Energy-Aware Routing in Sensor Networks: A Large System Approach ⋆

Longbi Lin [a], Ness B. Shroff [a,∗], R. Srikant [b]

[a] *Center for Wireless Systems and Applications(CWSA), School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, USA*

[b] *Department of Electrical and Computer Engineering and Coordinated Science Lab, University of Illinois at Urbana-Champaign, IL 61801, USA*

Abstract

In many sensor networking environments, the sensor nodes have limited battery capacity and processing power. Hence, it is imperative to develop solutions that are energy-efficient and computationally simple. In this work, we present a simple static multi-path routing approach that is optimal in the large-system limit. In a network with energy replenishment, the largeness comes into play because the energy claimed by each packet is small compared to the battery capacity. This static routing scheme exploits the knowledge of the traffic patterns and energy replenishment statistics, but does not need to collect instantaneous information on node energy. We also develop a distributed solution of the optimal policy, as well as heuristics to build the set of pre-computed paths. The simulations verify that the static scheme outperforms leading dynamic routing algorithms in the literature, and is close to the optimal solution when the energy claimed by each packet is relatively small compared to the battery capacity.

Key words: Energy-Aware Routing, Sensor Network, Large System, Mathematical Programming/Optimization, Simulations

1 Introduction

Energy-aware routing in sensor networks has received significant attention in recent years [8,9,15,16,18,19]. Finding a good routing algorithm to prolong

⋆ Earlier versions of this paper appeared in WONS 2006.

∗ Corresponding author.
Email addresses: `llin@purdue.edu` (Longbi Lin), `shroff@ecn.purdue.edu` (Ness B. Shroff), `rsrikant@uiuc.edu` (R. Srikant).

the network lifetime is an important problem, since sensor nodes are usually quite limited in battery capacity and processing power. For exactly the same reason, complex routing algorithms do not work well in this scenario, due to excessive overhead. In this work, we are interested in finding a simple and static routing approach. We will show that under reasonable assumptions, this static routing algorithm suffices: it is optimal in the large-system limit. In our context, *largeness* comes into play because *the energy claimed by each packet is small compared to the battery capacity.*

In this work, we study the routing problem in sensor networks with energy replenishment. Energy sources, e.g., solar cells, can be attached to sensor nodes to prolong the network lifetime [3,4,14]. For any individual node, on the one hand, there is energy consumption, which is mainly due to radio communications [1]. There is, on the other hand, incoming energy from an energy source. From this point of view, the battery acts as an energy buffer. We will show the optimality of the static routing algorithm when this buffer size is large, or equivalently, when the energy claimed by individual packets is small compared to the battery capacity.

In [10], a dynamic routing algorithm, E-WME, was proposed for sensor networks with energy replenishment. One interesting feature of that algorithm is that it does not need any information on the statistics of the input traffic. The E-WME algorithm was shown to be asymptotically optimal in the competitive ratio sense when the number of nodes in the network is large. It is optimal since it achieves a performance ratio (with respect to the best offline algorithm) that is logarithmic in the number of nodes in the network. It is shown in [10] that no algorithm can do better than this algorithm, *if no knowledge about future packet arrivals is used.* However, what if we had some knowledge of the future packet arrivals? For instance, in a sensor network that collects video footage at regular intervals, the data rate may be known a priori. Armed with this kind of information, an algorithm should be able to perform better. In fact, the proposed static routing approach in this paper does exploit the available statistical information on the packet arrivals and energy replenishment.

Most solutions proposed in the energy-aware routing literature incorporate certain measures of nodal residual energy, and therefore are dynamic in nature [5,8,9,10,15,16,18,19]. For example, in [5,9,10,18], algorithms have been presented to optimize the lifetime of the network. These algorithms can be viewed as different attempts to combine the elements of two routing approaches: Minimum Energy (ME) routing, which uses the least energy, and Max-min routing that selects the route with the maximum bottleneck residual node energy. In contrast, our algorithm is a static scheme with distributed computation. As a result, it can be implemented with greatly reduced overhead.

The rest of this paper is organized as follows: in Section 2, we formulate the problem of energy-aware routing with energy replenishment, and present our energy queue model. In Section 3, we present our algorithm, show its optimality, and discuss the implications. We proceed by discussing some issues related to the implementation of the static approach in Sections 4 and 5. Numerical results are provided in Section 6. Finally, concluding remarks are presented in Section 7.

2 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.

There are I classes of packets. Each class is associated with a different source-destination pair, and possibly different energy requirements for the nodes along the path. Class i packets arrive to the network according to a point process of rate λ_i. We will soon discuss our detailed assumptions on the arrival process.

For class i, there are $\theta(i)$ pre-computed paths. We use H_{ij}^n to denote the routing matrix: $H_{ij}^n = 1$ if node n is in path j of class i, and $H_{ij}^n = 0$ otherwise. The routing probabilities on class i packets can be described as

$$\vec{p}_i = (p_{i1}, p_{i2}, \dots, p_{i\theta(i)}),$$

where p_{ij} denotes the probability of packets of class i being sent along the j^{th} path. We use $\vec{p} = (\vec{p}_1, \vec{p}_2, \dots, \vec{p}_I)$ to denote the total routing decision. In a dynamic routing scheme, p_{ij} can depend on instantaneous information such as residual energy and energy replenishment rates at different nodes, and therefore can be a function of time. In a static routing scheme, pre-computed values of $\vec{p}$ are used, which is the same as the static splitting probability in a proportional routing scheme.

Each node is modelled as an energy queue, as shown in Figure 1. The battery is a buffer of size K and the nodal energy source serves as the server of the queue. When a real packet is routed through a node (and therefore incurs some amount of energy to replenish), a "virtual packet" arrives to the energy queue associated with this node. Note that our main focus is to model the dynamics of the energy consumption/replenishment processes in the network. Since packet transmissions happen at a much smaller time-scale than the time-scale of energy replenishment, it is assumed that, when a real packet is routed

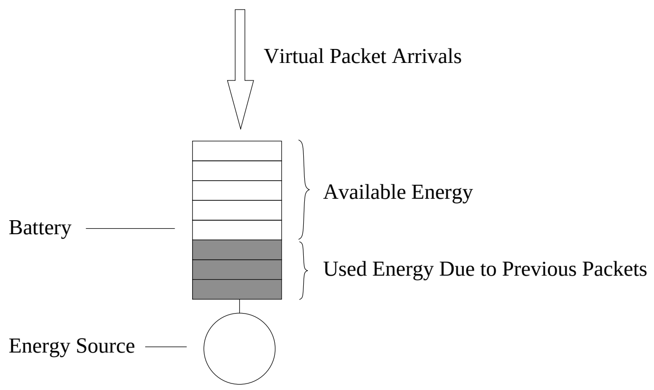

Fig. 1. The energy queue model: one queue case

through a path without blocking, a "virtual packet" arrives simultaneously to each of the energy queues along the path. This is illustrated by Figure 2. In other words, there is no notion of virtual packets leaving one energy queue and entering the other. Instead, the interaction of the queues happens through blocking and mean service time, which will be discussed next.

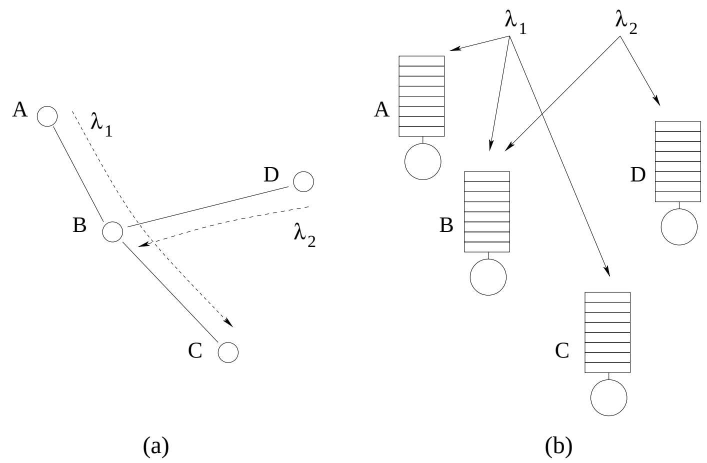

Fig. 2. The energy queue model: multiple queues

If a real packet cannot be sent to the next hop due to energy depletion at one of the nodes (say node n) along the route, this packet is blocked. In a dynamic scheme that utilizes instantaneous system state information, no virtual packet needs to be added to any of the energy queues along the path, since the real packet will be blocked in any case (i.e., there is no gain in admitting the packet). In a static scheme where instantaneous system state information may not be available, the packet may still be relayed down the path until it reaches node n. In this case, only the upstream nodes from node n receive the corresponding "virtual packet" arrivals.

The energy queue is work conserving: as long as there is at least one "virtual packet" in the queue, the energy source will be working on replenishing the used energy due to that packet. The time it takes for node n to replenish the energy consumed due to receiving and/or transmitting a packet of class i sent on path j is *i.i.d* with mean $1/\mu^n_{ij}$. The randomness is mainly due to the stochastic nature of the energy source: for a given energy source, the energy replenishment rate can vary from time to time, even though the overall process is assumed to be stationary. Note that the energy replenishment process could be non-stationary over long periods of time, e.g., night and day. However, this can be easily handled by developing a different solution for each time of day, e.g., using the distributed scheme described in Section 4. Furthermore, different energy sources can have different characteristics in replenishing a certain amount of energy. That is the reason why the service time distribution is assumed to be general. The average energy replenishment rate depends on the following three factors:

- *Node:* Heterogeneous energy distribution is allowed across the network.
- *Class:* Different classes of packets can have different energy requirements for the nodes along the path.
- *Path:* It is assumed that a node can have a complicated power control scheme in which multiple transmission power levels are used to communicate with different neighbors. Therefore, the energy requirement can be different depending on which neighbor it transmits to.

We assume that the packet arrival process, the energy replenishment process and routing algorithms that we consider ensure that the queueing system satisfies the following mild regularity conditions.

Assumptions:

(1) The system is stationary and ergodic:
  (a) Sample path averages of the system states exist almost surely.
  (b) The steady-state distribution exists and is ergodic. The sample path averages equal to the steady-state mean almost surely.
(2) The distribution of the inter-arrival time and the service time are such that for single-server queues,
  (a) The blocking probability of a queue with buffer size K is upper bounded by the tail probability of queue length exceeding K in a queue with infinite buffer space.
  (b) If the load $\rho = \lambda/\mu < 1$, where $1/\lambda$ and $1/\mu$ are the mean inter-arrival time and mean service time, respectively, then the expected queue length is finite in a queue with infinite buffer space.

Remarks: The assumptions on the arrival process and the service time distribution are quite general. (For instance, Assumption (2a) can be justified by a sample path argument in most cases.) A special case of such a single-server queue is $PH/G/1/\infty$, namely phase-type queue with general service time. More specifically, the inter-arrival time for class i packets is of phase-type distribution with mean $1/\lambda_i$. Any inter-arrival time can be approximated by a phase-type distribution [20]. In particular, the exponential inter-arrival time of a Poisson process is a special case of phase-type distribution.

Given a policy $\vec{g}$, let $N_{ij}^g(0,t)$ denote the total number of packets admitted to path j of class i during time window $[0,t]$. Define

$$\lambda_{ij}^g = \lim_{t\to\infty} \frac{N_{ij}^g(0,t)}{t}, \quad a.s. \tag{1}$$

to be the packet arrival rate of path j of class i.

Also let $N_i(0,t)$ denote the total number of class i packet arrivals in $[0,t]$. Clearly, we have

$$\lambda_i = \lim_{t\to\infty} \frac{N_i(0,t)}{t}, \quad a.s. \tag{2}$$

Define the average acceptance rate of class i packets on path j to be

$$q_{ij}^g = \frac{\lambda_{ij}^g}{\lambda_i}. \tag{3}$$

Then the total acceptance rate for class i packets is $\sum_{j=1}^{\theta(i)} q_{ij}^g$.

Let $U_i(\cdot)$ be a class-specific utility function of the total acceptance rate of this class. The utility function $U_i(x)$ measures the usefulness of having an acceptance rate of x for class i packets. We make the standard assumption that $U_i(\cdot)$ is strictly concave and non-decreasing for any class.

Our goal is to maximize the total utility:

$$\max_g \sum_{i=1}^{I} U_i(\sum_{j=1}^{\theta(i)} q_{ij}^g), \tag{4}$$

subject to energy constraints. Note that a packet can be routed through a path only if all the nodes along the path have sufficient energy (in other words, space in their buffer). The utility function can be chosen such that the above problem is equivalent to achieve some fairness objectives [12], or simply maximizing the total throughput.

It is worth noting that $q_{ij}^s = p_{ij}(1 - \mathbf{P}_{B,ij})$ for a static scheme, where p_{ij} is the pre-computed splitting probability and $\mathbf{P}_{B,ij}$ is the blocking probability of class i packets on path j.

We will show that there exists a static routing scheme whose performance is close to that of the best dynamic routing scheme if the energy claimed by each packet is small compared to the battery capacity. In other words, by carefully designing a static scheme, one can obtain most of the benefits of performance without the high cost of implementation.

3 Static Routing with Asymptotic Optimality

In this section, we will show that the performance of a carefully designed static routing scheme is asymptotically optimal in the large-system limit (i.e., the case when the energy claimed by a packet is vanishing small compared to the battery size). To this end, we first find an upper bound on the performance of the optimal dynamic routing scheme, construct a static routing scheme from the upper bound, and finally show the performance of the static scheme.

Before stating our main result, we introduce some notation here. In a system where each energy queue has a buffer of size K, let J_K^* denote the performance (the total utility as defined by Equation (4)) of the optimal dynamic routing scheme, and J_K^s the performance of the static scheme of interest.

Theorem 1 ***(Asymptotic Optimality of Static Routing)***

For all $\varepsilon > 0$, there exists a $\delta(\varepsilon) > 0$, such that

$$\limsup_{K\to\infty} J_K^* < \lim_{K\to\infty} J_K^s + \varepsilon, \tag{5}$$

where the static scheme uses the splitting probability from the solution $\vec{p}$ of the following optimization problem:

$$\begin{aligned} \max_{\vec{p}} \quad & \sum_{i=1}^{I} U_i\Big(\sum_{j=1}^{\theta(i)} p_{ij}\Big), \qquad (6)\\ \textit{subject to} \quad & p_{ij} \geq 0, \forall i, j, \\ & \sum_{j=1}^{\theta(i)} p_{ij} \in [0,1], \forall i, \\ & \sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} \frac{\lambda_i p_{ij} H_{ij}^n}{\mu_{ij}^n} \leq 1 - \delta(\varepsilon), \forall n \in V. \end{aligned}$$

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

Several remarks are in order:

(1) From Theorem 1, given a set of packet classes, as well as a set of pre-computed paths for each class, a static routing approach can be derived from optimization problem (6) whose total utility approaches the optimal value when the granularity of the battery gets finer and finer. The intuition behind this result is two-fold. First, from an energy conservation point of view, the constraints in optimization problem (6) give a fundamental limit on how much utility any dynamic algorithm can achieve, if p_{ij} in (6) is viewed as the average acceptance rate of class i packets on path j under this policy. Furthermore, the static approach using optimal splitting would be optimal, if there were no blocking, once a packet is assigned to a path. In fact, the probability of such blocking goes to zero, as the per-packet energy consumption becomes smaller and smaller, as compared to the battery size.

(2) Two types of blocking occur here: (a) the static controller decides from the splitting probability that a packet should be rejected at the source, and (b) a packet is admitted to one of the paths, but one or more of the nodes along the path does not have enough energy to forward this packet. When we say in the above paragraph that the blocking probably goes to zero, we are referring to the latter case. The reason that the blocking probability of type (b) goes to zero as the per-packet energy goes to zero can be understood as follows. If the battery energy is thought of as a queue, then the buffer size of this queue goes to infinity when the per-packet energy requirement goes to zero. Thus, if the arrival rate into the queue is less than the service rate (the energy replenishment rate), then there will be no blocking at the queue as the buffer size goes to infinity.

(3) The convergence of the performance of the static approach to the upper bound is at least as fast as $1/K$, where K is the battery size measured in per-packet energy consumption. This is evident from the second part of the proof in the Appendix.

The static approach is attractive for the following reasons:

- Unlike a dynamic routing algorithm, there is no need to collect information on instantaneous nodal energy. This amounts to a significant reduction in routing overhead, which in turn saves more energy for communications. In a practical system where some of the input parameters may be non-stationary (e.g., different average rate of energy replenishment due to seasonal solar radiation), one may need to recalculate the optimal routing probability. Nonetheless, such recalculations can be carried out at a much lower frequency.
- By using the static splitting probability from (6), the static approach adapts to the class-specific utility functions, the traffic load, the topology, and the available in-network energy resources. For instance, different shapes of utility function can lead to different ways of splitting the input traffic. Another example is the energy-aware admission control. If the offered traffic load is

quite heavy, with respect to the energy replenishment rate, (6) will produce an optimal solution that is more conservative in admitting the packets. In Section 6, we provide some numerical results to further justify the above claims.

4 Distributed Computation of the Optimal Splitting Probability

In practice, the packet arrival and energy replenishment processes can be non-stationary. Therefore, the optimal splitting probability needs to be calibrated from time to time to track these changes. As a result, it is desirable to solve optimization problem (6) efficiently even when the size of the network is relatively large. A centralized solution suffers from scalability issues in computation and communications. In this section, we describe our solution to compute the optimal splitting probability in a distributed fashion.

To this end, we consider a primal approach with penalty function [6,17]. The penalty function form of the optimization problem (6) is given by

$$\max_{\vec{p}} \quad \mathcal{V}(\vec{p}) = \sum_{i=1}^{I} \left[U_i\Big(\sum_{j=1}^{\theta(i)} p_{ij}\Big) + \xi \sum_{j=1}^{\theta(i)} U_i(p_{ij}) \right] - \sum_{n \in V} \int_0^{l^n(\vec{p})} f(y)dy - \sum_{i=1}^{I} \int_0^{\sum_{j=1}^{\theta(i)} p_{ij}} f(y)dy. \tag{7}$$

In the above expression, $l^n(\vec{p})$ is the load on node n, defined as

$$l^n(\vec{p}) = \sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \frac{\lambda_i p_{ij} H_{ij}^n}{\mu_{ij}^n} / (1-\delta),$$

and the price function $f(\cdot)$ is given by

$$f(y) = \frac{(y-1+\eta)^+}{\eta^\alpha},$$

where $\alpha > 1$, and $0 < \eta \ll 1$.

Note that we have added the term $\xi \sum_{j=1}^{\theta(i)} U_i(p_{ij})$ in (7) to ensure that the total utility is strictly concave in $\vec{p}$. It can be shown that $\mathcal{V}(\vec{p})$ is thus strictly concave and has a unique maximizer on $\{\vec{p} : p_{ij} \geq 0\}$.

Furthermore, as $\xi \to 0$ and $\eta \to 0$, the vector $\vec{p}$ that maximizes $\mathcal{V}(\vec{p})$ approximates arbitrarily closely the solution to (6). To solve the maximization problem of $\mathcal{V}(\vec{p})$ in a distributed fashion, we use the following algorithm to control the splitting probability $\vec{p}$ [17]:

$$\dot{p}_{ij} = k_{ij}(p_{ij}) \left\{ 1 - \frac{f(\sum_{\beta=1}^{\theta(i)} p_{i\beta}) + \sum_{n \in R(i,j)} f(l^n(\vec{p}))}{U_i'(\sum_{\beta=1}^{\theta(i)} p_{i\beta}) + \xi \sum_{\beta=1}^{\theta(i)} U_i'(p_{i\beta})} \right\}, \tag{8}$$

where $k_{ij}(\cdot)$ is any non-decreasing, continuous function such $k_{ij}(p) > 0$ for any $p > 0$, and $R(i, j)$ is the set of nodes in path j of class i. It can be shown that the function $\mathcal{V}(\vec{p})$ is a Lyapunov function for the above system of differential equations, and the stable point of the system maximizes $\mathcal{V}(\vec{p})$. (For a similar treatment, see [6].) *If $\vec{p}$ evolves according to the system of differential equations (8), all trajectories will converge to an approximate solution of (6).*

Intuitively, since $\mathcal{V}(\vec{p})$ is strictly concave and has a unique maximizer on $\{\vec{p} : p_{ij} \geq 0\}$, the optimal splitting probability satisfies the condition that the derivative against each p_{ij} should be 0. The control algorithm attempts to achieve this by controlling p_{ij} to equalize the rate of the total utility (the denominator in (8)) and the rate of the cost due to capacity constraints (the numerator in (8)).

The computation of p_{ij} can be done in parallel for different source nodes. From (8), the source node of class i can carry out the iterative computation of p_{ij} with all the information except the load on node n, which can be collected from the nodes along the path $R(i, j)$. This can be accomplished by updating the source when a packet is routed through this path, or utilizing a distributed Bellman-Ford type of routing mechanism.

5 Obtaining Pre-computed Paths

The proposed static approach is optimal *with respect to* the given set of pre-computed paths. Therefore, the quality of the pre-computed paths affects the optimality of the static solution. On the one hand, to maximize the total utility, it is desirable to have a very large set of alternative paths for each class; on the other hand, to lower the overall complexity of the algorithm, we need to limit the number of paths for each class. In fact, there is probably no need to enumerate all the paths between any source-destination pair. Consider the case of a network where the traffic load is relatively uniform. The path that goes through far away nodes probably would not be used even if it was included in the set of pre-computed paths. Therefore, we focus on finding a relatively small set of "good" paths.

Interestingly, it is a routing problem by itself to select a relatively small set of "good" paths in an energy-aware fashion. It is then natural to turn to a good dynamic routing heuristic, e.g., E-WME routing [10], to obtain a set of pre-computed paths. The idea is to first cache the paths used by the dynamic routing algorithm for each source-destination pair, and then use the static approach to further optimize on top of the given set of paths. The path caching only needs to be done infrequently and therefore the static approach dominates the routing performance.

More specifically, in the sensor networking scenario, one can design a routing setup phase which happens during the deployment of the network. Each node begins by simulating some dynamic routing protocol on the current topology. It is a simulation in the sense that nodes do not pass large data packets, and that they use a virtual battery instead of the real one. When any destination node receives a simulated data packet, it caches the path that the packet has traversed. At the end of this route setup phase, each destination node sends a route summary to the corresponding source node. The static approach can then calculate the routing probability using a distributed solution and take over the routing task.

6 Numerical Results

6.1 Interaction of Classes: A Simple Example

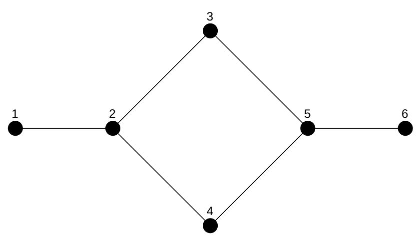

Fig. 3. Example of a small network

We now describe the results from our simulations. As shown in Figure 3, this network consists of 6 nodes and 6 links. All nodes have the same battery size. It takes unit energy to transmit a packet from one node to another. We consider only the energy consumption due to packet transmissions in simulations. (Note that reception energy is taken into account in our theoretical framework.) Table 1 shows the rate of energy replenishment at different nodes, where $1/\mu_n$ is the average time for node n to replenish the energy due to the transmission of one packet to next hop.

Table 1
Rate of energy replenishment at different nodes

μ_1	μ_2	μ_3	μ_4	μ_5	μ_6
1.0	1.0	3.0	3.0	1.0	1.0

There are three classes of packets. Each class of packets arrives to the network according to a Poisson process with unit arrival rate. The class-specific concave utility functions are

$$U_i(a) = \frac{\log(t_i a + 1)}{\log(t_i + 1)},$$

where a is the total acceptance rate of class i. The utility function is non-negative, equal to zero if $a = 0$, and equal to one if $a = 1$. The parameter $t_i > 0$ defines the concavity of the utility function: the larger the value of t_i, the more concave the utility function.

Table 2
Parameters of the three classes and the static solution

Class i	**Source**	**Destination**	$\mathbf{t_i}$	**Path(s)**	**Static Solution**
1	1	4	1	$R_{11} = [1, 2, 4]$	$p_{11} = 0.8640$
2	3	4	1	$R_{21} = [3, 2, 4]$	$p_{21} = 0.1350$
				$R_{22} = [3, 5, 4]$	$p_{22} = 0.7291$
3	6	4	100	$R_{31} = [6, 5, 4]$	$p_{31} = 0.2699$

Table 2 summarizes the parameters of the three classes and the static solution from optimization problem (6), where δ is chosen to be 0.001. We make the following observations:

- Class 1 and class 3 are symmetric in topology, nevertheless they have different acceptance rates in the static solution. This is because their utility functions are different. The energy available at nodes 2 and 5 is the bottleneck in this scenario. The way the energy resource, e.g., at node 5, is shared between class 2 and class 3 depends on the shape of the utility functions. The utility function of class 3 is more concave. In other words, given any acceptance rate, class 3 has an utility that is greater than or equal to that of class 1 or class 2. Therefore, in the optimal static solution, class 3 has the minimum total acceptance rate, since its marginal return diminishes faster than the other two.
- The interaction of class 1 and class 2 is through the resource contention at node 2. It turns out that they have the same *total* acceptance rate in the static solution, since they have the same utility function.
- All of the acceptance rates are non-zero due to the concavity of the utility functions.

Figure 4 shows that the total utility of the static routing approaches the upper bound as the battery size is increased. Each point of this figure is obtained by running the simulation with different random seeds (the topology remains unchanged) for 100 times and taking the average of the total end-to-end throughput in the steady state. The mean packet data rate is then substituted into the utility function to calculate the total network utility. The upper bound is computed from optimization problem (6), where δ is chosen to be zero.

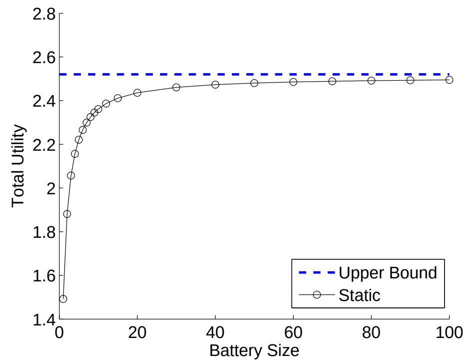

Fig. 4. Total utility as a function of battery size

It is evident from Figure 4 that the total utility of the static solution quickly approaches the upper bound, when the battery size is reasonably large. We have found this to be true in all of our simulation studies. For instance, if each battery supports transmitting 50 packets without replenishment, the gap between the upper bound and the static approach is less than 1.6%.

6.2 Distributed Computation of the Static Policy

In this section, we show the simulation results from the distributed computation of the static policy. The topology is the same as in Section 6.1.

To allow the mean rate of energy replenishment to change over time, we define four energy profiles in Table 3. At the beginning of each time period that lasts $T = 3750$ seconds, a profile is randomly selected. The mean rate of energy replenishment for each node is then configured according to this energy profile. Using the distributed computation in Section 4, we intend to verify that such an approach indeed tracks the changes in energy profiles.

Figure 5 and Figure 6 show the total utility and the underlying static splitting probability, respectively, as a function of time. The utility here is calculated based on the actual routed traffic, averaged over a sliding time window of 100 seconds. The upper bound is calculated according to (6) for each energy profile. Since the utility from the simulation is based on short term routed

Table 3
Rates of energy replenishment of different energy profile

Profile	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6
A	1.0	1.0	3.0	3.0	1.0	1.0
B	0.5	0.5	1.5	1.5	0.5	0.5
C	2.0	2.0	6.0	6.0	2.0	2.0
D	0.25	0.25	1.5	1.5	0.5	0.5

traffic, this utility sometimes exceeds the upper bound, which is a long term metric. Nevertheless, the performance of the static approach roughly tracks the changes in energy replenishment. This results from the fact that the distributed calculation of the static routing probability quickly finds the optimal solution for each energy profile when the change occurs.

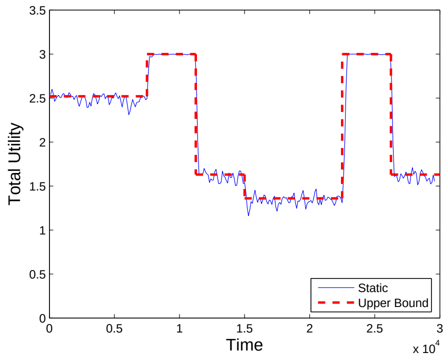

Fig. 5. Total utility as a function of time

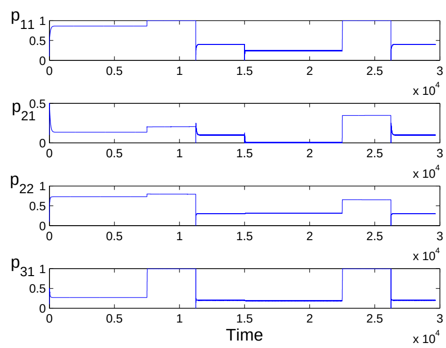

Fig. 6. Splitting probability as a function of time

6.3 Throughput Comparison: Static versus Dynamic

For this set of simulations, we randomly deploy 40 nodes on a 10×10 field. Again, all nodes have the same battery size, and it takes unit energy to transmit a packet from one node to another. Furthermore, all nodes have a uniform transmission range (we choose this transmission range to be 3 so that the network is initially connected). There is a link between nodes n and m if and only if (a) the distance between them is less than or equal to the transmission range of a node and (b) node n has enough energy to transmit a packet from n to m directly.

There are 16 classes of packets, each class with a randomly generated source-destination pair. The packets in each class arrive at the source node according to a Poisson process with rate $\lambda_i = 0.6$. For each node, the time it takes to replenish the energy due to transmitting one packet is exponentially distributed with mean $\mu_n = 1$.

We compared the throughput performance of the following three routing approaches:

- E-WME [10] as the dynamic routing approach. The E-WME approach has a built-in admission control component. To decide whether to admit a packet into the network, the E-WME algorithm compares the per-packet revenue to the dynamic E-WME cost metric. Since we want to maximize the throughput performance, we set the per-packet revenue to be a constant.
- Static routing proposed in this paper, with $\delta = 0.001$. The set of pre-computed paths is generated by the E-WME algorithm. For each class of packets, the first 20 paths used by the E-WME algorithm are cached and later passed to the static routing solver to compute the static splitting probability. Since we want to maximize the throughput performance, for each class, we set the utility function to be proportional to the product of the arrival rate and the total acceptance rate.
- Greedy minimum hop routing. This is a greedy approach in the following sense. On the one hand, this approach tries to take as little energy as possible from the network each time by choosing the path with minimum hop count. On the other hand, there is no admission control. As long as there is at least one path with enough energy connecting the source to the destination, the packet will be accommodated.

The E-WME algorithm is selected for comparison since it has been shown to outperform other energy-aware routing algorithms in the literature [10]. Furthermore, in a competitive ratio sense, the E-WME algorithm is optimal when number of nodes in the network is large.

Greedy minimum hop routing is chosen because it is a natural way to "saturate" the network in order to determine the network throughput capacity, which is limited by the energy replenishment. This should provide a base line approach to which we can compare the more sophisticated static and dynamic algorithms.

We now sketch the static solution by describing the routing decision on class 16, as shown in Figure 7. Class 16 packets travel from node 36 to node 6. As we can see from Figure 7, the static routing solution uses three paths. An incoming packet of class 16 is rejected with probability $P_{\text{reject}} = 0.2062$. An accepted packet is then assigned to one of the three paths with different probability, as indicated in Figure 7. It is interesting to note that the shortest path (shortest in hop count) is not the most preferred path in terms of splitting probability. This is because the routing decision depends on other classes of traffic, in addition to class 16 traffic. The need to load balance here outweighs the importance of using the minimum resources. This is consistent with some observation made in the dynamic routing literature [5,10,13] and the online load balancing literature [2].

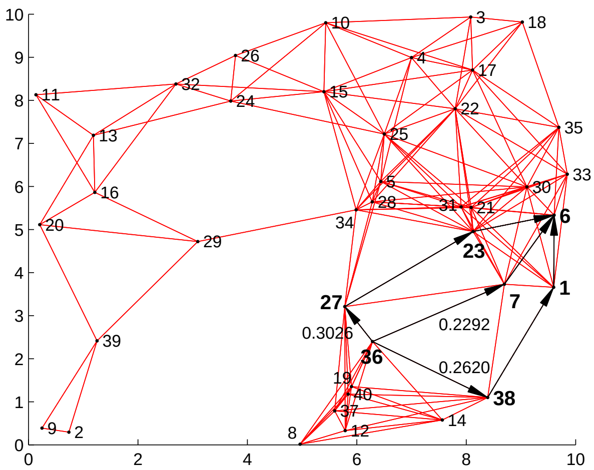

Fig. 7. Multi-path routing solution for class 16

Figure 8 shows the throughput comparison of the three routing approaches as a function of the battery size. The throughput here is the total long-term rate that the network can support, summing over all 16 classes. Each point of this figure is obtained by running the simulation with different random seeds (the topology remains unchanged) 10 times and taking the average of the total end-to-end throughput in the steady state. The upper bound is computed from optimization problem (6), where δ is chosen to be zero.

It is evident from Figure 8 that the static approach outperforms the E-WME approach and the greedy minimum hop approach, and the throughput performance is close to the upper bound, when the battery size is reasonably large. For instance, if each battery supports transmitting 200 packets without replenishment, the gap between the upper bound and the static approach is less than 0.12%.

There are two major differences between the static algorithm and the dynamic E-WME algorithm:

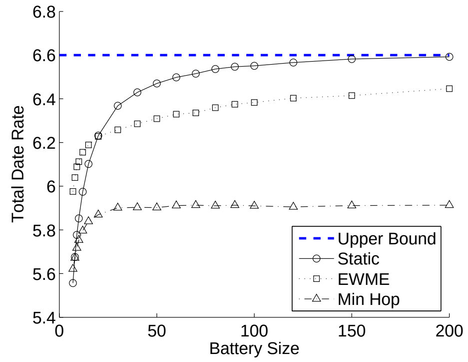

Fig. 8. Throughput comparison of the three routing approaches

- They belong to different information regimes [7]. In the static case, we are using statistical information about the input traffic and the rate of energy replenishment at the nodes. In the dynamic case, the E-WME algorithm does not utilize this kind of information explicitly. So this shows how much gain obtained if one knows rough information about the traffic, replenishment rate, etc.
- The static approach does not require instantaneous information on node residual energy, which amounts to potentially smaller amount of routing updates.

In setting up the E-WME approach, we use a set of parameters specified in [10]. If we use this routing approach as a heuristic cost metric and further fine-tune the E-WME routing parameters, it is possible to see even better throughput performance than with static routing. However, this kind of fine-tuning is topology-dependent: different network topology leads to different optimal dynamic routing parameters. Traffic pattern and rate of energy replenishment also have some impact on the choice of the optimal routing parameters. In the static case, this is automatically taken care of, since optimization problem (6) uses the topology/traffic rate/replenishment rate as the optimization constraints.

In general, it may not be important to ask which approach, static or dynamic, provides the best solution. Rather, it is more about the question of which approach is more suitable for a given scenario. If the application scenario is such that we know some information about traffic/replenishment pattern, but the cost is high to obtain information on instantaneous residual energy, the static approach is preferred. Otherwise, the dynamic approach can be an attractive alternative.

7 Conclusion

In this work, we address the problem of energy-aware routing in networks with renewable energy sources. Our energy model allows different kinds of energy sources and heterogeneous energy source distribution. The network model allows different classes of packets to have different energy requirement and class-specific utility function, which is defined on the total acceptance rate. We show that a carefully designed static routing algorithm can have performance that is arbitrarily close to that of the best dynamic routing algorithm, if the energy claimed by each packet is small compared to the battery capacity. In other words, by carefully designing a static scheme, one can obtain most of the benefits of performance without the high cost of implementation. The results from our simulations confirm the above claim.

The following are possible directions for future work. To compute the static solution in a distributed fashion, one can use a duality approach [11], instead of using a primal algorithm with a penalty function. It would be interesting to compare our primal approach to the duality-based proximal minimization solver, in terms of complexity and accuracy.

The performance of the static algorithm approaches the network capacity, which is constrained by the available energy. In addition to taking energy 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 simple and static cross-layer algorithms that favor good channel conditions in order to minimize packet retransmissions, and thus avoid unnecessary wastage of battery resources.

APPENDIX

A Proof of Theorem 1:

(a) Let J_{ub} be the maximum value of optimization problem (6). We first show that J_K^* is upper bounded by $J_{ub} + \varepsilon$.

Let $\tilde{J}_{ub}$ be the maximum value of the follow optimization problem:

$$\max_{\vec{p}} \quad \sum_{i=1}^{I} U_i(\sum_{j=1}^{\theta(i)} p_{ij}), \tag{A.1}$$
$$\text{subject to} \quad p_{ij} \geq 0, \forall i, j,$$

$$\sum_{j=1}^{\theta(i)} p_{ij} \in [0,1], \forall i,$$
$$\sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} \frac{\lambda_i p_{ij} H_{ij}^n}{\mu_{ij}^n} \leq 1, \forall n \in V.$$

Compared to optimization problem (6), the only difference is that the right hand side of the last inequality is now 1, instead of $(1-\delta(\varepsilon))$.

Let $N_{ij}^*(0,t)$, λ_{ij}^*, and q_{ij}^* be the corresponding quantities in Equations (1) and (3) for the optimal dynamic scheme. Since

$$\sum_{j=1}^{\theta(i)} N_{ij}^*(0,t) \leq N_i(0,t),$$

from Equations (1), (2), and (3), it is evident that (q_{ij}^*) satisfies

$$q_{ij}^* \geq 0, \forall i,j, \text{ and } \sum_{j=1}^{\theta(i)} q_{ij}^* \in [0,1], \forall i, \ a.s. \tag{A.2}$$

Furthermore, since each packet in $N_{ij}^*(0,t)$ is eventually served by the server at node n, if $H_{ij}^n = 1$, we apply Little's Law on the server at node n for the class i packets admitted and sent on its j^{th} path:

$$\mathbf{E}\{L_{ij}^n\} = \frac{\lambda_{ij}^* H_{ij}^n}{\mu_{ij}^n},$$

where L_{ij}^n is the in-server queue length of packets from class i, path j. Since the server either processes one packet when it is busy, or zero when it idles, we have

$$\begin{aligned}
&\mathbf{P}\{\text{Server Busy at node } n\} \\
=&\mathbf{E}\{\sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} L_{ij}^n\} \\
=&\sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} \mathbf{E}\{L_{ij}^n\} \\
=&\sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} \frac{\lambda_{ij}^* H_{ij}^n}{\mu_{ij}^n}.
\end{aligned} \tag{A.3}$$

The LHS of the above equation is a probability measure and therefore upper bounded by 1. Since $\lambda^*_{ij} = \lambda_i q^*_{ij}$, we have

$$\sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} \frac{\lambda_i q^*_{ij} H^n_{ij}}{\mu^n_{ij}} \leq 1. \tag{A.4}$$

From Equations (A.2) and (A.4), (q^*_{ij}) satisfies the constraints in optimization problem (A.1). It follows that

$$J^*_K \leq \tilde{J}_{ub}. \tag{A.5}$$

We claim that the following relationship also holds:

$$\tilde{J}_{ub} < J_{ub} + \varepsilon. \tag{A.6}$$

From Equation (A.5) and (A.6), it follows that $J^*_K < J_{ub} + \varepsilon$, which what we want to prove in Part (a).

Now let us show Equation (A.6) is indeed true. Let $(p^0_{ij})_{ij}$ be the solution of optimization problem (A.1). Clearly, $\forall \delta \in (0, 1)$, $(1 - \delta)(p^0_{ij})_{ij}$ satisfies the constraint in optimization problem (6). The corresponding function value of (6) is denoted as J^0_{ub}. It follows that

$$J^0_{ub} \leq J_{ub}. \tag{A.7}$$

Also,

$$\begin{aligned}
&\tilde{J}_{ub} - J^0_{ub} \\
=&\sum_{i=1}^{I}\left[U_i(\sum_{j=1}^{\theta(i)} p^0_{ij}) - U_i((1-\delta)\sum_{j=1}^{\theta(i)} p^0_{ij})\right] \\
\leq&\sum_{i=1}^{I} C\delta \sum_{j=1}^{\theta(i)} p^0_{ij},
\end{aligned} \tag{A.8}$$

where C is a constant. The above inequality is true since each $U_i(\cdot)$ is concave and therefore Lipschitz on the compact constraint set, and there are only finite number of such functions. Furthermore, we can choose $\delta > 0$ small enough such that the RHS of (A.8) is less than any given ε. In other words, we have

$$\tilde{J}_{ub} - J^0_{ub} < \varepsilon. \tag{A.9}$$

From Equations (A.7) and (A.9), we are done proving our claim (A.6).

To sum up, in Part (a), we show that $J_K^* < J_{ub} + \varepsilon$.

(b) Let $(p_{ij})_{ij}$ be the solution to optimization problem (6), then

$$J_{ub} = \sum_{i=1}^{I} U_i(\sum_{j=1}^{\theta(i)} p_{ij}).$$

The performance of the static scheme using $(p_{ij})_{ij}$ as the splitting probability is

$$J_K^s = \sum_{i=1}^{I} U_i(\sum_{j=1}^{\theta(i)} p_{ij}(1 - P_{B,ij})),$$

where $P_{B,ij}$ is the blocking probability of packets from class i, on path j.

If we can show that the blocking probability goes to zero as $K \to \infty$, it follows that $J_K^s \to J_{ub}$. This, combined with Part (a), gives the conclusion in the theorem.

We now show that the blocking probability $P_{B,ij}$ goes to zero as $K \to \infty$.

Let S denote the original system of energy queues with buffer size K, and $\tilde{S}^n$ the following system: node n still has buffer size K, and all other nodes have infinite buffer space. Let $P_{K,ij}^n$ and $\tilde{P}_{K,ij}^n$ denote the probability of the energy queue at node n being full in system S and $\tilde{S}$, respectively.

In the original system S, if the energy queue is full in one of the upstream nodes of node n, the virtual packet will not arrive to the queue at node n. Compared to system S, at least as many virtual packets arrive to the queue at node n in system $\tilde{S}$. Using a sample path argument, it is clear that $P_{K,ij}^n \leq \tilde{P}_{K,ij}^n$.

Let $R_{ij} = (n_1, n_2, \ldots, n_{m(i,j)})$ be path j of class i packets. Let M be the upper bound on the hop count of any path. We define event B_{ij} to be the event that a packet of class i assigned to path j is blocked, and F_n to be the event that energy queue at node n full. The blocking probability of class i packets assigned to path j is

$$\begin{aligned}
P_{B,ij} &= \mathbf{Pr}\{B_{ij}\} \\
&= \mathbf{Pr}\{\bigcup_{n \in R_{ij}} F_n\} \\
&\leq \sum_{n=n_1}^{n_{m(i,j)}} P_{K,ij}^n \\
&\leq \sum_{n=n_1}^{n_{m(i,j)}} \tilde{P}_{K,ij}^n \\
&\leq M \max_{n \in R_{ij}} \tilde{P}_{K,ij}^n. && \text{(A.10)}
\end{aligned}$$

From the above formula, to show $P_{B,ij} \to 0$, it suffices to show that $\tilde{P}^n_{K,ij} \to 0$. Note that $\tilde{P}^n_{K,ij}$ is the blocking probability of a single $\cdot/1/K$ queue with multiple classes of arrivals. Consider the corresponding $\cdot/1/\infty$ queue, and define the overflow probability

$$\tilde{P}^n_{K,\infty} = \mathbf{Pr}\{Q_n \geq K\},$$

where $Q_n \in \mathbb{Z}^+$ is the workload of energy queue at node n.

By Assumption (2a),

$$\tilde{P}^n_K \leq \tilde{P}^n_{K,\infty}. \tag{A.11}$$

Now we have a queue with infinite buffer, and we want to show its blocking probability $\tilde{P}^n_{K,\infty}$ goes to zero, as $K \to \infty$. The arrival rate to this queue is

$$\tilde{\lambda}_n = \sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \lambda_i p_{ij} H^n_{ij}. \tag{A.12}$$

A packet from class i on path j has a mean service time of $1/\mu^n_{ij}$, therefore the overall mean service time is

$$\frac{1}{\tilde{\mu}_n} = \frac{\sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \frac{1}{\mu^n_{ij}} \lambda_i p_{ij} H^n_{ij}}{\sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \lambda_i p_{ij} H^n_{ij}}. \tag{A.13}$$

From (A.12) and (A.13), the overall load is

$$\tilde{\rho}_n = \tilde{\lambda}_n \frac{1}{\tilde{\mu}_n} = \sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \frac{1}{\mu^n_{ij}} \lambda_i p_{ij} H^n_{ij}. \tag{A.14}$$

Recall that $(p_{ij})_{ij}$ is the solution to optimization problem (6). It follows that $\tilde{\rho}_n < 1$. By Assumption (2b), $\mathbf{E}\{Q_n\}$ is finite. From Markov Inequality,

$$\tilde{P}^n_{K,\infty} = \mathbf{Pr}\{Q_n \geq K\} \leq \frac{\mathbf{E}\{Q_n\}}{K}. \tag{A.15}$$

Therefore, $\tilde{P}^n_{K,\infty} \to 0$, as $K \to \infty$. It follows from (A.11) that $\tilde{P}^n_K \to 0$, as $K \to \infty$.

Q.E.D.

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] Y. Azar, B. Kalyanasundaram, S. Plotkin, K. R. Pruhs, and O. Waarts. On-line load balancing of temporary tasks. *Journal of Algorithms*, 22(1):93–110, January 1997.

[3] DARPA. Surface wave energy harvesting, 2005. Available on `http://www.darpa.mil/ato/programs/SWEH/DT.htm`.

[4] 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.

[5] K. Kar, M. Kodialam, T. V. Lakshman, and L. Tassiulas. Routing for network capacity maximization in energy-constrained ad-hoc networks. *IEEE INFOCOM'03*, 2003.

[6] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49, 1998.

[7] E. Koutsoupias and C. H. Papadimitriou. Beyond competitive analysis. In *IEEE Symposium on Foundations of Computer Science*, pages 394–400, 1994.

[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 *Proc. the Seventh Annual International Conference on Mobile Computing and Networking (ACM Mobicom'01)*, pages 97–107, 2001.

[10] L. Lin, N. B. Shroff, and R. Srikant. Asymptotically optimal power-aware routing for multihop wireless networks with renewable energy sources. In *IEEE INFOCOM'05*, Miami, Florida, March 2005.

[11] X. Lin and N. B. Shroff. An optimization based approach for quality of service routing in high-bandwidth networks. In *IEEE INFOCOM'04*, Hong Kong, March 2004.

[12] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, 2000.

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

[14] MicroStrain Press Release. Microstrain wins navy contract for self powered wireless sensor networks, December 2003. Available on `http://www.microstrain.com/news/article-29.aspx`.

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

[16] 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.

[17] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser, Boston, MA, 2003.

[18] C.-K. Toh. Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks. *IEEE communications Magazine*, 39(6):138–147, June 2001.

[19] J. Wieselthier, G. Nguyen, and A. Ephremides. Energy limited wireless networking with directional antennas: the case of session-based multicasting. In *Proc. IEEE INFOCOM'02*, New York, 2002.

[20] R. Wolff. *Stochastic modeling and the theory of queues*. Prentice Hall, 1989.