Asymptotically Optimal Power-Aware Routing for Multihop Wireless Networks with Renewable Energy Sources

Longbi Lin, Ness B. Shroff, and R. Srikant

***Abstract*— In this paper, we model and characterize the performance of multihop radio networks in the presence of energy constraints, and design routing algorithms to optimally utilize the available energy. The energy model allows vastly different energy sources in heterogeneous environments. The proposed algorithm is shown to achieve a competitive ratio (i.e., the ratio of the performance of any off-line algorithm that has knowledge of all past and future packet arrivals to the performance of our online algorithm) that is asymptotically optimal with respect to the number of nodes in the network. The algorithm assumes no statistical information on packet arrivals and can easily be incorporated into existing routing frameworks (e.g., proactive or on-demand methodologies) in a distributed fashion. Simulation results confirm that the algorithm performs very well in terms of maximizing the throughput of an energy-constrained network. Further, a new threshold-based scheme is proposed to reduce the routing overhead while incurring only minimum performance degradation.**

***Index Terms*— Energy Aware Routing, Competitive Analysis, Mathematical Programming/Optimization, Simulations**

I. Introduction

Ad hoc wireless networks have a broad range of applicability: they can be used to interconnect PCs and laptops in a wireless LAN setting, provide the means of communication between hand-held devices, as well as enable the transmission of events that are observed by sensor network nodes back to collection points or data processing centers. The operational capabilities of such networks are fundamentally limited by the energy available at the nodes (radios) in the network. New and exciting developments in the area of renewable sources of energy [3], [6], [13] can be used to replenish the energy of individual nodes without the need to tether them to an electrical outlet. For instance, self-powered sensors have been developed that rely on harvesting strain and vibration energies from their working environment [13]. However, the energy management of these networks is still very important since replenishment rates are typically small, and the available energy is still a bottleneck in being able to successfully transmit packets through the network. The introduction of renewable energy sources poses new problems in the energy management of these ad hoc networks. Among other possible techniques for energy conservation, power-aware routing is aimed at choosing the most energy-efficient route to forward the packets, at the cost of computational overhead. In this paper, we present an admission control/routing framework in which we formulate and solve the problem of power-aware routing with energy replenishment.

Energy-aware routing has received significant attention over the past few years [8], [9], [14], [15], [16], [17]. In [9], [16], algorithms have been presented to optimize the lifetime of the network. These algorithms can be viewed as different attempts to combine the key elements of two basic routing approaches: Minimum Energy (ME) routing that selects the route with least total link energy cost, and Max-min routing that selects the route with maximum bottleneck residual node energy. It is shown through simulations in [9] that the algorithm empirically achieves a good competitive ratio (The idea of competitive ratio will be explained in details later). In [17], the authors describe a way to incorporate a simple measure of a node's residual energy into the node's cost function in solving the problem of routing multicast circuits in an energy-limited wireless network. The authors realize the potential of developing alternative cost functions and make no claim of optimality. In [7], [10], energy-aware routing algorithms are presented, where the cost metric is defined as an exponential function in residual energy. The main contribution in these works is the analogy between the energy-aware routing problem and the routing of permanent virtual circuits (PVCs) as in [12], and the verification that shows the mapping from per-link resources to per-node resources does not change the nature of the problem.

L. Lin and N.B. Shroff are with the School of ECE Purdue University (email: llin@purdue.edu, shroff@ecn.purdue.edu).

R. Srikant is with the EE Department, University of Illinois at Urbana-Champaign, (email: rsrikant@uiuc.edu).

The intellectual merit of our work lies in the development of

- a mathematical framework that takes into account practical realities such as energy replenishment, mobility, and erroneous routing information,
- associated analytical techniques to provide an understanding of the performance benefits that can be achieved through power-aware routing, and
- distributed and scalable routing solutions that can be tailored to a variety of network topologies, traffic and mobility patterns.

Our energy model only assumes that each node in the network knows its own short-term energy replenishment schedule. (This will be explained in detail in the next section.) The energy flow into each node can be, for example, of different rates, or according to different on-off processes. The model also captures heterogeneous energy source distributions in the network, and our algorithm in fact uses the heterogeneity to do admission control and routing in an energy-opportunistic way. This algorithm is developed by making connections to routing of permanent virtual circuits (PVC) and switched virtual circuits (SVC) in the ATM (Asynchronous Transfer Mode) literature. This is an online algorithm that can be easily implemented in a distributed fashion. By "online" we mean the algorithm does not know future packet routing requests at decision time. In contrast, an offline algorithm knows the arrival times and packet sizes of all the packet routing requests, including those in the future. We show that our algorithm asymptotically achieves the best achievable performance by any online algorithm.

The rest of this paper is organized as follows: in Section II, we formulate the problem of energy-aware routing with energy replenishment, and present our network and energy model. In Section III and Section IV, we present our algorithm and briefly discuss its implications. In Section V, we discuss our main result on the competitive ratio of our algorithm. Numerical results are provided in Section VI. A threshold-based scheme to reduce routing overhead is presented in Section VII, and the integration of our algorithm and the DSR-like on-demand routing framework is discussed in Section VIII. Concluding remarks are presented in Section IX.

II. Problem Formulation

A wireless multi-hop network is described by a directed graph $G(V,E)$, where V is the set of vertices representing the sensor nodes, and E is the set of edges representing the communication links between them. Packets are sent in a multi-hop fashion: a path from source to destination consists of one or multiple edges.

A 2-tuple (t_{nm}, r_{nm}) is associated with each edge $(n,m) \in E$, where t_{nm} is the transmission energy requirement for node n and r_{nm} is the reception energy requirement for node m. More precisely, if a data packet of length l is sent from node n to node m directly, an amount of energy equal to lt_{nm} will be subtracted from the residual energy of node n, and lr_{nm} will be subtracted from the residual energy of node m. We assume that the size of a control packet is negligible compared to the size of a data packet.

We define the unit energy requirement of node n on path R as

$$e_n(R) = r_{n''n} + t_{nn'}, \forall n \in R,$$

where nodes n'' and n' are the upstream and downstream neighbors of node n in path R, respectively. For convenience, let $r_{n''n} = 0$ for source node and let $t_{nn'} = 0$ for destination node.

Often, it is assumed that $r_{mn} = 0$. Clearly, this is a special case of our model. However, studies on short-range communication with low radiation power show that the transmission and reception energy costs could nearly be the same [1]. We can incorporate this in our model by letting $t_{nm} = r_{nm}$. Alternatively, in [8], the reception energy cost is captured by adding a constant to the link cost at each hop. This is a special case of our model with $r_{nm} =$ constant.

We consider a discrete-time system in which each sensor node begins with a fully charged battery that has a capacity of u_n. At the end of each time slot τ, $P_n(\tau)$ is the residual energy at node n. At the beginning of time slot τ, node n receives the energy replenishment accumulated in the previous time slot, represented by $\gamma(\tau - 1)$. At all times, the maximum energy at node n is not allowed to exceed u_n.

Data packet routing requests arrive to the network sequentially, the j^{th} of which can be described as:

$$\beta(j) = (S(j), D(j), l(j), T^s(j), \rho(j)), \qquad (1)$$

where $S(j)$ is the source node of the j^{th} packet routing request, $D(j)$ is the destination, $l(j)$ is the packet length, $T^s(j)$ is the arrival time of the request, and finally $\rho(j)$ is the revenue gained by routing this packet through the sensor network. A request can be accepted only if there is at least one feasible path (that is, each node n along the path must have at least $l(j)e_n(R(j))$ amount of residual energy) in the system when the request arrives. If the routing request is accepted and $R(j)$ is the route used to accommodate the request, then $l(j)e_n(R(j))$ will be the amount of energy expenditure at node n, $n \in R(j)$. We also assume that the reduction of energy is instantaneous for all the nodes along the path, since the rate of energy replenishment is usually much slower than the energy consumed by transmitting a packet.

The energy model at node n can therefore be summarized by the following equation:

$$\begin{aligned} P_n(\tau) &= \min(P_n(\tau-1)+\gamma_n(\tau-1), u_n) \\ &\quad -I(a_n(j))l(j)e_n(R(j)), \end{aligned} \quad (2)$$

where $I(\cdot)$ is the indicator function and $a_n(j)$ is the event that $\beta(j)$ is accepted at τ, and $n \in R(j)$.

It is assumed that each node has an accurate estimate of its own short-term energy replenishment schedule. More precisely, at time slot τ, node n knows $\gamma_n(\tau), \gamma_n(\tau+1), \ldots, \gamma_n(\hat{\tau}_n)$, where $\hat{\tau}_n$ is the earliest time the battery at node n would be fully recharged if no request were accepted at or after time τ. It is worth noting that the $\hat{\tau}_n$ here is dependent on the residual energy of node n at the arrival time of a request. In practice, this type of short-term prediction can be easily implemented.

Our goal is to maximize the total *revenue* over some finite horizon $[0, t]$;

$$J_t := \sum_{j:\ j \le k(t)} \rho(j) I(a(j)), \quad (3)$$

where $a(j)$ is the event that $\beta(j)$ is accepted, and $k(t)$ is the index of the last arrival in the time horizon, or equivalently, $k(t)$ is the total number of arrivals in the time interval $[0, t]$.

We briefly comment on the choice of the revenue of the j^{th} packet, $\rho(j)$, in the above formulation:

- If $\rho(j) \equiv 1$, then J_t is simply the total throughput in $[0, t]$.
- If different packets have different priorities, then this can be reflected in the above formulation by choosing different values of $\rho(\cdot)$ for different packets. A larger value of ρ would then indicate a packet of high priority.
- Since the work of Gupta and Kumar [4], a new metric (bit-meters per sec.) that combines throughput as well as the distance traversed by a bit has become popular. This can also be incorporated in our model by simply choosing $\rho(j)$ to be proportional to the distance between $S(j)$ and $D(j)$.

III. Algorithm for the Case of Constant Replenishment Rate

To succinctly highlight the main attributes of our solution, in this section, we describe our algorithm for the case when the rate of energy replenishment is constant (in time) at each node (although different nodes can have different replenishment rates). The solution to the more general case will be presented in the next section.

The basic idea of our algorithm is to assign a cost to each node, which is an exponential function in its residual energy and then use shortest-path routing with respect to this metric. To account for the timing relationship between the energy consumption and replenishment, we also need to measure the effect of previously accepted requests. To this end, we define the power depletion index $\lambda_n(j)$ as

$$\lambda_n(j) = \frac{u_n - P'_n(j)}{u_n}, \quad (4)$$

where $P'_n(j)$ is the energy at node n *right before considering request* j. We will show in Sections IV and V that the appropriate cost metric C_n associated with each node is given by:

$$C_n(j, R) = \frac{u_n}{(\gamma_n + \epsilon)\log\mu}(\mu^{\lambda_n(j)} - 1) l(j) e_n(R(j)), \quad (5)$$

where we recall that R is a path from source to destination, u_n is the battery capacity of node n, γ_n is the rate of energy replenishment, $\lambda_n(j)$ is the fractional energy used up at node n when considering request j, $l(j)e_n(R(j))$ is the energy requirement for packet j of length $l(j)$, and μ, ϵ are appropriately chosen constants.

As in a typical weighted shortest path routing, the cost associated with R when considering request $\beta(j)$ will therefore be calculated as:

$$Cost_R(j) = \sum_{n \in R} C_n(j, R)$$

Our proposed algorithm can be described as follow:

E-WME (Energy-opportunistic Weighted Minimum Energy) Algorithm

For an incoming routing request j, check if the least-cost route R from $S(j)$ to $D(j)$ satisfies

$$Cost_R(j) \le \rho(j). \quad (6)$$

If yes, accept the request and route the packet on the least-cost route.

Otherwise reject the request.

Remark: The E-WME algorithm presented here has provably good performance (because the cost function has been appropriately chosen) in the sense that it can secure a relatively large amount of revenue without any statistical information about the routing requests. This point will be further discussed when presenting our main result using competitive analysis in Section V. Moreover, this algorithm requires only local information at each node and can be easily incorporated in traditional distance-vector type of routing framework in a distributed fashion. For DSR-like mobile ad hoc on-demand routing protocols, we have also designed a distributed algorithm using our proposed metric to render them energy-aware. This will be further discussed in Section VIII.

Before delving into our results from competitive analysis, it is more interesting to first look at the cost metric defined in (5), and to understand intuitively first why this algorithm has good performance:

1) Note that the metric in our E-WME algorithm for each node is an exponential function of the nodal residual energy, a linear function of the transmit and receive energies, and an inversely linear function of the replenishment rate. So E-WME provides us with a clear guideline of how to balance the importance of residual energy (related to load balancing), the transmit and receive energies (related to resource thriftiness), and the quality of the replenishment.
2) If we assume that the nodes have the same energy replenishment process, e.g., all have the same constant rate of replenishment, the cost function (5) can be viewed as combining elements of the so-called Minimum Energy (ME) and Max-Min approaches, similar to ideas in [7], [10]. Suppose that there are two parallel links whose transmission and reception nodes have the same residual energies, then the one with the smaller link energy cost will be selected. Thus, it resembles the ME algorithm in this case. On the other hand, if there is a choice between two nodes whose link energy costs are the same, the algorithm will choose the node with the larger residual power. This behavior is similar to the Max-min approach.
3) In an environment where the energy source distribution is heterogeneous, by using the cost function (5), the network automatically directs traffic to nodes with a faster energy renewal rate. Consider a set of nodes with similar residual energy as well as similar link transmission and reception energy requirements. Of these nodes, the ones which can replenish their batteries at a higher rate will advertise a cheaper cost. For instance, in a sensor network powered by solar cells, nodes receiving more sunlight will forward more data packets.
4) Note that even though u_n is in the numerator in (5), it does not imply that nodes with larger battery capacity are assigned a higher cost. The reason is that u_n is also embedded in the exponential cost metric since $\lambda_n(j) = 1 - \frac{P'_n(j)}{u_n}$, where $P'_n(j)$ is the residual energy at node n when considering request j.

IV. E-WME Algorithm for the General Case

In this section, we present the E-WME algorithm that allows a time-varying replenishment rate at each node.

We begin by defining a set of parameters to describe the effect of previously accepted routing requests when considering the new request $\beta(j)$. More specifically, let $\Delta t_n(j)$ be the amount of time it takes for the incoming energy, accumulated from time slot $T^s(j-1)$, to equal $u_n - P_n(T^s(j-1))$. As mentioned in Section II, we then define

$$\hat{\tau}_n(j) = T^s(j-1) + \Delta t_n(j).$$

$\hat{\tau}_n(j)$ is the earliest time the battery at node n would be fully recharged if no request were accepted after the $(j-1)^{\text{th}}$ request. It can also be written as:

$$\hat{\tau}_n(j) = \min_{\tau \geq T^s(j-1)} \Big[\sum_{t=T^s(j-1)}^{\tau-1} \gamma_n(t) \geq (u_n - P_n(T^s(j-1))) \Big].$$

In reality, it is conceivable that only a fraction of the last replenishment is received by the node, due to limited battery capacity. For ease of exposition, we ignore the fraction of energy replenishment lost in this scenario. We also assume that $\hat{\tau}_n$ is finite.

To characterize the energy consumption due to previous packets, we define the new power depletion index $\lambda_n(j,\tau)$ as

$$\lambda_n(j,\tau) = \begin{cases} 0, & \tau \geq \hat{\tau}_n(j), \\ \lambda_n(k_\tau, \tau), & \tau < T^s(j-1), \\ \frac{u_n - P_n(T^s(j-1)) - \sum_{t=T^s(j-1)}^{\tau-1} \gamma_n(t)}{u_n}, & \text{otherwise}, \end{cases}$$

where

$$k_\tau = \max[j : T^s(j-1) \leq \tau].$$

In fact, $\lambda_n(j,\tau)$ is the fraction of energy consumption due to $\{\beta(1), \beta(2), \ldots, \beta(j-1)\}$ for node n, as measured at time τ. Note that new routing requests (with index greater than $(j-1)$) can arrive at or before time τ, but their energy consumption will *not* be included in the calculation of $\lambda_n(j,\tau)$. There are three cases in the above definition:

- $\tau \geq \hat{\tau}_n(j)$: By the definition of $\hat{\tau}_n(j)$, $\lambda_n(j,\tau)$ should be zero at or after time $\hat{\tau}_n(j)$.
- $T^s(j-1) \leq \tau < \hat{\tau}_n(j)$: In this case, part of the energy consumption has been restored.

- $\tau < T^s(j-1)$: In this case, time slot τ is before the arrival time of $(j-1)^{\text{th}}$ request, hence, it is almost meaningless to talk about the energy consumption of $\{\beta(1), \beta(2), \ldots, \beta(j-1)\}$ at time τ. For preciseness, we define $\lambda_n(j,\tau)$ in this case to be $\lambda_n(k_\tau, \tau)$, where k_τ is the largest request index j such that $\lambda_n(j,\tau)$ is "meaningful".

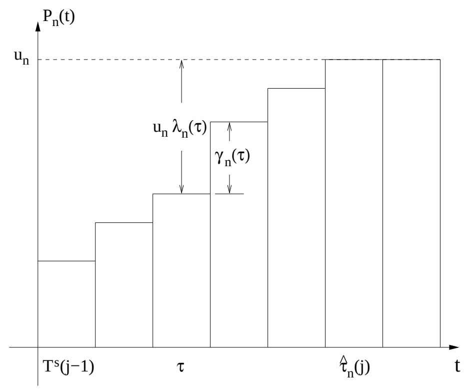

Fig. 1. The Amount of Energy at Node n Assuming No Request is Accepted after the $(j-1)^{\text{th}}$ Request

Figure 1 shows the amount of energy at node n assuming no request is accepted after the $(j-1)^{\text{th}}$ request.

We now define our routing metric used on each node as:

$$C_n(j,R) = \sum_{\tau=T^s(j)}^{\hat{\tau}_n(j)-1} l(j)e_n(R)(\mu^{\lambda_n(j,\tau)}-1), \quad (7)$$

where μ is a constant to be defined later, and R is a path from $S(j)$ to $D(j)$. Again, the cost associated with R when considering request $\beta(j)$ will therefore be calculated as:

$$Cost_R(j) = \sum_{n \in R} C_n(j,R)$$

The admission control part of our E-WME algorithm for time-varying replenishment rate is the same as the one in Section III.

Remarks: It is worth noting that the admission control of routing requests is done in an energy-opportunistic fashion. Again we turn to the example of a sensor network powered by solar cells. Let us assume that a request arrives at the network right after sunset. Recall our assumption that each node knows its short-term energy replenishment schedule. At this moment, each node knows that the energy replenishment rate will be much smaller for the several hours to come. (In practice, this type of knowledge can be gained by evaluating the energy replenishment schedule over the past few days.) The $\hat{\tau}_n(\cdot)$ calculated will then be relatively large, so the cost of routing the packet will be higher than that during the daytime. As compared to its daytime policy, the network is thus more conservative in accepting the request, which is precisely what the network should do in this particular scenario.

The cost function (7) is more complicated than that of the constant rate case (5). Nevertheless, it corresponds to a simple sum that can easily be computed at each node. Further, from an intuitive point of view, it still carries all the merits we have discussed in Section III.

Note that the cost function (5) for the case when the rate of energy replenishment is constant in time, i.e., $\gamma_n(\tau) \equiv \gamma_n$, can be approximated directly from the more general cost metric (7) when the node energy level is not full or close to full.

Consider the case where $\gamma_n \neq 0$. Since node n does not have a full battery upon the arrival of $\beta(j)$, and by the definition of $\hat{\tau}_n$, we have

$$u_n - P_n'(j) = (\hat{\tau}_n(j) - T^s(j))\gamma_n, \quad (8)$$

and

$$\lambda_n(j,\tau) = \frac{u_n - P_n'(j) - (\tau - T^s(j))\gamma_n}{u_n}, \quad (9)$$

where we recall that $P_n'(j)$ is the node energy right before considering request j. Putting (8) and (9) into (7) gives

$$\begin{aligned} C_n(j,R) &= \sum_{\tau=T^s(j)}^{\hat{\tau}_n(j)-1} l(j)e_n(R)(\mu^{\lambda_n(j,\tau)}-1) \\ &= l(j)e_n(R)\left(\frac{\mu^{\frac{u_n-P_n'(j)}{u_n}}-1}{1-\mu^{-\frac{\gamma_n}{u_n}}} - \frac{u_n-P_n'(j)}{\gamma_n}\right). \end{aligned}$$

Note that $\mu^{-x} \approx 1 - x\log\mu$, when $0 \leq x \ll 1$. Using this approximation and (4), the cost function can be further simplified:

$$\begin{aligned} C_n(j,R) &\approx \frac{l(j)e_n(R)u_n}{\gamma_n \log\mu}\left(\mu^{\lambda_n(j)} - 1 - \lambda_n(j)\log\mu\right) \quad (10) \\ &\approx \frac{l(j)e_n(R)u_n}{\gamma_n \log\mu}\left(\mu^{\lambda_n(j)} - 1\right). \quad (11) \end{aligned}$$

The last approximation is true since $\lambda_n(j)$ is not close to zero and $\mu \gg 1$.

In the case when the node energy is full or close to full, (10) can be used instead of (11).

Note that (11) is slightly different from the cost function (5) we proposed in Section III. The ϵ term in (5) is to handle the case of nodes without energy replenishment, i.e., $\gamma_n = 0$. From the theoretical point of view, (11) assigns a cost metric ∞ to nodes with $\gamma_n = 0$. Clearly this leads to over-conservative admission control for routing requests that can potentially utilize those nodes, and does not provide enough resolution to distinguish between two nodes without energy replenishment. Our earlier work [10] suggests that a good cost metric for nodes without renewable energy sources is a function that is exponential in the residual energy, and linear in transmission/reception energy requirement. Therefore, in a system where nodes with and without energy replenishment are both present, the afore-mentioned problems can be alleviated by adding an appropriately chosen small positive ϵ to the rate γ_n in the cost function.

V. Asymptotic Optimality of the E-WME Algorithm

The algorithms presented in Section III and Section IV are online algorithms with asymptotically optimal competitive ratio. The competitive ratio is defined as

$$\sup_{t} \sup_{\substack{\text{all input} \\ \text{sequences in } [0,t]}} \frac{J_{\text{t,off}}}{J_{\text{t,on}}},$$

where $J_{\text{t,off}}$ is the performance achievable by any offline algorithm and $J_{\text{t,on}}$ is the performance of the given online algorithm, where the performance is defined in equation (3). A competitive ratio of r means that the performance of the online algorithm is at least $1/r$ that of any offline algorithm. In other words, a smaller competitive ratio means higher performance.

We need the following two assumptions:

(A1) $1 \le \frac{1}{L} \cdot \frac{\rho(j)}{l(j)e_n(R(j))T} \le F, \qquad n \in R(j)$, and

(A2) $l(j)e_n(R(j)) \le \frac{u_n}{\log \mu}, \qquad n \in R(j)$,

where $R(j)$ is the path chosen by either the online or the offline algorithm to route $\beta(j)$, L is the maximum hop count allowed for any path, F is a constant chosen large enough to satisfy (A1), T is the upper bound on the time it takes to fully charge the empty battery at any given node, and $\mu = 2(LFT + 1)$. Assumption (A1) requires that the revenue from a packet scales with the amount of resource it requests. This is quite reasonable and certainly agrees with the definition of revenue as throughput or weighted throughput. Assumption (A2) guarantees that the energy claimed by a packet is not larger than a certain fraction of the total energy available at any single node. These assumptions are modifications of the assumptions in [12] and take into account some crucial differences that we will discuss shortly.

Under assumptions (A1) and (A2), we have the following theorem. We prove this theorem for the E-WME algorithm in the general case using the cost function given by (7). A similar result can be proven using the cost function (5) for the special case with constant energy replenishment.

Theorem 1: **(Asymptotic Optimality of the E-WME Algorithm)**

(A) The E-WME algorithm has a competitive ratio upper bounded by $O(\log(|V|))$, where $|V|$ is the number of nodes in the network.

(B) The competitive ratio of any online routing scheme is lower bounded by $\Omega(\log |V|)$.

From (A) and (B), our algorithm is asymptotically optimal.

Proof of Theorem 1: Please refer to the Appendix for the proof.

Compared to the SVC routing in the ATM literature presented in [12], our routing problem has several crucial differences that are worth pointing.

1) The replenishment of energy, or the release of resources in our case is a per-node activity, while it is per-request in the routing SVC case. This is the single most important difference that makes it impossible to solve our problem by trivially applying the SVC routing algorithm.
2) The release of resources in our system is a replenishment *process*, while the bandwidth occupied by a SVC is released at the end of the connection.
3) The SVC case is a typical loss system where there are multiple servers with no waiting room. In our system, each node can be viewed as an energy queue where the workload is the energy to replenish and the battery is the buffer. As a result, the limits of the summation over time in the cost metric equation (7) actually depends on the residual energy seen by the incoming request. In the SVC case, the summation over time depends only on the holding time of the incoming request itself.

In the appendix, we prove the main result using techniques developed for the SVC case while taking into account the crucial differences between the two scenarios described above.

VI. Numerical Results

We now describe the results from our simulations. For our simulation, we randomly deploy 200 nodes on a 10×10 field. All nodes have an initial power of 1. The power consumption to send a unit packet directly is $10^{-4}d_{nm}^3$, where d_{nm} is the distance between two nodes. There is a link between node n and m if and only if (a) the distance between them is less than or equal to the maximum transmission range of a node and (b) node n has enough power to transmit a packet from n to m directly. The maximum transmission range[1] is 3. Packet lengths are all 100. For each routing request, the source and destination pair is randomly selected among all the nodes (we have obtained similar results when packets are directed to a single node, such as data collection center in a sensor network. See Section VII for such an example). Each node is responsible for generating its own packets as well as forwarding packets for others. Energy replenishment processes at the nodes are assumed to be *i.i.d.* random processes. At each time slot, the amount of energy that a node receives is uniformly distributed over a certain interval. The average replenishment rate of half of the nodes is 3 times greater than the other half.

Even though our algorithm attempts to maximize the revenue of the network, to illustrate that the algorithm also has good performance under other metrics, we also use the oft-used notion of lifetime to compare our algorithm with other algorithms. Specifically, we say that a *partition* has occurred for a node pair if there is no path between the nodes with sufficient energy to route a packet. The above definition leaves the definition of network lifetime "open." Lifetime could then be defined as the time that it takes for a certain fraction of the node pairs to experience partition. We believe a good definition of network lifetime is strongly application-dependent. Some applications may require that all nodes stay connected at any given time, as in the traditional ad hoc computer networks. In that case, the throughput until the first node down time will be a good candidate for the network lifetime. In the case where nodes are densely deployed, losing connectivity at a few nodes may not pose great danger to the health of the network. To take into the many possible definitions of lifetime, we plot the end-to-end throughput against the number of node pairs that have experienced partition. For example, a point $(500, 5000)$ would mean that 5000 packets were delivered between the randomly chosen source-destination pairs by the time 500 packets were dropped by the network because there was not sufficient energy to transmit the packets. Of course, since we allow energy replenishment, a partitioned node-pair could regain their connectivity later on.

In our simulation, we do not allow rejection of packets. In fact, if we were to allow rejection, we would have to find a good, if not the best, admission control algorithm for each of the routing algorithm we compare the E-WME algorithm to. Therefore it would be harder to justify a fair comparison of the routing algorithms. Hence, for the numerical results shown here, we have decided not to allow rejection.

[1]The selection of maximum transmission range by itself is an interesting problem. Here we just choose a value so that any two nodes are initially connected to each other in a multi-hop fashion.

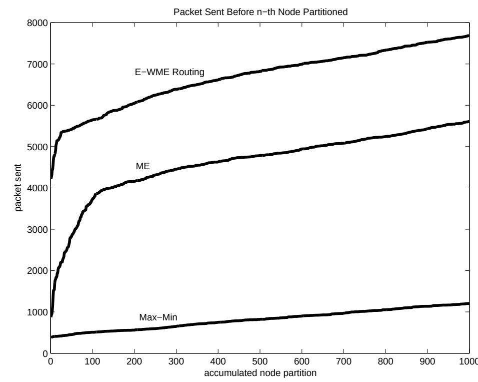

Fig. 2. Throughput Comparison of our Proposed Algorithm to ME and Maxmin Algorithms

Figure 2 shows the throughput comparison between our algorithm and the two representative routing algorithms in the literature: minimum energy routing and max-min routing [1] (See Section I for a brief description of these two algorithms). It can be seen that the E-WME routing always has better throughput than minimum energy routing, and much better throughput than the Max-min routing. Similarly, we have conducted other simulations that show that E-WME outperforms some other commonly used power aware routing algorithms in the literature. The two main reasons are that E-WME is optimal in the sense of minimizing the competitive ratio, and is also energy opportunistic, characteristics that are not present in other power aware routing algorithms.

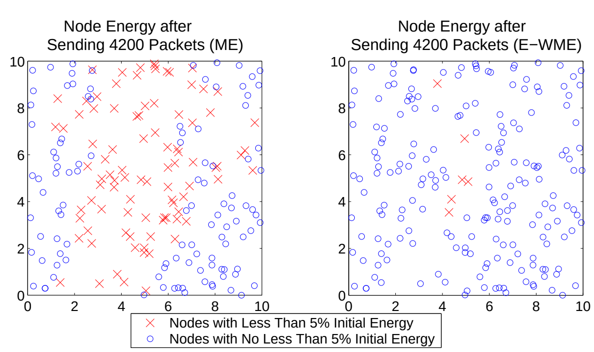

Fig. 3. Node Energy Distribution after Sending 4200 packets

Figure 3 depicts the node energy distribution after 4200 successful end-to-end packet deliveries. This corresponds approximately to the time-instance at which the first node partitions take place in E-WME (see Figure 2). It is clear that the network with the ME algorithm has many more nodes with low energy levels (the "crosses" corresponds to nodes with less than 5% of their battery capacity, while "circles" correspond to nodes with greater than 5% of their battery capacity). The reason for the poor performance of the ME algorithm is its failure to load-balance between the nodes.

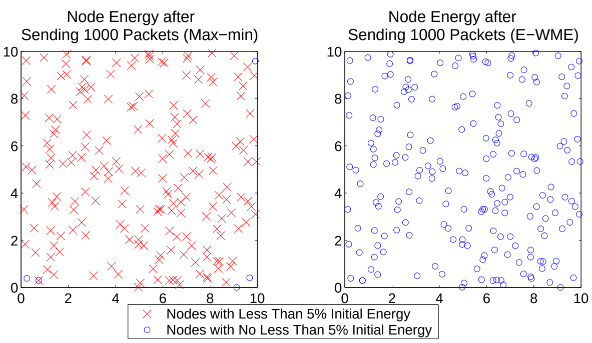

Fig. 4. Node Energy Distribution after Sending 1000 packets

Figure 4 depicts the node energy distribution after 1000 successful end-to-end packet deliveries. It is clear that the network with Max-min algorithm has many more nodes with low energy levels. If this is a sensor network, coverage and connectivity are greatly impaired. The reason for the poor performance of the Max-min algorithm is its failure to consider transmit and receive energies, which leads to routes with only a few hops and very large average energy expenditure per packet.

VII. Reducing Routing Overhead

The proposed algorithm relies on instantaneous nodal information, so changes in the energy level at each node have to be instantaneously communicated to other nodes. In practice, this load balancing need not be carried out frequently. Our approach is as follows: routing updates are only initiated when the residual energy at a node passes some preset threshold. Intuitively, thresholds should be more finely tuned in nodes that are closer to energy depletion (so that these nodes can be avoided, if possible). Towards this end, we define a set of thresholds

$$T(i) = \log_\mu \frac{M}{i} \qquad i = 1, 2, \ldots, M.$$

After forwarding a packet, each node will check its fractional residual power. If one or more thresholds is crossed, the node will then initiate an updated.

This threshold-based scheme chooses the right moment to initiate re-routing. It is clear that an error term will appear in the cost metric since we are not using the most up-to-date node energy information. However, it can be easily shown that, as long as no threshold has been passed since the last update, the error in the per-node cost is upper bounded by $C\mu/M$, where C is a constant with respect to node energy. A set of equally spaced thresholds, on the other hand, will have a much larger error when the node residual energy is low.

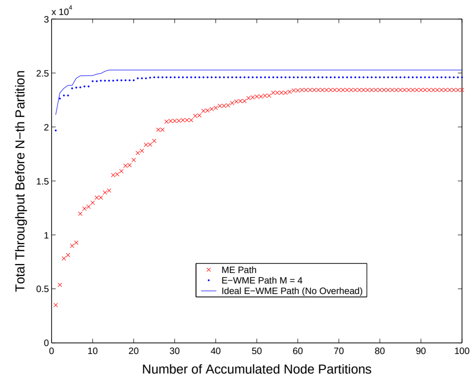

Fig. 5. Throughput Comparison of Minimum Energy Routing and E-WME with different amounts of overhead

Our routing algorithm can therefore adapt to different traffic patterns. A heavily loaded network does more re-routing. In a network where different types of low-duty cycle traffic patterns are possible, this routing algorithm self tunes accordingly. This can result in order of magnitude reduction in routing overhead, and incur only minimal degradation in performance.

Simulation results in Figure 5 show that a four-threshold scheme has performance close to that of the ideal algorithm. (In the ideal case re-routing is done upon every change in nodal energy and the energy for exchanging updates is totally omitted.) It is assumed in this set of simulation that the energy at each node is non-renewable. For each routing request, the source node is randomly chosen and the destination is a common data collection gateway at the center of the field.

When nodes are mobile, routing overhead can be further reduced by using an on-demand routing scheme. We will discuss this in the following section.

VIII. Routing with Mobility

When ad hoc network nodes are highly mobile, a proactive distance vector implementation could lead to a large amount of overhead. In a mobile environment, on demand routing protocols, e.g., Dynamic Source Routing (DSR) [5], have the potential of reducing routing overhead, since there is no need to constantly update the routing tables. Ideally, routing should be tailored for different degrees of mobility. Proactive routing should be used in a low mobility environment, while on-demand routing should be used in a high mobility environment. Given our proposed dynamic routing framework, an interesting question is the following: can we design a distributed algorithm to integrate E-WME into an on-demand routing framework?

The difficulty lies primarily in the route discovery process. Here, we would like to use the E-WME routing metric, and at the same time incur only a minimum amount of routing overhead. We propose the following approach to translate the E-WME cost metric linearly to waiting time, and forward only the best metric based on ideas in [9].

The algorithm for the route discovery process is given as follows. For simplicity of presentation, we assume that the intermediate nodes do not know a path to the destination. Let M_n be the E-WME metric associated with the best path (currently known to node n) from the source to node n, and M_{packet} be associated with a route request packet, representing the E-WME metric of the best path discovered so far. Let δ be an appropriately chosen small positive constant.

Energy-Aware Route Discovery Algorithm

1) Each node n calculates the E-WME cost metric C_{mn} on each of its incoming links from local communication. M_n is initialized to be ∞ for all nodes.
2) The source node S initiates the route discovery by broadcasting a route request packet with source identification S, destination identification D, a unique request packet identification i, cost metric $M_{\text{packet}} = 0$ and the time-stamp T_0.
3) For any other node n,
   a) upon receiving a route request packet, update M_n by
   $$M_n = \min(M_n, C_{mn} + M_{\text{packet}}).$$
   b) If M_n is updated with a smaller value by performing 3a, compute the delay T_w as
   $$T_w = \delta M_n,$$
   and set the timer to expire at time $(T_0 + T_w)$. Cancel any timer that was set to expire after $(T_0 + T_w)$, and is associated with the route discovery initiated by node S with the same request id.
   c) Upon the expiration of the timer, if node n is the destination D, it transmits a route reply packet back to the source using the reverse route. Otherwise, node n propagates the routing request by setting $M_{\text{packet}} = M_n$ and appending its own ID to the source route list in the packet. In both cases, node n ignores any further route request packet initiated by node S with the same request id.

Note that the cost function (7) is path-dependent: the same node can have different costs depending on its upstream and downstream neighbors in the path. In the route discovery process of DSR, when an intermediate node propagates the route request packet, there can be more than one possible next hop, since the path is not finalized yet. Because the E-WME metric dictates the delay at each node in the above algorithm, it is then impossible to have a single delay value for each node. To solve this problem, in our algorithm, we calculate the cost of a *link* (n, m) as follows:

$$C_{nm}(j) = \frac{t_{nm}}{e_n(j)} C_n(j, R) + \frac{r_{nm}}{e_m(j)} C_m(j, R),$$

where we recall that t_{nm} and r_{nm} are the transmission and reception energy requirement of a unit packet for node n and node m, respectively. Note that the link cost $C_{nm}(j)$ calculated this way is independent of the path that the link is in.

The following theorem shows the above algorithm finds the correct E-WME path with little communication overhead.

Theorem 2: **(Validity of the Energy-Aware Route Discovery Algorithm)**

The energy-aware route discovery algorithm finds the shortest path with respect to the E-WME metric with no more than $|V|$ transmissions in total.

Proof of Theorem 2: For part 3c of the algorithm, it is clear that each node in the network transmit at most one route request packet for each round of route discovery process. Therefore no more than $|V|$ route request packets are transmitted in total.

To prove that the algorithm finds the correct shortest path, we show that each node gets the shortest path from source node S to itself when its timer expires. (Note that all the "shortest" paths mentioned in this proof are with respect to the E-WME metric.) Let h be the hop count of the shortest path from the source node S to node n. We prove this result by induction over h as follows:

1) If $h = 1$, from part 2 in the algorithm, it is clear that node n gets the shortest path from source node S to itself when its timer expires.
2) Let us assume that any node with $h = k$ gets the shortest path from source node S to itself when its timer expires. For any node n with $h = k + 1$,

the shortest path R_{Sn} from S to n consists of the short path R_{Sm} from S to a node m and the link (m, n), where node m is a neighbor of node n. Since R_{Sm} is a path with k hops, node m sends the correct routing update to node n when the timer of node m expires. Since R_{Sn} is the "shortest" path, the delay calculated for this path is the smallest. This has two implications: on the one hand, no timer of node n can expire before this timer, which implies the update from node m is not ignored; on the other hand, the update sent out when this timer expires carries the path R_{Sn} as well as the correct E-WME metric. ■

In the above theorem, the delay due to queueing and MAC contention, etc., is ignored. By choosing a large enough δ, we can ensure that the algorithm is still correct in the presence of such delay [9]. Using the above approach, we can reduce the overhead at the cost of a larger delay in route discovery. Meanwhile, other features of on-demand routing, e.g., detecting a change in network topology, time to keep routes in cache, etc., are not affected.

IX. Conclusions

In this work, we address the problem of power-aware routing with distributed energy replenishment. We formulate the problem as an integrated admission control and routing framework by appealing to ideas from PVC/SVC routing in the ATM literature. The energy model in this framework allows vastly different energy sources in heterogeneous environments. We have shown that our E-WME algorithm has an asymptotically optimal competitive ratio, which suggests that in practice this algorithm can lead to significant improvements in the performance of the network. The algorithm is also easy to implement: it requires local short-term energy replenishment information and assumes no knowledge about the statistical information on the packet arrivals. The algorithm can be seamlessly integrated with distance-vector-like proactive routing protocols, and with minor modifications, can also be integrated with on-demand routing protocols. A threshold-based scheme is also introduced to reduce routing overhead while incurring minimum performance degradation.

Some possible future directions are as follows. Routing with a certain amount of known statistical information is one of the more interesting directions. The strength of the competitive analysis lies in the fact that no statistical information on packet arrivals is assumed. This, however, can also be viewed as a weakness if some statistical information were available. In multi-hop networks, some information about the packet arrival pattern can be known in advance, or through adaptive learning. Taking advantage of such knowledge may help develop algorithms for these scenarios.

In addition to taking power considerations into account, our routing decisions should also take into account different channel conditions, especially in a wireless environment. The goal will be to develop optimal opportunistic routing algorithms that favor good channel conditions in order to minimize packet retransmissions, and thus avoid unnecessary wastage of battery resources.

Appendix

Proof of Theorem 1:

(A) We begin by proving the following useful result. Let $\mathcal{A}$ be the set of requests accepted by the E-WME algorithm, and k be the index of the last request. Given a node n, a time slot τ, we have

$$\sum_{\substack{j\in\mathcal{A}:n\in R(j),\\ T^s(j)\le\tau<\hat{\tau}_n(j+1)}} \{f_n(\lambda_n(j+1,\tau)) - f_n(\lambda_n(j,\tau))\} = \sum_{j=1}^{k}\{f_n(\lambda_n(j+1,\tau)) - f_n(\lambda_n(j,\tau))\}, \quad (12)$$

where $f_n(x) = 0$ if $x = 0, \forall n$.

We consider the following three cases that correspond to the complement of $\{j \in \mathcal{A} : n \in R(j), T^s(j) \le \tau < \hat{\tau}_n(j+1)\}$:

1) if $j \notin \mathcal{A}$, or $j \in \mathcal{A}$ but $n \notin R(j)$, the load created by the first j requests is the same as the first $(j-1)$ ones, since the j^{th} request has no impact on the energy of node n at time τ. It follows that $\lambda(j+1,\tau) = \lambda(j,\tau)$.
2) If $\tau \ge \hat{\tau}_n(j+1)$, then the energy consumed by the first j requests is fully recharged at time τ. It follows that $\lambda(j+1,\tau) = \lambda(j,\tau) = 0$.
3) If $\tau < T^s(j)$, then $\lambda(j+1,\tau) = \lambda(j,\tau)$, by the definition of $\lambda(j,\tau)$.

In the above three cases, $f_n(\lambda_n(j+1,\tau)) = f_n(\lambda_n(j,\tau))$ always holds since $\lambda(j+1,\tau) = \lambda(j,\tau)$. In other words, for any index j satisfying any of the above three conditions, the corresponding term in the summation of Equation (12) gives no contribution to the sum. Hence, we can calculate the summation according to a smaller set of indices, as indicated by the left hand side of Equation (12).

We are now ready to derive the relationship between the residual energy and the revenue secured by the online algorithm.

For any $j \in \mathcal{A}$, $n \in R(j)$, and τ satisfying $T^s(j) \leq \tau < \hat{\tau}_n(j)$,

$$\begin{aligned} Y_n(j,\tau) &:= u_n\{(\mu^{\lambda_n(j+1,\tau)}-1)-(\mu^{\lambda_n(j,\tau)}-1)\} \\ &= u_n(\mu^{\lambda_n(j,\tau)+\frac{l(j)e_n(R(j))}{u_n}}-\mu^{\lambda_n(j,\tau)}) \\ &= u_n\mu^{\lambda_n(j,\tau)}(\mu^{\frac{l(j)e_n(R(j))}{u_n}}-1) \\ &= u_n\mu^{\lambda_n(j,\tau)}(2^{\frac{l(j)e_n(R(j))}{u_n}\log\mu}-1). \end{aligned}$$

Since $2^x - 1 \leq x$ for $x \in [0,1]$ and using (A2), we have

$$\begin{aligned} Y_n(j,\tau) &\leq u_n\mu^{\lambda_n(j,\tau)}\frac{l(j)e_n(R(j))}{u_n}\log\mu \\ &\leq (\mu^{\lambda_n(j,\tau)}-1)l(j)e_n(R(j))\log\mu \\ &\quad +l(j)e_n(R(j))\log\mu. \end{aligned} \tag{13}$$

Similarly, for any $j \in \mathcal{A}$, $n \in R(j)$, and τ satisfying $\hat{\tau}_n(j) \leq \tau < \hat{\tau}_n(j+1)$, the following equation holds:

$$\begin{aligned} Y_n(j,\tau) &:= u_n\{(\mu^{\lambda_n(j+1,\tau)}-1)-(\mu^{\lambda_n(j,\tau)}-1)\} \\ &= u_n(\mu^{\lambda_n(j+1,\tau)}-1) \\ &\leq u_n\lambda_n(j+1,\tau)\log\mu \\ &= u_n\frac{l(j)e_n(R(j))-\sum_{t=\hat{\tau}_n(j)}^{\tau-1}\gamma_n(t)}{u_n}\log\mu \\ &\leq l(j)e_n(R(j))\log\mu. \end{aligned} \tag{14}$$

Summing $Y_n(j,\tau)$ over n, τ, and j, by the virtue of (12), (13), and (14), we have

$$\begin{aligned} &\sum_n\sum_\tau u_n(\mu^{\lambda_n(k+1,\tau)}-1) \\ &= \sum_n\sum_\tau\sum_{j=1}^{k} Y_n(j,\tau) \\ &= \sum_n\sum_\tau \sum_{\substack{j\in\mathcal{A}:n\in R(j),\\ T^s(j)\leq\tau<\hat{\tau}_n(j+1)}} Y_n(j,\tau) \\ &= \sum_{j\in\mathcal{A}}\sum_{n\in R(j)}\sum_{\tau=T^s(j)}^{\hat{\tau}_n(j+1)-1} Y_n(j,\tau) \\ &\leq \sum_{j\in\mathcal{A}}\sum_{n\in R(j)}\left\{\sum_{\tau=T^s(j)}^{\hat{\tau}_n(j)-1}(\mu^{\lambda_n(j,\tau)}-1)l(j)e_n(R(j))\log\mu \right. \\ &\quad \left. +\sum_{\tau=T^s(j)}^{\hat{\tau}_n(j+1)-1} l(j)e_n(R(j))\log\mu\right\}. \end{aligned}$$

Using the fact that $\beta(j)$ is accepted and part of assumption (A1), it follows that

$$\begin{aligned} &\sum_n\sum_\tau u_n(\mu^{\lambda_n(k+1,\tau)}-1) \\ &\leq \sum_{j\in\mathcal{A}}\log\mu\,\{\rho(j)+LTl(j)e_n(j)\} \\ &\leq \sum_{j\in\mathcal{A}}\log\mu\,\{\rho(j)+\rho(j)\} \\ &\leq \sum_{j\in\mathcal{A}}2\rho(j)\log\mu. \end{aligned} \tag{15}$$

Next, we derive the relationship between the residual energy and the *additional* revenue secured by the offline algorithm.

Let $\mathcal{Q}$ be the set of requests accepted by offline algorithm and rejected by our online algorithm, and $R(j)$ be the path chosen by any given offline algorithm for $\beta(j)$, $j \in \mathcal{Q}$. Since $\beta(j)$ is rejected by the E-WME algorithm,

$$\begin{aligned} \rho(j) &< \sum_{n\in R(j)} C_n(j,R) \\ &= \sum_{n\in R(j)}\sum_{\tau=T^s(j)}^{\hat{\tau}_n(j)-1}(\mu^{\lambda_n(j,\tau)}-1)l(j)e_n(R(j)) \\ &\leq \sum_{n\in R(j)}\sum_{\tau=T^s(j)}^{\hat{\tau}_n(j)-1}(\mu^{\lambda_n(k,\tau)}-1)l(j)e_n(R(j)). \end{aligned}$$

Summing over all $j \in \mathcal{Q}$ and exchanging the order of summation, we have

$$\begin{aligned} &\sum_{j\in\mathcal{Q}}\rho(j) \\ &\leq \sum_{j\in\mathcal{Q}}\sum_{n\in R(j)}\sum_{\tau=T^s(j)}^{\hat{\tau}_n(j)-1}(\mu^{\lambda_n(k+1)}-1)l(j)e_n(R(j)) \\ &= \sum_{n\in V}\sum_\tau u_n(\mu^{\lambda_n(k+1)}-1)\sum_{\substack{j\in\mathcal{Q}:n\in R(j),\\ T^s(j)\leq\tau<\hat{\tau}_n(j)}}\frac{l(j)e_n(R(j))}{u_n} \\ &\leq \sum_{n\in V}\sum_\tau u_n(\mu^{\lambda_n(k+1)}-1). \end{aligned} \tag{16}$$

The last step is true since

$$\sum_{\substack{j\in\mathcal{Q}:n\in R(j),\\ T^s(j)\leq\tau<\hat{\tau}_n(j)}}\frac{l(j)e_n(R(j))}{u_n}\leq 1,$$

i.e., the energy claimed by the offline algorithm from each node at each time slot cannot exceed the battery capacity of that node. Note that the condition $T^s(j) \leq \tau < \hat{\tau}_n(j)$ implies that the energy consumption due to $\beta(j)$ has not been replenished yet in our energy model.

Therefore $l(j)e_n(R(j))$ is part of the current "energy to replenish" for the offline algorithm.

Inequality (15) basically establishes the relationship between the secured revenue and the residual energy. Inequality (16) indicates that the additional revenue gained by the offline algorithm is upper-bounded by a function of the residual energy. Denoting the set of all calls accepted by the offline algorithm by $\mathcal{A}^*$, we have:

$$\begin{aligned}\frac{J_{\text{off}}}{J_{\text{on}}} &= \frac{\sum_{j\in\mathcal{Q}}\rho(j)+\sum_{j\in\mathcal{A}^*\setminus\mathcal{Q}}\rho(j)}{\sum_{j\in\mathcal{A}}\rho(j)} \\ &\leq \frac{\sum_{j\in\mathcal{Q}}\rho(j)+\sum_{j\in\mathcal{A}}\rho(j)}{\sum_{j\in\mathcal{A}}\rho(j)} \leq 1+2\log\mu.\end{aligned}$$

Recall that $\mu = 2(LFT+1)$, where L is the maximum hop count allowed for any path, F is a constant chosen large enough to satisfy (A1), T is the upper bound on the time it takes to fully charge an empty battery. Therefore $(1+2\log\mu) = O(\log|V|)$, since $L \leq |V|$.

It remains to be shown that our routing algorithm does not violate the energy constraint at each node. Again let $\mathcal{A}$ be the set of requests accepted by our online algorithm. Suppose by way of contradiction that $\beta(j)$ is the first accepted request to violate power capacity constraint at node n at its arrival time slot τ. (Due to the replenishment, the first time slot such a violation can happen is the arrival time slot.) Then

$$\lambda_n(j,\tau) > 1 - \frac{l(j)e_n(R(j))}{u_n}. \qquad (17)$$

From the above inequality and (A2),

$$\begin{aligned}\mu^{\lambda_n(j,\tau)} - 1 &> (\mu^{1-\frac{l(j)e_n(R(j))}{u_n}} - 1) \\ &\geq (\mu^{1-\frac{1}{\log\mu}} - 1) \\ &= \frac{\mu}{2} - 1 = LFT. \qquad (18)\end{aligned}$$

From (18),

$$\begin{aligned}(\mu^{\lambda_n(j,\tau)} - 1)l(j)e_n(R(j)) &> l(j)e_n(R(j))LFT \\ &\geq \rho(j).\end{aligned}$$

From the description of our algorithm, the above inequality shows that $\beta(j)$ could not have been accepted in the first place. Therefore our routing algorithm does not violate the energy constraint at each node.

(B) The proof of the lower bound follows the proof for the SVC case [2], where examples are shown to prove the lower bound of $\Omega(\log|V|)$ in circuit-switched networks. In these examples, capacity constraints are put on the links. However, it so happens that in the examples, every link (n,m) that can possibly have the maximum congestion connects to exactly one distinct node m. This property makes it relatively straightforward to convert the examples into special cases of the power-aware routing problems we study. In this case, the energy replenishment does not complicate the proof either. For these reasons and space considerations we omit the proof here. Interested readers are referred to our technical report [11]. ■

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cyirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.

[2] B. Awerbuch, Y. Azar, and S. A. Plotkin. Throughput-competitive on-line routing. In *IEEE Symposium on Foundations of Computer Science*, pages 32–40, 1993.

[3] DARPA. Darpa energy harvesting program, 2004. Available on `http://www.darpa.mil/dso/trans/energy/overview.html`.

[4] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[5] D. B. Johnson and D. A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[6] A. Kansal and M. B. Srivastava. An environmental energy harvesting framework for sensor networks. In *Proceedings of the 2003 international symposium on Low power electronics and design*, pages 481–486. ACM Press, 2003.

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

[8] L. Li and J. Halpern. Minimum energy mobile wireless networks revisited. In *Proc. IEEE International Conference on Communications (ICC)*, January 2001.

[9] Q. Li, J. A. Aslam, and D. Rus. Online power-aware routing in wireless ad-hoc networks. In *Mobile Computing and Networking*, pages 97–107, 2001.

[10] L. Lin, N. B. Shroff, and R. Srikant. A distributed power-aware routing algorithm with logarithmic competitive ratio for sensor networks. Technical report, Purdue University, 2002. Available on `http://yara.ecn.purdue.edu/~shroff/PRF/Lin02.pdf`.

[11] L. Lin, N. B. Shroff, and R. Srikant. Asymptotically optimal power-aware routing for multihop wireless networks with renewable energy sources. Technical report, Purdue University, 2004. Available on `http://web.ics.purdue.edu/~llin/PRF/Lin04.pdf`.

[12] S. A. Plotkin. Competitive routing of virtual circuits in ATM networks. *IEEE Journal of Selected Areas in Communications*, 13(6):1128–1136, 1995.

[13] MicroStrain Press Release. Microstrain wins navy contract for self powered wireless sensor networks, December 2003. Available on `http://www.microstrain.com/news_29.htm`.

[14] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1333–1344, 1999.

[15] S. Singh, M. Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Mobile Computing and Networking*, pages 181–190, 1998.

[16] C.-K. Toh. Online power-aware routing algorithms for sensor networks. *IEEE communications Magazine*, June 2001.

[17] J. Wieselthier, G. Nguyen, and A. Ephremides. Energy limited wireless networking with directional antennas: the case of session-based multicasting. In *Proc. IEEE Infocom*, 2002.