Maximum Flow-Life Curve for a Wireless Ad Hoc Network

Timothy X Brown
Electrical and Computer Engineering
University of Colorado
Boulder, CO 80309-530
timxb@colorado.edu

Harold N. Gabow
Computer Science
University of Colorado
Boulder, CO 80309-430
hal@cs.colorado.edu

Qi Zhang
Electrical and Computer Engineering
University of Colorado
Boulder, CO 80309-425
qi.zhang@colorado.edu

ABSTRACT

This paper proposes a new power aware routing objective for an ad hoc network of battery-limited wireless nodes—the maximum flow-life curve—that maximizes the traffic flow utility over time. The objective improves upon related objectives such as minimizing the total power or maximizing the time to network partition. To find a routing that maximizes the flow-life curve, we prove an equivalence with a simpler problem and present an algorithm based on linear programming. The efficiency and fairness of the objective are demonstrated on several examples.

1. INTRODUCTION

Consider a wireless ad hoc network where each node has a limited battery energy supply used mainly for data transmission. The nodes in this paper have fixed position and between any two nodes, there are certain data transmission requirements. Since it is an ad hoc network, all the nodes are assumed to be able to forward packets, i.e., all the nodes can work as relay nodes. For small battery operated computers, energy usage can be dominated by the wireless transmission. To reduce energy consumption, the transmitted power level can be reduced from the maximum power to a level that is lower but still sufficient for the receiver to receive the data correctly. We expect that this can have a significant effect on the energy consumption since the required transmit power varies rapidly with distance. For a separation of d, a typical path-loss equation yields a required transmit power proportional to d^n where $n = 4$ is common [12]. Traditional routing will attempt to limit the number of hops, the congestion, or the delays for a given packet connection. This paper explores how transmission power can be incorporated into routing decisions so as to minimize the impact of energy limitations.

Power management is often cited as an important aspect of ad hoc routing [6, 10]. Most early works in this area focus on shortest path routing that minimizes the number of hops based on pre-choosing a strong connected backbone network [7, 9]. However, the minimum number of hops does not imply minimum energy consumption. Because of the non-linear attenuation, multiple shorter hops can use less power than one large hop. Power has a role in achieving topology objectives [11], but in our model, power levels (and thus topology) are driven by the needs of individual traffic flows. A number of works focus on minimizing network energy consumption [13, 16, 18]. However, while the total network power is minimized, some relay nodes will be sacrificed unfairly, and cause the network to partition sooner. This occurs when a node is located between two clusters and all minimum energy routes between the clusters traverse the intermediate node. In [3, 4, 16], they maximize the time until the first node exhausts its energy, the so-called network life. This problem is first proved solvable as a linear programming problem in [3], while [4] further extends the problem to a multi-commodity case. Such network lifetime criteria can focus routing on a single outlier node which will exhaust its energy quickly no matter what routing is used and provide little guidance for the remaining nodes [2]. Although closely related, we do not consider here power aware data link layer or MAC modifications [1, 8, 15, 17].

Here we further explore the power aware routing problem, and get a complete theoretical solution for the so-called maximum flow-life curve of a network that better captures network-wide objectives. In Section 3 we define and analyze the natural extension of the network life criteria, the maximum node-life curve. In Section 4, we define the maximum flow-life curve of a network and prove a correspondence to the maximum node-life curve. In Section 5, we derive a linear programming based algorithm to compute the maximum flow-life curve. In Section 6 we present several examples of the algorithm. We start with formal model definitions.

2. MATHEMATICAL MODEL

The wireless ad hoc network studied in our work can be described as a directed graph $G(V, A, E, F, g)$, where V is the set of all nodes $\{v_i,\ i = 1, 2, \ldots, N\}$, A is the set of directed arcs v_iv_j, E is the initial energy of each node $\{E_i, i = 1, 2, \ldots, N\}$, F is the set of traffic flow requirements, $\{f_{sd}\}$, $s, d \in V$, and g is the direct link cost $\{g^s_{ij}, g^r_{ij}, i, j = 1, 2, \ldots, N\}$.

In the link costs, g^s_{ij} is the transmission power needed by sender i per unit flow to j, and g^r_{ij} is the receive power needed by the receiver j per unit flow from i. The power costs are all non-negative and finite; if two nodes can not communicate with finite power, there is no edge v_iv_j in A. The link costs factor in media access control and other overhead. For example, given fixed size packets with β data bytes, if ϵ is the total energy to send the packet including packet headers reservation packets, acknowledgment packets, packet processing, etc.; then the average cost to send one byte of data is ϵ/β. Further, the specific protocol would determine how this cost falls on the transmitter and receiver.

Nodes start with fixed energy and only spend energy communicating. A node is *exhausted* and can no longer participate in the network communication when it runs out of energy. Nodes that are not battery limited (e.g. connected to a power source or vehicle mounted) have $E_i = \infty$.

A flow, f_{sd}, represents the long term rate of data transmission from source node s to destination node d. Node pairs not communicating have $f_{sd} = 0$. To simplify the analysis, we focus on the low utilization case where collisions are negligible and every node has sufficient bandwidth to carry any set of flows. A flow is *exhausted* and no longer uses network resources when the network has no path containing only unexhausted nodes.

A flow can be divided in arbitrary fractions across different paths from a source to destination. A routing scheme r is how we divide the flows in F across different paths, and is defined as follows. Let $p_k = \{v_{i_0} = s, v_{i_1}, v_{i_2}, \ldots, v_{i_{l-1}}, v_{i_l} = d\}$ be an l hop path from source s to destination d. The intermediate nodes, $v_{i_1}, v_{i_2}, \ldots, v_{i_{l-1}}$ are relay nodes in this path. The subscript k indexes over all possible paths. Let $s(k)$ and $d(k)$ be the source and destination for path k. A routing is defined as $r = \{x_k(r)\}$, where $x_k(r)$ is the flow r allocates to p_k.

If $x_k(r) > 0$, then energy is drawn from the nodes along path p_k. The energy is drawn from node v_i at rate $a_{ik}x_k(r)$, where

$$a_{ik} = \begin{cases} 0 & \text{if } i \text{ is not in path } p_k \\ g^s_{i,i_1} & \text{if } i \text{ is the source for path } p_k \\ g^r_{i_{l-1},i} & \text{if } i \text{ is the dest for path } p_k \\ g^r_{i_{j-1},i} + g^s_{i,i_{j+1}} & \text{if } i \text{ is node } j \text{ on path } p_k \end{cases} \quad (1)$$

Let $I_{sd} = \{k | s(k) = s \text{ and } d(k) = d\}$ be the set of all indices for paths from s to d. The networks that we consider have a commitment to carry all flows between active nodes, as long as this is possible. A routing is valid if

$$\sum_{\{k \in I_{sd}\}} x_k(r) = f_{sd} \text{ for all } s, d.$$

If s and d are disconnected so that f_{sd} is exhausted, then we no longer consider the flow f_{sd}. In particular, the routing is no longer required to carry the flow, and even if $x_k(r) > 0$ for $k \in I_{sd}$ it does not draw power from any nodes.

Arithmetic on routings is defined in the obvious way, i.e. $r_1 + r_2 = \{x_k(r_1) + x_k(r_2)\}$ and for a scaler α, $\alpha r = \{\alpha x_k(r)\}$.

The next two sections use this model and definitions to define two routing objectives.

3. MAXIMUM NODE-LIFE CURVE

This section defines the maximum node-life curve and develops three properties of the maximum node-life curve central to the paper. Let $n(t, r)$ be the number of unexhausted nodes at time t under some routing, r. We call this function the *node-life curve* of a network. Nodes have a fixed initial supply of power so that $n(t, r)$ is a nonincreasing function with time as nodes exhaust their energy. The number of nodes decreases at discrete times, so called *drop points*. Three life-curve examples are shown in Figure 1a.

If $n_1(t) = n(t, r_1)$ and $n_2(t) = n(t, r_2)$ are different node-life curves for a network, according to different routing schedules, we say $n_1 > n_2$, if and only if: $\exists t_0$, s.t. $\forall t < t_0$, $n_1(t) = n_2(t)$, and $n_1(t_0) > n_2(t_0)$. In words, n_1 is better than n_2 if it keeps the most nodes alive longer. For example, in Figure 1a, $n_1 > n_3 > n_2$. We call n^* the *maximum node-life curve* of a network, if and only if $\forall n$, $n^* \geq n$.

The node-life curve considers more than the first node or node set that dies, it also tells how many nodes will be exhausted, and about the second time point for a second node or node set, and so on. Here we are not only trying to maximize the time for the initial network partition, but also trying maximize the size of the remaining network, and the time until the second partition, etc. In Figure 1a, we say that $n_1 > n_2$ because the time until the first node exhausts is 50% longer in n_1 than in n_2. We say $n_1 > n_3$ because fewer nodes exhaust at the first drop point in n_1 than in n_3. Although $n_1 > n_2$, the time until the *last* node exhausts is longer in n_2. But, this is at the expense of other nodes exhausting earlier in n_2. In general, we consider communicators who prefer more communication sooner rather than later. An alternative criteria might consider the total life volume (i.e. the area under the node-life curve). This is not as useful as it first sounds since this can easily be achieved by blocking all but the least costly flows. Deciding which flows can be blocked is a matter of policy which is outside the scope of this paper.

The node-life curve of a network depends on the routing scheme applied, so our goal is to find a routing scheme which can achieve the maximum node-life curve which we denote a *maximum node-life curve routing.*

To get a maximum node-life curve and the corresponding routing, first, we need to find the longest possible time points until nodes are exhausted and then we need to find the smallest exhausted node set at each of these drop point, and then finally we can get the maximum node-life curve. The following variables will be useful in our discussion:

$T_j(r)$: the time node j is exhausted under routing r;

τ_i: time of the i^{th} drop point in the maximum node-life curve;

R_i: the set of routings that guarantee the i^{th} drop point is τ_i;

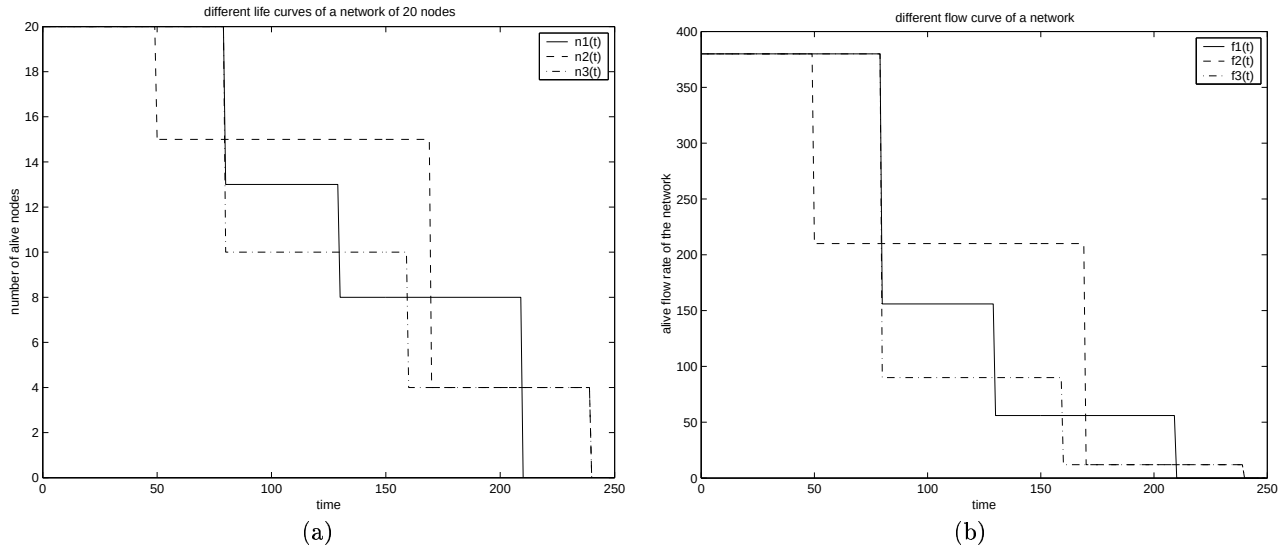

Figure 1: Three node-life curves (a) and flow-life curves (b) in a 20 node network.

$D_i(r)$: the node set exhausted at τ_i, when routing $r \in R_i$ is implemented;

$\widetilde{D}_i$: the smallest node set exhausted at τ_i;

$\widetilde{R}_i$: the set of routings in R_i which yield $\widetilde{D}_i$;

S_i: the set of nodes still alive after τ_i in the maximum node-life curve.

Formally, we define these values iteratively:

$S_0 = V$;
$\widetilde{R}_0$ = all the possible routing for $G(V, A, E, F, g)$;
$i = 0$;
do {

$\tau_{i+1} = \max_{\forall r \in \widetilde{R}_i}\{\min_{\forall j \in S_i}(T_j(r))\}$;
$R_{i+1} = \{r | r \in \widetilde{R}_i, \min_{\forall j \in S_i}(T_j(r)) = \tau_{i+1}\}$;
$D_{i+1}(r) = \{j | T_j(r) = \tau_{i+1}\}, \forall r \in R_{i+1}$
$\widetilde{D}_{i+1} = D_{i+1}(r)$ where $r = \text{argmin}_{r \in R_{i+1}}\{\|D_{i+1}(r)\|\}$;
$\widetilde{R}_{i+1} = \{r | D_{i+1}(r) = \widetilde{D}_{i+1}\}$;
$S_{i+1} = S_i - \widetilde{D}_{i+1}$;
$i = i + 1$;

} while $\|S_i\| < \|S_{i-1}\|$

The termination test is because some nodes may have infinite power while other nodes may remain alive even after all source and destination nodes and their associated flows are exhausted. We will show three properties of the maximum node-life curve in three lemmas.

LEMMA 3.1. *In the maximum node-life curve, $\forall \tau_i$, $\widetilde{D}_i$ is unique.*

That is: in the maximum node-life curve, the exhausted node set for each drop point is unique.

Proof: If $\widetilde{D}_i$ is not unique, then there exist two distinct smallest node sets $D_i(r_1) \neq D_i(r_2)$, such that $\|D_i(r_1)\| = \|D_i(r_2)\| = \left\|\widetilde{D}_i\right\|$. Define $r = \frac{1}{2}(r_1 + r_2)$. It follows that $D_i(r) = D_i(r_1) \bigcap D_i(r_2)$. Since $D_i(r_1) \neq D_i(r_2)$, $\|D_i(r)\| < \|D_i(r_1)\| = \left\|\widetilde{D}_i\right\|$. Thus we can find an even smaller exhausted node set at time τ_i, contrary to our definition for $\widetilde{D}_i$. So the assumption is false, and $\widetilde{D}_i$ is unique. △

LEMMA 3.2. *There exists a single routing, r, that achieves the maximum node-life curve.*

That is: a routing for a flow can be chosen so it is optimal even as other nodes dies.

Proof: Let $\tau_{i'}$ be the final drop point in the maximum node-life curve. Since $\widetilde{R}_{i+1} \subseteq \widetilde{R}_i \ \forall i$, $r \in \widetilde{R}_{i'}$ is optimal. △

LEMMA 3.3. *In the maximum node-life curve, $\forall i$, $\exists f_{sd}$ such that f_{sd} is exhausted at time τ_i.*

That is: in the maximum node-life curve at least one flow is exhausted at each drop point.

Proof: Let r be the initial routing. For any $v \in \widetilde{D}_i$, $\exists f_{sd} > 0$ which is routed in r using v. Since at time τ_i no flow is exhausted, f_{sd} continues after τ_i and it must use some new routing $r' \neq r$. But, this contradicts the previous lemma. △

Thus, we have shown the maximum node-life curve can be achieved by a single routing where at each drop point a unique node set and at least one flow is exhausted.

4. MAXIMUM FLOW-LIFE CURVE

The above maximum node-life curve treats all nodes equally. However, the nodes are less important than the communication in the flows. We want to keep as many flows active as possible over time. Furthermore, different flows may have different utility. For simplicity in this paper, a flow's utility will be equal to its rate. The utility generalizes to any positive utility per flow.

Let $f(t, r)$ be the sum of all flow values that can be sent at time t with routing r. We call $f(t, r)$ the *flow-life curve* of a network. As with $n(t, r)$, $f(t, r)$ decreases with time. Three examples are shown in Figure 1b.

If $f_1(t) = f(t, r_1)$ and $f_2(t) = f(t, r_2)$ are different flow-life curves for a network, according to different routing schedules, we say $f_1 > f_2$, if and only if: $\exists t_0$, s.t. $\forall t < t_0$, $f_1(t) = f_2(t)$, and $f_1(t_0) > f_2(t_0)$. In words, f_1 is better than f_2 if it keeps the most flow utility alive longer. We call f^* the *maximum flow-life curve* of a network, if and only if $\forall f$, $f^* \geq f$. A routing that achieves f^* we call a maximum flow-life curve routing.

By Lemma 3.3 we know a maximum node-life curve routing will cause at least one flow to exhaust at each drop point. But, what is the exact relationship? We prove the set of drop points is the same for both the maximum node-life curve and the maximum flow-life curve. What's more, a maximum node-life curve routing is also a maximum flow-life curve routing.

Let τ_i^f be the time for the i^{th} drop point in the maximum flow-life curve. Define S_i^f and $\widetilde{D}_i^f$ to be S_i and $\widetilde{D}_i$ as in Section 3, but with respect to τ_i^f, instead of τ_i. With these definitions we state the main theorem of the paper.

THEOREM 4.1. *A maximum node-life curve routing is also a maximum flow-life curve routing.*

Proof: We use induction on the set of drop points and the smallest set of nodes exhausted at each drop point. As the induction base define $\tau_0 = \tau_0^f = 0$ and $\widetilde{D}_0 = \widetilde{D}_0^f = \emptyset$. Next suppose $t_{i'} = t_{i'}^f$ and $\widetilde{D}_{i'} = \widetilde{D}_{i'}^f$ $\forall i' < i$ then we show $t_i = t_i^f$ and $\widetilde{D}_i = \widetilde{D}_i^f$

If $\tau_i^f < \tau_i$, then a flow drops even before any new node would be exhausted with the maximum node-life routing and so can not be a drop time in a maximum flow-life routing. If $\tau_i^f > \tau_i$, some new nodes die at τ_i, but no flow is exhausted, contrary to Lemma 3.3. So $\tau_i^f = \tau_i$.

Suppose under a maximum flow-life routing the nodes $\widetilde{D}_i^f$ are exhausted at τ_i and $\widetilde{D}_i^f \neq \widetilde{D}_i$. If $\widetilde{D}_i^f$ is not a superset of $\widetilde{D}_i$ then as in Lemma 3.1, we could find a smaller set of nodes to be exhausted at τ_i contrary to the definition of $\widetilde{D}_i$. If $\widetilde{D}_i^f \supset \widetilde{D}_i$, then there exists a node $v \in \widetilde{D}_i^f$, but $v \notin \widetilde{D}_i$. We know an exhausted node will do no help for the remaining flows, so v is unnecessarily exhausted, contrary to the definition of $\widetilde{D}_i^f$. So $\widetilde{D}_i^f = \widetilde{D}_i$.

Thus a maximum node-life curve routing is a maximum flow-life curve routing since it produces the same drop times and exhausted sets. □.

Now the problem is how to compute the maximum node-life curve.

5. MAXIMUM FLOW-LIFE CURVE ALGORITHM

This section describes an algorithm to find the maximum flow-life curve routing. The algorithm is based on linear programming (LP) using the revised simplex algorithm [14], and works well in practice. The algorithm directly computes the maximum node-life curve routing, which by Theorem 4.1 is also a maximum flow-life curve routing. We first describe the LP, and then how we solve the LP efficiently.

5.1 LP formulation

Lemma 3.2 implies that a path p_k in the maximum node-life routing is used until we cease transmitting messages from $s(k)$ to $d(k)$ entirely. We give an LP formulation based on this principle. We use the following variables:

t_{sd}: the time we cease transmitting messages from s to d. We route flow from s to d at the required rate f_{sd} from time 0 until time t_{sd}.

ϕ_k: the total flow that the routing sends through path p_k. So if p_k goes from s to d the transmission rate through p_k is ϕ_k / t_{sd}.

The maximum node-life routing satisfies these linear constraints:

(1) $\sum_{k \in I_{sd}} \phi_k = f_{sd} t_{sd}$ for all s, d with $f_{sd} \in F$

(2) $\sum_{k \in I_F} a_{ik} \phi_k \leq E_i$ for all nodes v_i

where a_{ik} is defined in (1).

We use these constraints to formulate LPs that find the maximum node-life curve. Assume we have determined the drop points $t_0 = 0, t_1, \ldots, t_{i-1}$ $(i \geq 1)$. We will have also determined two more pieces of information: We will know the sets $\widetilde{D}_0$, $\widetilde{D}_1, \ldots, \widetilde{D}_{i-1}$ of nodes that become exhausted at the corresponding drop point. Also we will know every value of t_{sd} that is $\leq t_{i-1}$.

We first describe an LP called LP_i that sets variable t to the next drop point t_i. The constraints of LP_i are the above (1) and (2) with this change: In (1), set t_{sd} to the variable t if t_{sd} is still unknown; in the opposite case set t_{sd} to its known value. The objective is to maximize t. We solve LP_i to determine the next drop point t_i.

Next we determine $\widetilde{D}_i$. We also must find the demands f_{sd} that drop to 0 after time t_i, i.e, $t_{sd} = t_i$. The latter is simple given $\widetilde{D}_i$: For each positive f_{sd} with t_{sd} still unknown, set $t_{sd} = t_i$ if there is no path from s to d using the remaining unexhausted nodes. We find $\widetilde{D}_i$ by solving a sequence of LPs. For each node v_j let s_j be the slack variable for constraint (2). Initialize a set D to all the nodes that have

equality in (2) (i.e., $s_j = 0$) but are not exhausted at earlier drop points. We will delete nodes from D until it becomes the desired set $\widetilde{D}_i$.

We will reference each of the new LPs as LP_i'. Each LP_i' has the same constraints as LP_i, with t set to the constant t_i. The objective of each LP_i' is to maximize $\sum_{j \in D} s_j$.

We repeat the following procedure until the correct set D is determined: Solve LP_i'. If the optimum objective is 0 then the current set D equals $\widetilde{D}_i$. Otherwise, i.e., when the optimum is positive, delete all nodes v_j with $s_j > 0$ from D. Then continue with the next repetition, solving the new LP_i', etc.

5.2 Solving the LP

Formally, we can state the linear program as $\max_X C \cdot X$ given $HX = B$. We describe how to solve the initial LP. Later LP are solved similarly. For simplicity, let the flows be numbered 1 to M with flow i having rate f_i, and assume that the paths in I_F are numbered 1 to $||I_F|| = L$. Then X, H, B, and C are defined as:

$$\begin{aligned} X' &= [t, \phi_1, \ldots, \phi_L, s_1, \ldots, s_N] \\ H &= \begin{bmatrix} 0_N & A & I \\ -F & R & 0 \end{bmatrix} \\ C' &= [1, 0_{L+N}] \\ B' &= [E_1, \ldots, E_N, 0_M] \end{aligned}$$

where prime indicates transpose, I is an identity matrix, 0 is an all zero matrix, 1_n and 0_n denote vectors of n 1's and 0's. $A = \{a_{ik}\}$ is the $N \times L$ array of power costs per unit flow. $R = \{r_{ik}\}$ is an $M \times L$ matrix where $r_{ik} = 1$ iff flow i can use path k. F is the vector of flow requirements.

H is an $(N+M) \times (1+L+N)$ matrix. Since $L \approx (N-2)!Me$ (for a fully connected network), the number of variables in X is large and solving the linear program directly is not tractable. Therefore we use a delayed column generation technique with the revised simplex method [5, 14].

Since $||F|| = M$ and $||V|| = N$ there are only $M + N$ constraints and it follows from the fundamental theorem of linear optimization that at any time at most $N + M$ variables in X are non-zero (i.e. at most $N + M$ non-zero flows). So the problem reduces to tracking which subset of $N+M$ variables can be non-zero. As an initial basis we can find the shortest (lowest power) path for each of the M flows plus the N slack variables.

To solve the primal problem, we use information from the dual linear program to iteratively choose the next entry and exit variables from the basis. Let H_i be the columns of H corresponding to the basis at iteration i of the linear program. Similarly define C_i. The dual linear program is $\min_Y B \cdot Y$ given $H'Y = C$.

We solve the problem with the following iterations:

Step 1: In the primal problem, restricting only to t and the slack variables, add every variable to the basis that is possible.

Step 2: Solve the system $H_i'Y = C_i$ for

$$Y = [y_1^e, \ldots, y_N^e, y_1^f, \ldots, y_M^f].$$

Step 3: Choose an entering column with positive cost. There are $1 + L + N$ constraints to consider. If any constraint is met in the dual problem, then the corresponding variable in the primal problem can enter the basis. From Step 1, the only variable that can enter the basis is one of the path variables. The corresponding path constraint follows from the constraint matrix H, and cost weights C. For flow j over path k the constraint is:

$$\sum_i y_i^e a_{ik} < -y_j^f$$

This implies a very large number of comparisons. For each path, k, the path variable comparison is simply a weighted sum of the path costs. For a given flow, the weights are fixed and they are all positive since the a_{ik} are positive and since $y_i^e < 0$ is precisely the constraint for a slack variable to enter (but this is not possible given Step 1). It is enough to check only the shortest path (i.e. if the shortest path can not enter then no path can enter for f_{sd}). Since the weights are fixed and positive the shortest path can be computed via Dijkstra's algorithm. The link weights can be computed from the definition of a_{ik}:

$$w_{ij} = y_i^e g_{ij}^s + y_j^e g_{ij}^r.$$

Thus the $L \approx (N-2)!Me$ comparisons are reduced to at most M computations of Dijkstra's algorithm.

If a variable can be added to the basis, we go to the next step. Otherwise, if every comparison fails, then the current basis is optimal and the linear program is finished.

Step 4: Let h be the new entering column from H. Solve the system $H_i d = h$ for vector d. Let $\widehat{X} = [\widehat{x}_1, \ldots, \widehat{x}_{N+M}]$ be the variables in X in the current basis. Find

$$u = \min_{\{i | d_i > 0\}} \{\widehat{x}_i / d_i\},$$

and

$$i^* = \text{argmin}_{\{i | d_i > 0\}} \{\widehat{x}_i / d_i\}$$

Replace $\widehat{X} = \widehat{X} - ud$, and set $\widehat{x}_{i^*} = u$. $H_{i+1} = H_i$ with column i^* replaced with h. Go back to Step 1.

Note that it is possible that $u = 0$ and we take a degenerate simplex step. It is possible that this leads to cycling among degenerate steps that do not improve the solution. We found that occasional degenerate steps did occur and it was sufficient to prevent cycles by randomizing the order that flows were checked in Step 3.

The algorithm is stated for the initial LP. For later LP, slack variables for exhausted nodes are eliminated, and for exhausted flows the total traffic volume for that flow is changed

from a variable to a constant in H and B. Further, the previous basis can serve as an initial basis for the next LP.

5.3 Algorithm Time Complexity

The algorithm consists of several components. It computes the timing and nodes at each drop point. The number of drop points is at most N. The simplex algorithm typically requires $NumConstraints \log(NumVariables)$ steps, i.e. $(1+M+N)\log L \approx (M+N)N \log N$ steps. In the worst case, it can be exponential in the number of variables. A simplex step uses Dijkstra's algorithm which is $O(N^2)$. Dijkstra's may need to be computed for each node. In addition, working between the primal and dual problem requires solving $N+M$ linear equations and unknowns, an $O((N+M)^3)$ computation. Solving the linear equations dominates the time complexity of Dijkstra's algorithm. Thus, the overall routing time complexity is typically $N^2(M+N)^4 \log N$.

To give a measure of the timing, Example 2 in the next section has 12 nodes and 20 flows and required a 5.5sec on a 300MHz CPU. The total number of iterations of Steps 1–4 in the algorithm was 290.

We note there is a polynomial time algorithm which solves the problem if we use a similar algorithm, changing the LP so it has polynomial size. To do this we use a network flow formulation rather than a path-based formulation. That is, each traffic requirement f_{sd} has a set of variables ϕ_{ij}^{sd}, $i,j \in V$. The linear constraints express the following facts:

(a) ϕ^{sd} describes a flow from s to d, i.e., flow is conserved at each node $i \neq s,d$,

$$\sum_j \phi_{ij}^{sd} = \sum_j \phi_{ji}^{sd} \quad \forall i \neq s,d;$$

(b) the required flow is routed as in (1) of Section 5.1, i.e.,

$$\sum_i \phi_{si}^{sd} = f_{sd} t_{sd};$$

(c) the energy constraints (2) are satisfied, i.e.,

$$\sum_{s,d,j} \phi_{ij}^{sd} g_{ij}^{s} + \phi_{ji}^{sd} g_{ij}^{r} \leq E_i.$$

This LP has polynomial size and so can be solved in polynomial time [14]. The rest of the algorithm is similar to Section 5.1.

We did not use this LP in our implementation because we believe the formulation of Sec. 5.1 gives better results in practice. Instead of M traffic constraints (1), this LP has $(N-1)M$ constraints. The matrix stored is an order of magnitude larger, as is an LP basis. Also we expect the algorithm of Sec. 5.1 to produce routes containing fewer paths.

6. TWO EXAMPLES

This section presents the maximum flow-life curve routing for two examples and compares it to maximizing the network life and the minimum total power routing.

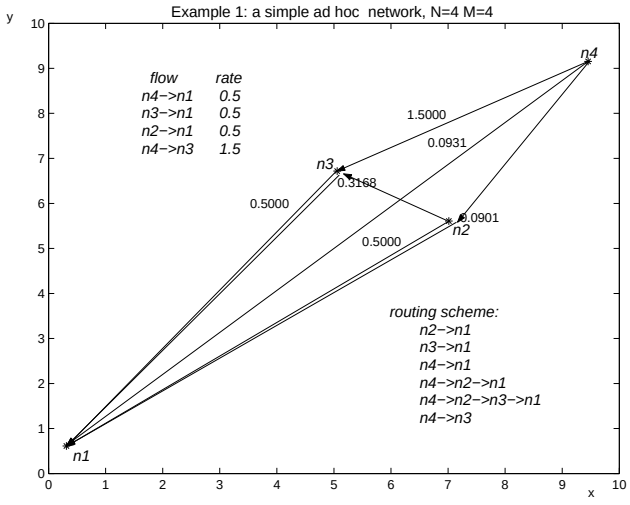

Figure 2: Node placement and flows in a four node network where the labels are the traffic carried on each path under the maximum flow-life curve routing.

6.1 Example 1:

This example evaluates the 4 node network with 4 flows in Figure 2. The nodes are located at

$$\{(0.31, 0.61), (7.01, 5.61), (5.06, 6.72), (9.45, 9.15)\}$$

and have an initial energy of 10000 units. The power costs per unit flow are given by

$$\begin{aligned} g_{ij}^{s} &= 1 + d_{ij}^{4}; \\ g_{ij}^{r} &= 1; \end{aligned}$$

where d_{ij} is the distance between v_i and v_j. The flow utility for a flow is equal to its flow rate. With this set up, we compute the maximum flow-life routing. The resulting flow rate on each path is:

$$\begin{aligned} v_2 \to v_1: &\quad 0.5000; \\ v_3 \to v_1: &\quad 0.5000; \\ v_4 \to v_1: &\quad 0.0931; \\ v_4 \to v_2 \to v_1: &\quad 0.0901; \\ v_4 \to v_2 \to v_3 \to v_1: &\quad 0.3168; \\ v_4 \to v_3: &\quad 1.5000; \end{aligned}$$

The drop points and exhausted nodes are listed in Table 1. For comparison, we compute the minimum total power routing [13]. When a node dies in the minimum total power routing, we recompute the minimum total power routing with the remaining nodes and energy which is used until the next node dies and so on. The result is in Table 2. We see that the maximum flow-life curve keeps the entire network connected for 80% longer than the minimum total power routing, while reducing the time until the last flow is exhausted by only 25%. Interestingly, the total traffic

i	τ_i	S_i^n	S_i^f	$\sum f_{kl}$
0	0	$\{v_1, v_2, v_3, v_4\}$	$\{f_{41}, f_{31}, f_{21}, f_{43}\}$	3.0
1	3.410	$\{v_1\}$	$\emptyset$	0

Table 1: Drop points and surviving nodes with maximum flow-life curve routing (Ex. 1).

i	t_i	S_i^n	S_i^f	$\sum f_{kl}$
0	0	$\{v_1, v_2, v_3, v_4\}$	$\{f_{41}, f_{31}, f_{21}, f_{43}\}$	3.0
1	1.857	$\{v_1, v_2, v_4\}$	$\{f_{41}, f_{21}\}$	1.0
2	3.878	$\{v_1, v_4\}$	$\{f_{41}\}$	0.5
3	4.562	$\{v_1\}$	$\emptyset$	0

Table 2: Drop points and surviving nodes with minimum total power routing (Ex. 1).

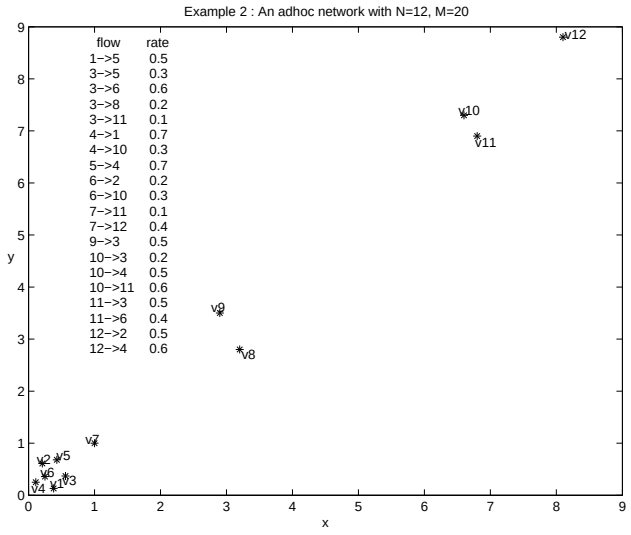

Figure 3: Node placement and flows in a 12 node network.

volume sent with the maximum flow-life routing (10.2) is greater than the total traffic volume sent with the minimum total power routing (7.8). This happens in this case because a relay node (v_3) is also a large source and sink of traffic. Since there is only one drop point, an objective that only maximizes the time to first partition would have yielded a similar routing. The next example shows that this is not always the case.

6.2 Example 2:

This example evaluates the 12 nodes network with 20 flows in Figure 3. The initial energies and link costs are as in the previous example given the node positions in the figure. The flow rates on every path in the maximum flow-life curve routing are:

$$
\begin{aligned}
v_1 \rightarrow v_5&: && 0.5000;\\
v_3 \rightarrow v_5&: && 0.3000;\\
v_3 \rightarrow v_6&: && 0.6000;\\
v_3 \rightarrow v_7 \rightarrow v_8&: && 0.2000;\\
v_3 \rightarrow v_7 \rightarrow v_8 \rightarrow v_9 \rightarrow v_{11}&: && 0.1000;\\
v_4 \rightarrow v_1&: && 0.7000;\\
v_4 \rightarrow v_7 \rightarrow v_8 \rightarrow v_9 \rightarrow v_{10}&: && 0.3000;\\
v_5 \rightarrow v_4&: && 0.7000;\\
v_6 \rightarrow v_2&: && 0.2000;\\
v_6 \rightarrow v_7 \rightarrow v_8 \rightarrow v_9 \rightarrow v_{10}&: && 0.3000;\\
v_7 \rightarrow v_8 \rightarrow v_9 \rightarrow v_{11}&: && 0.1000;\\
v_7 \rightarrow v_{12}&: && 0.0401;\\
v_7 \rightarrow v_1 \rightarrow v_{12}&: && 0.0484;\\
v_7 \rightarrow v_2 \rightarrow v_{12}&: && 0.0540;\\
v_7 \rightarrow v_3 \rightarrow v_{12}&: && 0.0530;\\
v_7 \rightarrow v_4 \rightarrow v_{12}&: && 0.0466;\\
v_7 \rightarrow v_5 \rightarrow v_{12}&: && 0.0560;\\
v_7 \rightarrow v_6 \rightarrow v_{12}&: && 0.0503;\\
v_7 \rightarrow v_8 \rightarrow v_9 \rightarrow v_{12}&: && 0.0517;\\
v_9 \rightarrow v_8 \rightarrow v_7 \rightarrow v_3&: && 0.5000;\\
v_{10} \rightarrow v_9 \rightarrow v_8 \rightarrow v_7 \rightarrow v_3&: && 0.2000;\\
v_{10} \rightarrow v_9 \rightarrow v_8 \rightarrow v_7 \rightarrow v_4&: && 0.5000;\\
v_{10} \rightarrow v_{11}&: && 0.6000;\\
v_{11} \rightarrow v_9 \rightarrow v_8 \rightarrow v_7 \rightarrow v_3&: && 0.5000;\\
v_{11} \rightarrow v_9 \rightarrow v_7 \rightarrow v_6&: && 0.0826;\\
v_{11} \rightarrow v_9 \rightarrow v_8 \rightarrow v_7 \rightarrow v_6&: && 0.3174;\\
v_{12} \rightarrow v_9 \rightarrow v_8 \rightarrow v_7 \rightarrow v_2&: && 0.0568;\\
v_{12} \rightarrow v_{10} \rightarrow v_9 \rightarrow v_7 \rightarrow v_2&: && 0.4432;\\
v_{12} \rightarrow v_9 \rightarrow v_7 \rightarrow v_4&: && 0.2361;\\
v_{12} \rightarrow v_{10} \rightarrow v_{11} \rightarrow v_9 \rightarrow v_7 \rightarrow v_4&: && 0.3639;
\end{aligned}
$$

The drop points and flows exhausted are shown in Table 3. We see that the nodes are exhausted in two steps: First the outer nodes, v_{10}, v_{11}, and v_{12}, then the remaining nodes. Only maximizing the time to the first drop point would have provided a route up to time $\tau_1 = 11.01$, but this is only a fraction of the time to exhaustion for the remaining nodes. We can compare this routing with the minimum total power routing results shown in Table 4. The maximum flow-life curve routing maintains all the flows twice as long as the minimum total power routing. Furthermore, the minimum total power routing is more complex in that all routes are recalculated with each node that is exhausted.

It is interesting to note the role of v_7. Although it is a relay node used in most routes to and from the outer cluster, it offloads its own traffic to each of the nodes in the tight cluster on the left, which in turn forward the traffic to node v_{12}. In this way, most nodes can survive to several hundred time units. In the minimum energy routing, v_7 survives longer but only since the most of the connections to the outlier nodes die earlier.

7. CONCLUSIONS AND FUTURE WORK

We have explored a new routing objective for a power aware ad hoc network, called the maximum flow-life curve. This objective captures the general network goal of maximizing network connectivity over time. We showed several important properties of the maximum flow-life curve: the unique smallest exhausted node set, the sufficient single routing scheme, and the correspondence of the maximum node-life curve routing and maximum flow-life curve routing. These

i	t_i	S_i^n	S_i^f	$\sum f_{kl}$
0	0	$\{v_1, \ldots, v_{12}\}$	$F = \{f_{ij}\}$	8.2
1	11.01	$\{v_1, \ldots, v_9\}$	$\{f_{15}, f_{35}, f_{36}, f_{38}, f_{41}, f_{54}, f_{62}, f_{93}\}$	3.7
2	260.4	$\emptyset$	$\emptyset$	0.0

Table 3: Drop points and surviving flows with the maximum flow-life curve routing (Ex. 2).

i	t_i	S_i^n	S_i^f	$\sum f_{kl}$
0	0	$\{v_1, \ldots, v_{12}\}$	$F = \{f_{ij}\}$	8.2
1	5.149	$\{v_1, \ldots, v_{10}, v_{12}\}$	$\{f_{15}, f_{35}, f_{36}, f_{38}, f_{41}, f_{4,10}, f_{54}, f_{62}, f_{6,10}, f_{7,12}, f_{93}, f_{10,3}, f_{10,4}, f_{12,2}, f_{12,4}\}$	6.5
2	12.07	$\{v_1, \ldots, v_8, v_{12}\}$	$\{f_{15}, f_{35}, f_{36}, f_{38}, f_{41}, f_{54}, f_{62}, f_{7,12}, f_{12,2}, f_{12,4}\}$	4.7
1	14.51	$\{v_1, \ldots, v_8\}$	$\{f_{15}, f_{35}, f_{36}, f_{38}, f_{41}, f_{54}, f_{62}\}$	3.2
3	949.2	$\{v_1, \ldots, v_4, v_6, v_8\}$	$\{f_{36}, f_{38}, f_{41}, f_{62}\}$	1.7
4	1211	$\{v_1, v_2, v_4, v_6, v_8\}$	$\{f_{41}, f_{62}\}$	1.7
5	13,200	$\{v_1, v_2, v_6, v_8\}$	$\{f_{62}\}$	0.2
6	46,130	$\{v_1, v_2, v_8\}$	$\emptyset$	0.0

Table 4: Drop points and surviving flows with the minimum total power routing (Ex. 2).

properties enabled a LP algorithm for computing a maximum flow-life curve routing. Several examples demonstrated the merits of the objective compared to other objectives.

The approach and results in this paper are theoretical in nature. For instance, the routing is centralized and based on having full information about the flows, power costs, and node energy. But, the results provide some practical insights. First, a single static routing is possible as nodes are exhausted. Thus, only the nodes involved in a flow must be notified when the flow is exhausted and not the entire network. Second, the same routing can maximize over time the remaining flow utility and the remaining number of nodes independent of the utility attached to the flow. Thus, nodes can reach consensus on the routing without reaching consensus on the utility of the different flows. Third, the optimal route is not complex with only $N+M$ paths involved. For instance, if $M > N$, most flows will use only one or two paths. Fourth, we saw in Example 1 that, counter to intuition, minimizing the total network power does not maximize the total volume of traffic sent. This suggests that more work needs to be done in this area.

Another direction for future work is to develop a more distributed version of the algorithm. For instance, if the nodes distributed the dual variables Y, then they could use these as an energy-based weighting for choosing routes on existing and new flows.

The paper does not directly consider node mobility, changing flow requirements over time, and quality of service issues such as congestion and delay. Mobility and dynamic flows can be included by periodically updating the routing using the most current information. Congestion can be included by adding constraints that limit the total flow on any link. This potentially could add $N(N-1)$ additional constraints, but these constraints can be added as needed for links that are congested in the unconstrained solution. Delay, hop limits, etc, can be added by removing some path variables from consideration. In some cases this may mean only these prohibited paths are selected to enter in the Dijkstra step of the algorithm. In such cases, Dijkstra can be modified to yield not just the shortest path, but the l shortest paths where l is large enough to decide if there is any allowed path that can enter. These modifications have not been tested and are the subject of future work.

8. ACKNOWLEDGMENTS

Research supported by the NSF under CAREER Award: NCR-9624791, NSF Grant NCR-9725778 and NSF Grant ANI-0082998. Thanks to Bing Li who implemented the algorithm for the examples.

9. REFERENCES

[1] T.X Brown, "Low power wireless communication via reinforcement learning, *Advances in NIPS 12*, S.A. Solla and T.K. Leen and K.-R. Müller eds., MIT Press, 2000, pp. 893-899.

[2] T.X Brown, S. Doshi, and Q. Zhang, "Optimal Power Aware Routing in a Wireless Ad Hoc Network, " *LANMAN 2001: 11th IEEE Workshop on Local and Metropolitan Area Networks,* pp 102-105, March 2000.

[3] J.H. Chang and L. Tassiulas, "Routing for Maximum System Lifetime in Wireless Ad Hoc Networks," *Proceedings of 37th Annual Allerton Conference on Communication, Control, and Computing,* Monticello, IL, September 1999.

[4] J.H. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad Hoc Networks," *Proceedings of IEEE INFOCOM 2000,* pp. 22-31, 2000

[5] V., Chvatal, *Linear Programming,* W.H. Freeman and Co., New York, 1983.

[6] C. Elliot, B. Heile, "Self-Organizing, Self-Healing Wireless Networks," *Proc. IEEE Int. Conf. on Personal Wireless Communication*, pp. 355–362, 2000.

[7] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile computing,* pp. 153-181 1996.

[8] P. Lettieri, C. Schurgers, M. Srivastava, "Adaptive link layer strategies for energy efficient wireless networking," *Wireless Networks*, v. 5, 1999, pp. 339–355.

[9] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM Mobile Networks and Applications Journal, Special Issue on Routing in Mobile Communication Networks,* pp. 183-197, October 1996.

[10] M. Perkins, ed., *Ad Hoc Networking,* Addison Wesley, Upper Saddle River, NJ. 370p.

[11] R. Ramanathan, R. Rosales-Hain, "Topology control of Multihop Wireless Networks using transmit power adjustment," *IEEE INFOCOM 2000* pp. 404–413.

[12] T.S. Rappaport, *Wireless Communications: Principles and Practice,* Prentice-Hall Pub., Englewood Cliffs, NJ, 1996.

[13] V. Rodoplu and T.H. Meng, "Minimum energy mobile wireless networks," *IEEE JSAC,* v. 17, n. 8, pp. 1333-44, Aug. 1999.

[14] A. Schrijver, *Theory of Linear and Integer Programming,* John Wiley, 1986 p.148

[15] Singh, S. and Raghavendra, C.S., "PAMAS - Power aware mulit-access protocol with signaling for ad hoc networks, *ACM Computer Communications Review*, July, 1998.

[16] S. Singh, M. Woo, and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," *Proceedings of ACM/IEEE MOBICOM'98,* October 1998.

[17] K.M. Sivalingam, M.B. Srivastava, P. Agrawal, "Low-power link and access protocols for wireless multimedia networks," *Proc. of IEEE Vehicular Technology Conference*, Phoenix, AZ, May, 1997.

[18] M.W. Subbarao, "Dynamic Power-Conscious Routing for MANETs: An Initial Approach," *Proc. IEEE VTC*, Amsterdam, The Netherlands, Sept. 1999.