On Energy Aware Routing in Wireless Networks

Aravind B. Mohanoor, Sridhar Radhakrishnan*
School of Computer Science
University of Oklahoma, Norman, OK 73019
(aravindmc, Sridhar)@ou.edu
* Corresponding Author.

Venkatesh Sarangan
Computer Science Department
Oklahoma State University
Stillwater, OK 74078
saranga@cs.okstate.edu

Abstract—Online energy aware routing in wireless networks is the problem of finding energy efficient routes that maximize the network lifetime without the knowledge of future message flows. To maximize network lifetime, the paths for message flows are chosen in such a way that the total energy consumed along the path is minimized while avoiding energy depleted nodes. Finding paths which consume minimum energy and finding paths which do not use energy depleted nodes lead to conflicting objectives. In this paper, we propose a two-phased energy-aware routing strategy that balances these two conflicting objectives by transforming the routing problem into a multi-metric widest path problem. We find that the proposed approach outperforms the best known algorithm in literature. We also demonstrate a simple but insightful relationship between the total energy required along a path and the minimum remaining energy of a node along the path.

Keywords - Wireless networks, Energy aware routing, Combinatorial algorithm, QoS metrics.

I. INTRODUCTION

Energy management in wireless networks is of paramount importance due to the limited energy availability in the wireless devices. Since wireless communication consumes a significant amount of energy, it is important to minimize the energy costs for communication as much as possible by practicing energy aware routing strategies. Such routing strategies can increase the network lifetime. In this paper, we focus on developing routing strategies for a multiple hop wireless network which has significant energy constraints, like a multi-hop mesh network where all the nodes are powered by battery or other external power sources such as solar energy. One way to quantify network lifetime is through the number of packets that can be transferred in the network before the source and destination get disconnected from each other. A suitable energy-aware routing strategy for wireless networks is to use those wireless nodes with high energy levels and avoid those with low energy levels.

Wireless networks for energy-aware routing techniques are modeled as graphs wherein, a vertex is a wireless device and an edge between two vertices indicate that they are in direct communication range of each other. The weight on a vertex indicates the energy level available at that sensor node and the weight on an edge \((u, v)\) represents the amount of energy required by node \(u\) (resp. \(v\)) to communicate to node \(v\) (resp. \(u\)). The residual energy of a path is defined as the minimum energy level of any node in the path (metric 1). The max-min routing paradigm suggested in the literature [1, 4, 7] proposes to find the path where the residual energy is the maximum and forwards packets through this path termed as the maximum residual energy path. The energy consumed along a path (or simply the energy of a path) is the sum of the weights on the edges along the path (metric 2).

Notable routing strategies which utilize the concept of the residual energy (either directly or indirectly) proposed so far include MMBCR [7], MRPC [1] and max-min \(zP_{min}\) [4]. These research works also caution that merely using the residual energy strategy may lead to higher energy consumption in the network, since the energy consumed along the data forwarding path is not taken into consideration. They suggest that a good energy-aware routing technique should balance two different goals: choosing a path with maximal residual energy and choosing a path with minimal energy consumption. We note that the residual energy along a path is a concave metric\(^1\), whereas the energy consumed along a path is an additive metric.

The approach in max-min \(zP_{min}\) [4] attempts to balance metric 1 and metric 2 by calculating a path based on the residual energy levels, but then rejecting any path whose total energy is more than a factor \(z\) times the minimum energy path. We note that the quality of its solution depends on an empirically generated parameter \(z\), and does not always provide an optimal solution. The MRPC algorithm, which is a generalization of the MMBCR algorithm, uses the residual ‘packet capacity’ instead of the residual energy for optimization. As we will illustrate later, even the MRPC algorithm can fail to maximize a network’s lifetime. Chang and Tassiulas [2] combine metrics 1 and 2 into a single metric and run Dijkstra’s on this new metric. While it is a good heuristic, this method does not actually optimize either metric.

Park and Suhni [5] present the Online Maximum Lifetime (OML) heuristic, which is an enhancement of the CMAX algorithm presented by Kar et al [3]. OML uses a two-step approach where they remove those edges with low energy from the graph, and then run Dijkstra’s on a graph where the edge weights have been modified in such a way that the paths found usually use nodes with high energy levels and edges

---

\(^1\) For definitions of concave and additive metrics, see Wang and Crowcroft[8].
with low energy costs. They report the best performance in the current literature.

In this paper, we present a polynomial time combinatorial technique which can provide a good balance between metrics 1 and 2. The proposed technique first maximizes the concave metric (the residual energy of a path) and then minimizes the additive metric (energy consumed along a path). We qualitatively justify why this order of optimization – concave first, additive second – is better than the other possible order – additive first, concave second. On an illustrative topology, we show that the proposed two-phased routing prolongs the network lifetime better than other routing techniques. Our simulation studies also show that the performance of the proposed technique is superior to that of the best known routing approach proposed in the literature (Park and Sahni [5]). An additional incentive for using the proposed approach is that it allows us to address (i.e. it can be easily combined with) other QoS metrics such as delay, which can be beneficial for resource constrained networks.

The paper is organized as follows. Section II provides the definition of the network lifetime problem and the solution proposed. Section III discusses the performance of our approach and other approaches on a benchmark topology. In Section IV, we deduce a relationship between the total energy consumed along a path and the residual energy of edges on the graph, which provides additional insight into the nature of the problem. In Section V we discuss how other QoS metrics such as delay and error rate can be used in the second phase of our solution. We discuss the rationale behind the chosen order of optimizing the two metrics (in our problem the energy consumed along a path and the residual energy at the nodes) in Section VI. Section VII discusses our simulation setup and results. Section VIII concludes the discussion.

II. PROBLEM DEFINITION AND SOLUTION

Let $G = (V, E)$ represent a wireless network with nodes $V$ and edges $E$. Let $w(u)$ (also referred to as the residual energy of a node), $u \in V$ represent the available energy at node $u$. Let $c (u, v), (u, v) \in E$ be the energy required to transmit a packet from node $u$ to node $v$. We assume that $c (u, v) = c (v, u)$, for all $(u, v) \in E$.

Let $P(v_0, v_k) = v_0, v_1, \ldots, v_k$ be a path in $G$. The energy of the path $P(v_0, v_k)$ denoted $e(P(v_0, v_k))$ is

$$e(P(v_0, v_k)) = \sum_{i=0}^{k-1} c (V_i, V_{i+1})$$

The residual energy of a path $P(v_0, v_k)$ denoted $r(P(v_0, v_k))$ is

$$r(P(v_0, v_k)) = \min(v_i), 0 \leq i < k.$$  

When a packet is sent along $P(v_0, v_k)$, we need to perform the following energy decrease operation on each node along the path except on the node $v_j$: $w(v_j) = w(v_j) - c(v_i, v_{i+1}), 0 \leq i < k$. That is, after the packet is sent by a node, the energy level of the node is decremented by the amount of energy required to send the packet. In our model, we have not included the cost of reception explicitly to avoid clutter in our discussions. Such a cost can be easily incorporated in our proposed work.

Given a source $s$, a destination $t$, and a single packet to be routed, we can define two problems formally:

a. Minimum energy path problem: Find a path $P(s, t)$ with minimum $e(P(s, t))$.

b. Maximum residual energy path problem: Find a path $P(s, t)$ with maximum $r(P(s, t))$.

Let $G_0$ be set to the initial network $G$. Assume that $P_0(s, t)$ is a path in $G_0$. Now after routing a single packet along the path $P_0(s, t)$ and following the decrease operation we obtain a new network $G_1$. In the network $G_1$ the edge weights are the same as in $G_0$ but the nodes energy levels are different. If a node $u$’s energy level becomes 0 after the decrease operation the node $u$ and the edges $(u, v) \in E$ are removed from the network. For the second packet we can again find a path $P_1(s, t) \in G_1$ and the process continues until there exists no path between $s$ and $t$ in some network $G_k$. That is, we can send at most $k$ packets from $s$ to $t$ before the network is disconnected.

The goal of the network lifetime problem with respect to a source $s$ and destination $t$ is to find paths $P_0(s, t), P_1(s, t), \ldots, P_{k-1}(s, t)$, such that the value of $k$ is maximized.

Our solution to the network lifetime problem as follows. We modify the graph $G$ into an energy graph $EG = (V, E')$ as follows. We leave the vertices intact but replace each single undirected edge in $G$ with two directed edges. The weight of a directional edge in $EG$ is made equal to the difference between the originating node’s energy level and the transmission cost along the edge. This is also the residual energy of a node as defined in Li et al [4]. In Figure 1 (a) we have shown an example wireless network and in Figure 1 (b), the corresponding energy graph.

![Figure 1](image)

Given a source node $s$ and a destination node $t$, we then run our two-phased routing algorithm on this energy graph $EG$ to find a suitable path between $s$ and $t$. In Phase I, we apply a variant of the Dijkstra’s algorithm shown in Figure 2 to find a path with the maximum residual energy. Phase I of our solution will return a path whose residual energy will be the maximum in the network. Let the path returned by phase I have a residual energy of $B$. It is to be noted that there could be many paths in the network between $s$ and $t$ with a residual energy of $B$. 
In Phase II, we choose from the set of all paths with a residual energy of \( B \), a path which has the lowest energy consumption. If \( E' \) is the set of edges whose residual energy is less than \( B \), this can be accomplished by first pruning those edges from \( EG \) and using Dijkstra’s algorithm to find the least energy cost path on \( EG \). If there are many such paths, we arbitrarily choose one among them. It can be noted that this algorithm can also handle the energy cost of reception if such information were available. We would need to modify step 1 of the RELAX procedure to add the energy cost of reception. It must be noted that the pruning is temporary, in other words, the edges are restored before the next route computation.

Each time a path is computed, we will invoke Dijkstra’s algorithm twice in sequence. Hence it can be seen that our algorithm has a complexity equal to \( k \) times the complexity of Dijkstra’s algorithm, where \( k \) is the number of packets transmitted.

### MaxCapacity(EG)

// \( s \) - source node
// weight\((u,v)\) = capacity of edge \((u, v)\) in graph \( EG \)
// width\((u)\) = weight function for a node in graph \( EG \)

1. width\([s]\) = 0
2. width\([v]\) = weight\((s, v)\) if \( v \in \text{Adj}[s]\)
3. width\([v]\) = 0 for all other nodes

4. \( S \leftarrow s \)
5. \( Q \leftarrow V[EG] - s \)
6. while \( Q \neq \emptyset \)
   1. do \( u \leftarrow \text{EXTRACT-BEST}(Q) \)
   2. \( S \leftarrow S \cup \{u\} \)
   3. for each vertex \( v \in \text{Adj}[u] \)
      1. if \( v \) does not belong to \( S \)
      2. do RELAX\((u, v, \text{weight}(u, v))\)

### EXTRACT-BEST

1. return \( u \in Q \) where \( u \) has maximum width

### RELAX \((u, v, \text{weight}(u,v))\)

1. if width\([v]\) < min(width\([u]\), weight\((u, v)\))
2. width\([v]\) = min(width\([u]\), weight\((u,v)\))
3. \( \text{Pred}(v) \leftarrow u \)

**Figure 2:** Dijkstra’s shortest path algorithm modified to compute the maximum residual energy path.

---

### III. PERFORMANCE OF THE PROPOSED APPROACH

We compare the performance of the proposed two-phased approach with other existing approaches on an illustrative topology shown in Figure 3 [from Li et al. [4]].

In the network shown in Figure 3, each node (other than source \( 1 \) and destination \( n \)) has energy \( 20 + \gamma \). The weight of each edge (along the semi-circle) is set to 1, but the weight of each straight edge is set to 2. The energy of the source is infinite. Li et al. [4] state that using a single-pass max-min approach, it is possible that only twenty messages can be sent before the network gets disconnected. The authors then state that using the straight line edges \( 10(n-4) \) messages can be sent before the network gets disconnected [4], where \( n \) is the number of nodes in the network. The first message will take the path \((1, n-2, n)\). The second message will take the path \((1, n-3, n)\), and so on until the \((n-4)\)th message will take the path \((1, 3, n)\).

![Figure 3: How do we apply max-min on this graph?](image)

Banerjee and Misra[1] define the residual packet capacity as the number of packets which can be transmitted by a node at its current energy level. Directly applying MRPC on this graph is identical to max-min, and may end up sending at most 20 packets in the worst case. But they also provide another algorithm called the CMRPI, where they define a parameter \( \gamma \) which represents the threshold energy level of the critical nodes. When nodes reach this energy level, they shift from minimum energy routing to maximum residual capacity routing.

Suppose the parameter \( \gamma \) is set at 0.5 (representing 50% of node’s energy). The authors of the MRPC algorithm do not mention how they make the choice of minimum energy paths when there is more than one. Suppose we use the sequence \{(1, n-2, n), (1, n-3, n), ..., (1, 4, n), (1, 3, n)\}, we can send \( 5(n-4) \) messages using the straight line edges after which each forwarding node (except 2 and \( n-1 \)) will be left with energy 10 + \( \gamma \). Since we reach the threshold value, we start using the maximum residual capacity paths. Using the maximum residual capacity paths, only 10 more messages could be sent, for a total of \( 10 + 5(n-4) \) messages. In fact, we may not be able to send more than \( 10\gamma(n-4) + 20(1-\gamma) \) messages. The maximum value of this quantity happens when \( \gamma = 1 \) (which means packet capacity is never used), in which case we can still send only \( 10(n-4) \) messages.

On the other hand, if one were to use the two-phased approach we have proposed, the following paths will be used for routing. We will repeat the sequence \{(1, n-2, n-1, n), (1, n-2, n-1, n), (1, n-3, n), (1, n-4, n), (1, n-5, n), ..., (1, 4, n), (1, 3, n)\} before source and destination get disconnected. Consequently, it is easy to see that a total of \( 10(n-3) \) messages can be sent before the nodes run out of energy.

### IV. RELATIONSHIP BETWEEN THE TOTAL ENERGY AND THE RESIDUAL ENERGY OF PATHS

While the maximum residual energy path computation identifies the bottleneck edge and allows us to discover the maximum residual energy subgraph, we can also define what are called residual energy constrained subgraphs.
Definition: Let $EG(w)$ represent the subgraph constructed from the original residual energy graph $EG$, by pruning all those edges in $EG$ which have residual energies less than $w$.

For example, for the original graph $EG$ shown in Figure 4(a), we can observe that the graphs in Figures 4(b) and 4(c) show examples of residual energy-constrained subgraphs $EG(66)$ and $EG(85)$. The bidirectional edges have not been shown for the sake of clarity. The edges which are removed from the original graphs are such that their residual energies fall below the threshold limit along both directions. Using this construction, we can see that the total energy required for the minimum energy path on a subgraph increases as the residual energy constraint of the subgraph increases. If we were to repeatedly compute the minimum energy path for all the possible residual energy values of a graph (a graph can have at most $O(m)$ such discrete values, where $m$ is the number of edges) we would obtain the non-decreasing graph similar to the one shown in Figure 5. This can be easily proved.

Lemma 1:

Let $E_{\text{min}}(w)$ represent the minimum energy of a path in $EG(w)$. Given the residual energies of the graph $EG$ in increasing order as $(w_1, w_2, \ldots, w_m)$, i.e. $w_1 \leq w_2 \leq \ldots \leq w_m$, then $E_{\text{min}}(w_1) \leq E_{\text{min}}(w_2) \leq \ldots \leq E_{\text{min}}(w_m)$.

Proof:
Let $G_1 = EG(w_1)$ and $G_2 = EG(w_2)$ where $w_1 \leq w_2$. Any edge in $G_2$ also exists in $G_1$, by definition. Thus the minimum energy path in $G_1$ cannot have higher energy than the minimum energy path in $G_2$. In other words, $E_{\text{min}}(w_1) \leq E_{\text{min}}(w_2)$. By induction, we get the result.

We gain some useful insight from the relationship between the residual energy along a path and the minimum energy path possible for such a residual energy. For one, knowing that we have a lower energy path, we could avoid routing the packet along a path with identical residual energy but which consumes a much higher energy for the entire path. This is precisely the reason why a maximum residual energy path still turns out to be a poor choice in the benchmark topology considered in Figure 3. We can note that the path along the arc in the figure has an initial residual energy of 19, and a total energy of $(n-1)$ units. The path (1, n-2, n-1, n) also has a residual energy of 19 but consumes only 4 units of energy for transmission. When we perform such route computations repeatedly, it can lead to substantial savings in energy.

V. USING OTHER METRICS FOR THE SECOND PHASE

Using a similar argument as in Section IV, we can generalize to say that any path-based metric (such as delay, error rate, jitter) will have a similar graph. This has two implications for QoS metrics on wireless networks.

If our priority is to perform routing according to QoS constraints, then the problem of also ensuring that we choose energy aware paths is easily handled by using a maximum residual energy (widest) path approach. Since the maximum residual energy path computation yields a subgraph whose edges already have high residual energy, the other QoS constraints could be computed on the resulting subgraph. Thus an existing polynomial solution for the QoS constraint problem will not be worsened by adding energy awareness.

![Figure 4](image1.png)  
**Figure 4** Total energy of minimum energy path vs the constraint residual energy (width). (a) Original graph – this is also $EG(38)$, as 38 is the lowest residual energy edge in the residual energy graph. The minimum energy path has been marked with arrows and has energy = 10 units. (b) $EG(66)$ and minimum energy path = 16 units (c) $EG(85)$ and minimum energy path = 20 units.

![Figure 5](image2.png)  
**Figure 5** Total energy of path Vs residual energy
Contrast this with the issue of choosing the absolute minimum energy path which would make the original problem (e.g. minimum delay) NP-Complete, since the problem of computing paths which simultaneously optimize two additive metrics is known to be NP-Complete [8].

A second benefit comes from the use of the property mentioned in the earlier section. Figure 5 provides a clue as to the maximum possible residual energy we could use for finding a path with a given constraint value for the second stage metric. For example, we can plot the minimum possible value of the additive constraint (e.g. delay) on the y-axis of Figure 5 as a function of the residual energy. This means that given an upper bound on the additive constraint, we can find the highest residual energy for which it is feasible. Repeatedly utilizing this idea might actually ensure that the paths used are always having high residual energy while also satisfying the additive constraint. In general this should give us better energy awareness. In contrast, other QoS metrics cannot be combined in the Online Maximum Lifetime heuristic (which has the best performance in current literature) since it modifies the graph in such a way that the original information about the graph weights are lost.

VI. DISCUSSIONS ON THE ORDER OF OPTIMIZATION
We could also consider the other possible order of optimization for this problem, namely the additive metric first (minimum energy) and the concave metric second (widest). This is also called the widest shortest path. However, since the width is a concave metric, different paths with the same width can be constructed by simply removing the bottleneck nodes. However, the energy cost of a path is an additive metric and depends on all the edges in the path. If we do remove some edge(s), we face a much more difficult constraint in choosing other paths with the same energy cost. This means there can be several widest paths in a network (that may include almost all the nodes in the network) among which we could choose the one with the least energy cost. Over time, this would mean that the energy burden of forwarding packets is shared among many more packets, which is a desirable situation. However, there may be only a few minimum energy paths, and hence choosing the widest (maximum residual energy) among them may not utilize all the nodes in the network very effectively.

The order in which we wish to optimize the metrics plays an important role in determining the existence of a single-pass algorithm for our problem. Sobrinho [6] shows that there exists a single pass algorithm for determining a multi-metric path only when the metrics governing the path are isotonic. Sobrinho also proves that the shortest-widest path is non-isotonic. Since minimal energy path is the ‘shortest’ path in terms of energy consumption, and the maximum residual energy path is the ‘widest’, a single-pass algorithm for discovering the minimum energy-maximum residual energy path cannot be constructed.

VII. EXPERIMENTAL RESULTS

A. Simulation Settings
In our experimental study, we compare the performance of the proposed shortest-widest path (SWP) and on-line maximum life-time (OML) heuristic proposed by Park and Sahni [5]. Though several works have been proposed in the literature, reference [5] has established the superiority of OML over other existing works. Therefore, we compare SWP with OML alone.

Topologies Used: We use a topology which is identical to that used for OML. We randomly populate a 25×25 grid with 50 nodes. We add edges to the network if the nodes are within each others’ transmission range, which is decided by the transmission radius $r_T$. The energy cost of transmitting a single packet is calculated as $0.001 \times d^3$ where $d$ is the Euclidean distance between the nodes. These settings are identical to the ones used for OML.

Session failures tolerated: In the literature, network life-time is traditionally measured using the number of packets that can be transferred in the network until a session failure occurs. In other words, the life-time is measured until the time when two nodes get disconnected. In practice however, certain classes of networks such as sensor networks, prove to be useful and continue with their functioning even though a single node pair gets disconnected. Therefore, we generalize the definition of network life-time as “the number of packets that can be transferred in the network until $s$ session failures are occur”, where $s$ is a parameter to be set by the network manager. The value for $s$ can be set based on the number of session failures that can be tolerated by the application. We study the performance of the routing schemes under different values of $s$.

Session Length: Another parameter that we generalize in our simulation study is session length. Earlier works assume that a single packet is transmitted in a session between a given node-pair. However, in reality, it is highly likely that multiple packets will be exchanged in a session between two nodes. Therefore, in our experiments, we assume that $k$ packets are transmitted in a session between a given node-pair. We vary the value of $k$ and observe the performance of the different routing schemes. As in other works in the literature, we calculate the route afresh for each packet transmission.

Traffic pattern used: We conduct our experiments assuming a any-to-any communication model, i.e. source-destination pairs are selected at random and packets are transmitted between them.

Unless otherwise mentioned, we use the following default values: there are 50 nodes placed randomly on a 25×25 grid, the transmission radius is set to 8, the session length is set to 1 (single packet), the number of session failures tolerated is set to 1 (first failure to find a route), the initial energy level for
each node is set to 30 and any-to-any communication pattern is assumed.

B. Effect of transmission radius

We vary the transmission radius from 7 to 30 and evaluate the performance of SWP and OML algorithms (Figures 6 and 7). We can notice that SWP equals or better the lifetime of OML for all values of transmission radii. However, it must be noted that a higher value of transmission radius comes with an associated energy cost for the nodes, which will then lose energy much more rapidly when not transmitting data. Figure 7 shows the remaining energy as a fraction of the total initial energy in the network. We notice that SWP can actually send more packets using lesser energy.

![Figure 6 Lifetime vs Transmission radius](image1)

![Figure 7 Energy Remaining vs transmission radius](image2)

![Figure 8 Lifetime vs session failures](image3)

![Figure 9 Energy remaining vs session failures](image4)

C. Effect of session failures tolerated

The number of session failures tolerated can be a number \( s > 1 \), as set by the network manager. We vary \( s \) from 1 to 10 and analyze the performance of the two algorithms. We notice that when we allow the number of session failures to be greater than one, we benefit from an improved lifetime.

Figures 8 and 9 show the performance of SWP and OML as the number of session failures tolerated is increased. From the figures we again observe that, SWP outperforms OML with respect to both network lifetime and the average residual energy in the nodes. We use an initial energy level of 50, to better highlight the performance difference between the two algorithms. The lifetime is measured by calculating the average over ten different runs at the specific session failure point. In other words, we find the number of packets transmitted before the \( s \)th session failure over multiple runs and use their average.

D. Effect of session length

When we need to transmit more than a single packet from a source to a destination, the length of the session is (perhaps much) greater than 1. So we evaluate the performance of the two algorithms when the number of packets transmitted per session, i.e. the session length, is varied to be more than 1. We compute fresh routes for each packet transmitted for both the
algorithms. We compare the performance of SWP and OML for session lengths of 1, 10, 25 and 100 in Figures 10 and 11. Again, we find that the performance of the SWP algorithm is better than that of OML.

E. Effect of node density

Figures 12 and 13 show the relative performance of the two algorithms when the number of nodes in the network is increased. We vary the number of nodes from 40 to 100 (in increments of 10) and find the lifetime which is achieved. We notice the same trends – SWP consistently outperforms OML as the node density increases.

F. Remarks

The number of path calculations is the same as the number of packets which need to be sent. When we need to send a lot of packets, we could avoid this situation by constructing the energy graph based on $k$ messages, where $k$ is some predetermined constant. Gathering information about the energy levels of all nodes has an associated energy cost (although all the previous energy aware algorithms suffer from the same limitation). The effects of this problem can also be partially offset by calculating the energy graph for $k$ messages at a time.

VIII. CONCLUSIONS

The online energy aware routing problem is conducive to a solution using the widest path approach. We have shown through simulations that the proposed approach easily outperforms the best known solution in the literature. An additional incentive for using the proposed approach is that it allows us to address other QoS metrics such as delay, which can be beneficial for resource constrained networks. This is
due to the combinatorial nature of the solution which makes it possible to enumerate and discover all favorable route choices.

REFERENCES


