consumed per session for each r_T. For all 700 test instances, the lifetime using OML was at least as large as that obtained using CMAX. MRPC consistently underperformed



(a) Average lifetime

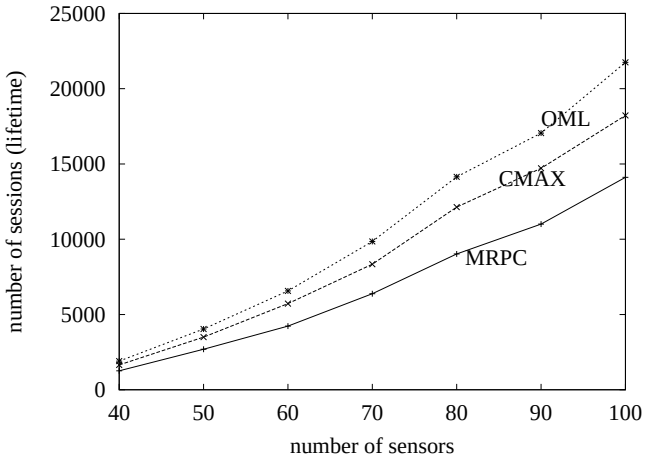
(b) Average energy per session

Figure 9: Lifetime and energy consumed as a function of n, $r_T = 10$

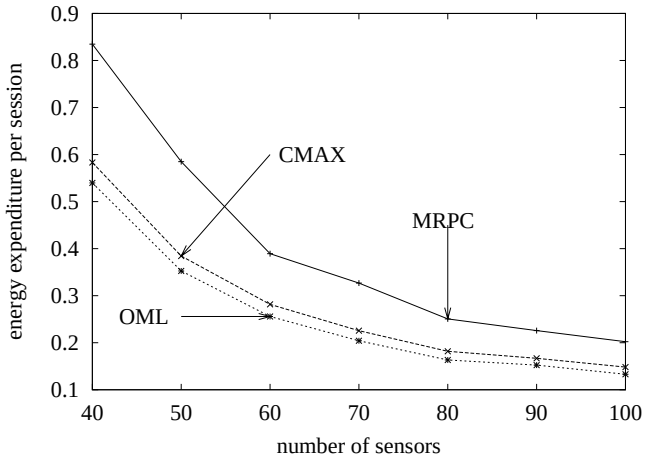
both OML and MRPC. As can be seen from the figures, for all three choices of r_T, the percentage increase in lifetime obtained by OML versus CMAX and MRPC increases as the sensor density increases. When $n = 100$ and $r_T = 30$, the average lifetime using OML is 31.9% larger than when CMAX is used and is 55% larger than when MRPC is used. These percentages at $r_T = 20$ are 16.5% and 53.9% and at $r_T = 10$, they are 4.3% and 53%. Although at $r_T = 10$, the energy consumed per session by OML and CMAX is virtually the same, OML and CMAX consume, on average, 33.2% less energy than consumed by MRPC. When $r_T = 20$ OML consumes, on average, 9.1% less energy per session than does CMAX and 35.5% less energey than consumed by MRPC; at $r_T = 30$, this reduction in energy consumed per session is 17.8% nd 35.5%.

7.4 Comparison with max-min $zPmin$

The CMAX algorithm was compared experimentally to the max-min $zPmin$ algorithm of [3] in [12]. From this comparison and our experimental results reported so far, we expect that OML and CMAX will result in better lifetimes than min-max $zPmin$. Hoever, the standing of min-max $zPmin$ relative to MRPC is not known. To resolve this, we ran a lifetime experiment using 20 randomly generated networks; each obtained by randomly placing 50 sensors on a 25×25 grid. The initial energy at each sensor was set to 30 units and an r_T value of 20 was used. The remaining

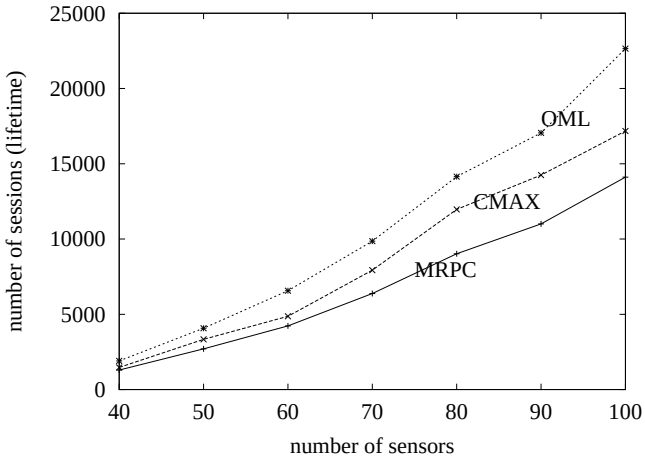


(a) Average lifetime

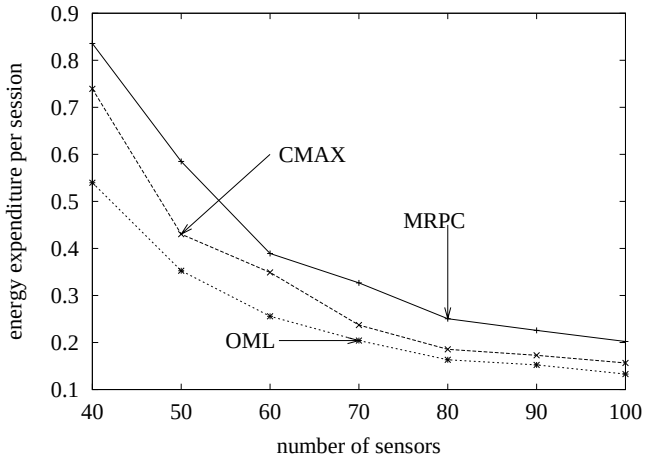


(b) Average energy per session

Figure 10: Lifetime and energy consumed as a function of n, $r_T = 20$



(a) Average lifetime



(b) Average energy consumed per session

Figure 11: Lifetime and energy consumed as a function of n, $r_T = 30$

parameters were set to the same values used in our earlier experiments.

Figure 12 shows the average lifetime over 10 different randomly generated request sequences for each of the 20 random networks. The results are consistent with those reported in [12], CMAX is superior to max-min $zPmin$. The new information is that min-max $zPmin$ is superior to MRPC.

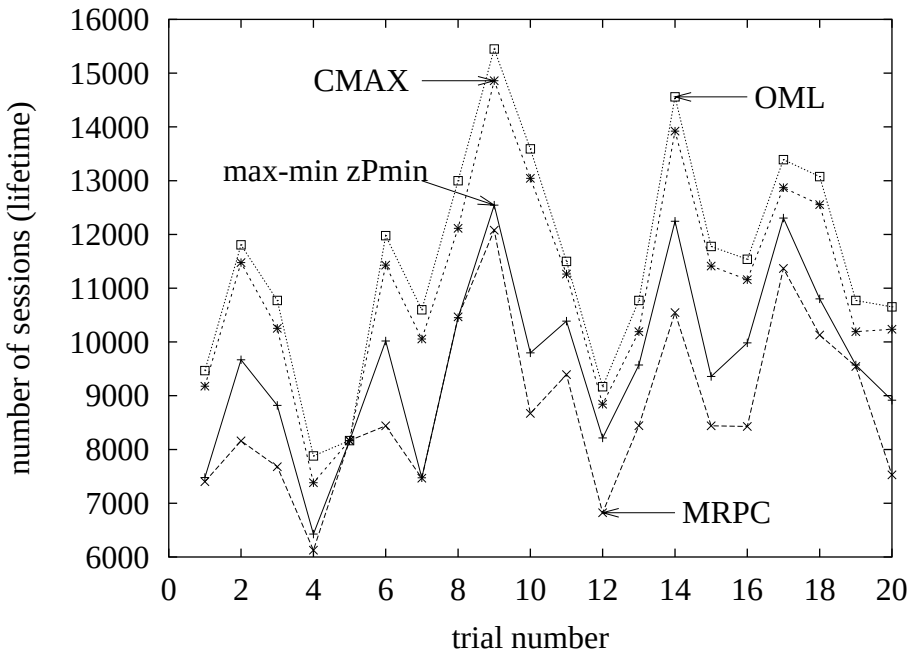


Figure 12: Lifetime under various algorithms

7.5 Network Capacity

As mentioned in Section 3, the CMAX algorithm was developed originally to maximize network capacity–the number of successful routes in a given time period. Online outing algorithms for capacity maximization may employ an admission control policy to determine which route requests they will service. Algorithms developed for lifetime maximization are required to service every route request for which there is a feasible path. We may use a lifetime maximization algorithm in a capacity maximization application by using the admission control policy "service the current route request is there is a feasible path for this request." We compared the performance of OML, CMAX and MRPC using the capacity metric. For our experiment, we generated 20 networks by randomly placing 20 sensors in a 10×10 grid. Each sensor started with 30 units of energy and r_T was set to ∞. For each of the 20 networks, a random sequence of 10,000 message routing requests was generated. The remaining parameters were as used in our earlier experiments.

Figure 13 shows the average number (for each of the 20 random networks, 10 request sequences of size 10,000 each were tried) of routing requests successfully completed by each of the 3 test algorithms. Network capacity using OML is approximately 6.7% higher, on average, than when

CMAX is used; the capacity is about 58.1% higher using OML rather than MRPC.

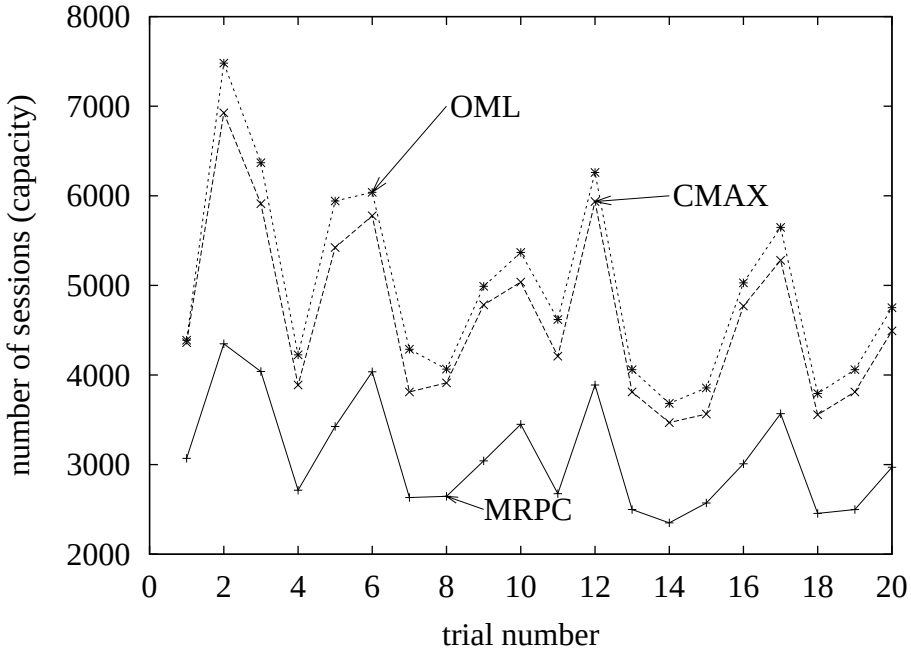


Figure 13: Capacity under various algorithms

8 Conclusion

We have shown that the lifetime maximization problem is NP-hard. We also have proposed a new online heuristic–OML–for lifetime maximization. Extensive simulations show that our new heuristic is superior to previously published heuristics for lifetime maximization both in terms of providing larger lifetime and in terms of sensitivity to algorithm parameters. Additionally, our proposed heuristic provides larger network capacity than provided by competing heuristics.

References

[1] G. Li and J. Aslam and D. Rus, "On-line power-aware routing in wireless ad hoc networks," *Proc. of Mobicom*, July 2001.

[2] K. Kar and M. Kodialam and T. V. Lakshman and L. Tassiulas, "Routing for network capacity maximization in energy-constrained ad-hoc networks," *Proc. of Infocom*, pp. 673–681, 2003.

[3] J. Aslam, Q. Li and R. Rus, Three power-aware routing algorithms for sensor netowrk, *Wireless Communications and Mobile Computing*, 3, 2003, 187-208.

[4] J. Chang and L. Tassiulas, Routing for maximum system lifetime in wireless ad-hoc networks,, *37th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sept. 1999.

[5] J. Chang, and L. Tassiluas, Energy conserving routing in wireless ad-hoc networks, *IEEE INFOCOM*, 2000.

[6] J. Chang and L. Tassiulas, Fast approximate algorithms for maximum lifetime routing in wireless ad-hoc networks, *IFIP-TC6 Networking 2000*, *LNCS*, 1815, Springer Verlag, 2000, pp. 702-713.

[7] C. Florens and R. McEliece, Scheduling algorithms for wireless ad-hoc sensor networks, *IEEE GLOBECOM*, 2002, 6-10.

[8] M. Garey and D. Johnson, *Computers and intractibility: A guide to the theory of NP-completeness*, W. H. Freeman and Co., 1979.

[9] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, *IEEE HICSS*, 2000.

[10] R. Kannan, L. Ray, R. Kalidindi, and S. Iyengar, Energy threshold constrained geographic routing protocol for wireless sensor networks, *Signal Processing Letters*, 1, 1, 79-84, 2003.

[11] R. Kannan and S. Iyengar, Game theoretic models for reliable path-length and energy constrained routing with data aggregation in wireless sensor networks, *IEEE JSAC*, 2004.

[12] K. Kar, M. Kodialam, T. Lakshman and L. Tassiulas, Routing for network capacity maximization in energy-constrained ad-hoc networks, *IEEE INFOCOM*, 2003.

[13] M. Maleki, K. Dantu, and M. Pedram, Power-aware source routing protocol for mobile ad hoc networks, *SLPED'02*, 2002.

[14] T. Melodia, D. Pompili, and I. Akyildiz, Optimal local topology knowledge for energy efficient geographical routing in sensor networks, *IEEE INFOCOM*, 2004.

[15] A. Misra and S. Banerjee, MRPC: maximizing network lifetime for reliable routing in wireless,, *IEEE Wireless Communications and Networking Conference (WCNC)*, 2002.

[16] S. Sahni, Data structures, algorithms, and applications in Java, 2nd Edition, Silicon Press, NJ, 2005.

[17] S. Singh, M. Woo, and C. Raghavendra, Power-aware routing in mobile ad hoc networks, *ACM/IEEE MOBICOM*, 1998.

[18] A. Spyropoulos and C. Raghavendra, Energy efficient communications in ad hoc networks using directional antenna, *IEEE INFOCOM*, 2002.

[19] I. Stojmenovic, and Xu Lin, Power-aware localized routing in wireless networks, *IEEE Transactions on Parallel and Distributed Systems*, 2000.

[20] C. Toh, H. Cobb, and D. Scott, Performance evaluation of battery-life-aware routing optimization schemes for wireless ad hoc networks, *IEEE ICC'01*, 2001.

[21] J. Wu, M. Gao and I. Strojmenovic, On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks, *Jr. Communications and Networks*, 4, 1, 2002.

[22] G. Zussman and A. Segall, Energy efficient routing in ad hoc disaster recovery networks, *IEEE INFOCOM*, 2003.