Energy-Aware Routing in Sensor Networks: A Large Systems Approach

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

***Abstract*— Sensor network nodes are often limited in battery capacity and processing power. Thus, it is imperative to develop solutions that are both energy and computationally efficient. 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. Compared to the other routing algorithms in the literature, this static routing scheme exploits the knowledge on the patterns of traffic and energy replenishment, and does not need to collect instantaneous information on node energy. We also outline possible approaches for a distributed computation of the optimal policy, and propose 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 optimal when the energy claimed by each packet is relatively small compared to the battery capacity.**

***Index Terms*— Energy-Aware Routing, Sensor Network, Large System, Mathematical Programming/Optimization, Simulations**

I. Introduction

Energy-aware routing problem in sensor networks has received significant attention in recent years [10], [11], [16], [18], [20], [21]. Finding a good routing algorithm to prolong 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 also show that under reasonable assumptions, this static routing algorithm suffices: it is optimal in the large system limit.

In our context, the *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 [5], [6], [15]. 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 the 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, the energy claimed by individual packets is small compared to the battery capacity.

In [12], a dynamic routing algorithm, E-WME, was proposed for sensor networks with energy replenishment. It was shown to be asymptotically optimal when the number of nodes in the network is large. One interesting feature of this algorithm is that it does not need any information on the statistics of the input traffic. The E-WME algorithm 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 [12] that no algorithm can do better than this algorithm, *if no knowledge about future packet arrivals is present.* 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 exploits the available statistical information on the packet arrivals and energy replenishment.

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 energy queue model. In Section III, 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 Section IV. Numerical results are provided in Section V. Finally, concluding remarks are presented in Section VI.

L. Lin and N.B. Shroff are with Center for Wireless Systems and Applications (CWSA), School of ECE, Purdue University (email: llin@purdue.edu, shroff@ecn.purdue.edu).

R. Srikant is with the Department of Electrical and Computer Engineering and the Coordinated Science Lab, University of Illinois at Urbana-Champaign, (email: rsrikant@uiuc.edu).

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.

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 Poisson process with rate λ_i. 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 decision on class i packets can be described as

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

where p_{ij} denotes the probability of packets of class i being sent along the j^{th} path. We use $\vec{p} = (\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_I)$ 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 $\vec{p}$ is used, which is the same as the static splitting probability in a proportional routing scheme.

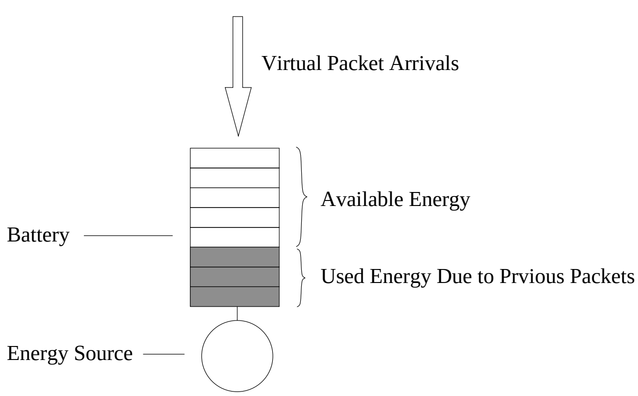

Fig. 1. The energy queue model: one queue case

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 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 packets leaving one energy queue and enter the other. Instead, the interaction of the queues happens through blocking and mean service time, which will be discussed next.

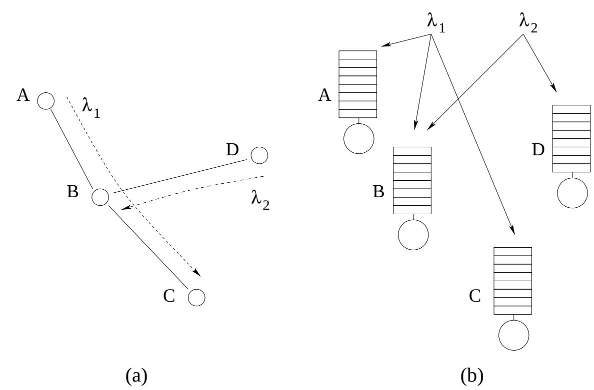

Fig. 2. The energy queue model: multiple queues

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 which 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 consumption due to receiving and/or transmitting a packet of class i sent on path j is *i.i.d* with mean $1/\mu_{ij}^n$. 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 stationary. [1] 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 requirement 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.

Since the queueing system has finite buffers and the input process is memoryless, under any dynamic policy $\vec{g}$, we assume the system states (the vector of the backlog of all the energy queues) evolve according to a stationary and ergodic stochastic process.

Consider any 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

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

to be the packet arrival rate of path j of class i. This is well defined since the system is stationary and ergodic.

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

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

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 usual assumption that $U_i(\cdot)$ is strictly concave and non-decreasing for any class.

[1] Note that the energy replenishment process could be non-stationary over very long periods of time, e.g., night and day. However, this can be easily handled by developing different solution for each time of day.

Our goal is to maximize the total weighted utility:

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

subject to energy constraints: 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).

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 a static routing scheme can have performance that is arbitrarily 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 reap most of the benefits of performance without the high cost of implementation.

III. 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. 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 optimal performance of the static scheme if the energy claimed by each packet is small compared to the battery capacity.

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

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

$\forall \varepsilon > 0, \exists \delta(\varepsilon) > 0$, s.t.

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

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

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

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

Remarks:

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 weighted utility approaches the optimal value when the granularity of the battery gets finer and finer. The intuition behind this result is two-fold. First of all, 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 the optimal splitting would be in fact 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) There are in fact two types of blocking taking place 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, however, one or more of the nodes along the path in fact 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 why it goes to zero is that the ability for the battery to absorb the difference between the incoming and outgoing energy becomes stronger and stronger, as the per-packet energy consumption becomes smaller and smaller. It gives one an illusion that it is the increase in the initial energy that gives rise to the decrease in blocking probability. In fact, in Theorem 1, we do not make any assumption on the initial energy, except that it is certainly upper bounded by the size of the battery. We can assume that all nodes have an empty battery to begin with, and still prove the result of Theorem 1.
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 great reduction in routing overhead, which in turn saves more energy in 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 V, we provide some numerical results to further justify the above claims.

IV. Implementing the Static Algorithm

A. Distributed Computation of the Optimal Splitting Probability

To find a way to compute the optimal splitting probability in a distributed fashion, one possibility is to consider the dual of (6). The major challenge here is that the dual function may not be differentiable due to the lack of strict concavity of the primal function. As pointed out in [4], there are at least two ways to handle this problem: one is to use nondifferentiable optimization [3] to solve the dual, and the other is to use the proximal minimization algorithm [13]. We will study the use of both approaches for future work.

B. *Obtaining Pre-computed Paths*

The proposed static approach is optimal *with respect to* the given set of pre-computed path. 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 enough, 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 [12], 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.

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.

V. Numerical Results

A. *Interaction of Classes: A Simple Example*

We now describe the results from our simulations. As described 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. Table I 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.

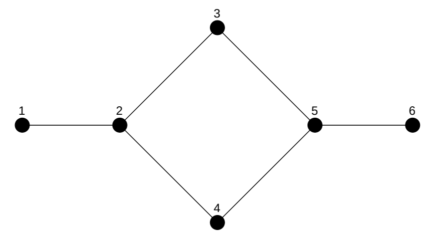

Fig. 3. Example of a small network

TABLE I
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 is

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

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 t_i is, the more concave the utility function is.

Table II summarizes the parameters of the three classes and the static solution from optimization problem (6), where δ is chosen to be 0.001. We have 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, since we only consider energy consumption due to packet transmissions. The way the energy resource, e.g., at node 5, is shared between class 2 and class 3 depends on the shape of their utility function. 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

TABLE II

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	$R_{21} = [3, 2, 4]$ $R_{22} = [3, 5, 4]$	$p_{21} = 0.1350$ $p_{22} = 0.7291$
3	6	4	100	$R_{31} = [6, 5, 4]$	$p_{31} = 0.2699$

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 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 resulted 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.

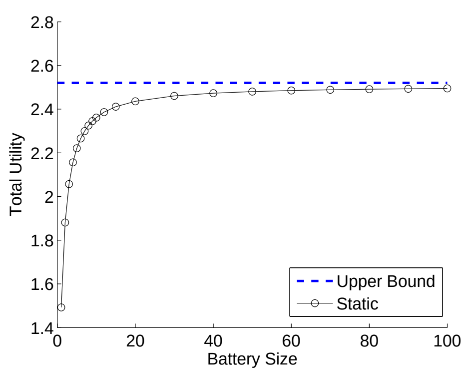

Fig. 4. Total utility as a function of battery size

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. 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%.

B. 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 an 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 to their 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 [12] 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, we set the utility function to be proportional to 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 is among the best dynamic energy-aware routing algorithms. In [12], it is shown that it outperforms other energy-aware routing algorithms in the literature. Furthermore, in a competitive ratio sense, the E-WME algorithm is optimal when number of nodes in the network is large.

The 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 5. Class 16 packets travel from node 36 to node 6. As we can see from Figure 5, 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 5. 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 [7], [12], [14] and the online load balancing literature [2].

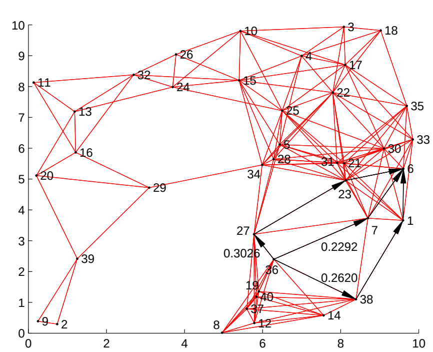

Fig. 5. Multi-path routing solution for class 16

Figure 6 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 the 16 classes. Each point of this figure is obtained by running the simulation with different random seeds (the topology remains unchanged) for 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.

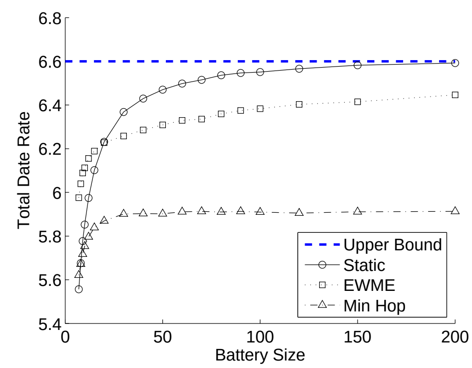

Fig. 6. Throughput comparison of the three routing approaches

It is evident from Figure 6 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:

- They belong to different information regimes [9]. 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 one interesting aspect shown by this set of simulation is how much it helps if one knows some 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 [12]. 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 the 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, is the optimal 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.

VI. 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 reap 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. Section IV-A lists a few options to compute the static solution in a distributed fashion. Instead of using a duality approach, one can also use a primal algorithm with a penalty function [19] to solve the optimization problem approximately. It would be interesting to compare this approach to the 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

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:

$$\begin{aligned} \max_{\vec{p}} \quad & \sum_{i=1}^{I} \lambda_i U_i(\sum_{j=1}^{\theta(i)} p_{ij}), \qquad (7)\\ \text{subject to} \quad & p_{ij} \geq 0, \forall i, j, \\ & \sum_{j=1}^{\theta(i)} p_{ij} \in [0,1], \forall i, \\ & \sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \frac{\lambda_i p_{ij} H_{ij}^n}{\mu_{ij}^n} \leq 1, \forall n \in V. \end{aligned}$$

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

$$q_{ij}^* \geq 0, \forall i, j, \text{ and } \sum_{j=1}^{\theta(i)} q_{ij}^* \in [0,1], \forall i. \qquad (8)$$

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

$$\begin{aligned}
&\mathbf{P}\{\text{Server Busy at node } n\} \\
= \;& \mathbf{E}\{\sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} L_{ij}^n\} \\
= \;& \sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} \mathbf{E}\{L_{ij}^n\} \\
= \;& \sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} \frac{\lambda_{ij}^* H_{ij}^n}{\mu_{ij}^n}. \qquad (9)
\end{aligned}$$

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

$$\sum_{i=1}^{I}\sum_{j=1}^{\theta(i)} \frac{\lambda_i q_{ij}^* H_{ij}^n}{\mu_{ij}^n} \le 1. \qquad (10)$$

From Equations (8) and (10), (q_{ij}^*) satisfies the constraints in optimization problem (7). It follows that

$$J_K^* \le \tilde{J}_{ub}. \qquad (11)$$

We claim that the following relationship also holds:

$$\tilde{J}_{ub} < J_{ub} + \varepsilon. \qquad (12)$$

From Equation (11) and (12), it follows that $J_K^* < J_{ub} + \varepsilon$, which what we want to prove in Part (a).

Now let us show Equation (12) is indeed true. Let $(p_{ij}^0)_{ij}$ be the solution of optimization problem (7). Clearly, $\forall \delta \in (0,1)$, $(1-\delta)(p_{ij}^0)_{ij}$ satisfies the constraint in optimization problem (6). The corresponding function value of (6) is denoted as J_{ub}^0. It follows that

$$J_{ub}^0 \le J_{ub}. \qquad (13)$$

Also,

$$\begin{aligned}
&\tilde{J}_{ub} - J_{ub}^0 \\
= \;& \sum_{i=1}^{I} \lambda_i \left[U_i(\sum_{j=1}^{\theta(i)} p_{ij}^0) - U_i((1-\delta)\sum_{j=1}^{\theta(i)} p_{ij}^0) \right] \\
\le \;& \sum_{i=1}^{I} \lambda_i C \delta \sum_{j=1}^{\theta(i)} p_{ij}^0, \qquad (14)
\end{aligned}$$

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 RHS of (14) is less than any given ε. In other words, we have

$$\tilde{J}_{ub} - J_{ub}^0 < \varepsilon. \qquad (15)$$

From Equations (13) and (15), we are done proving our claim (12).

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

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

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

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

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.

Since more virtual packets arrive to the queue at node n in system $\tilde{S}$, using a sample path argument, it is clear that $P_{K,ij}^n \le \tilde{P}_{K,ij}^n$.

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 $\mathcal{B}_{ij}$ to be the event that a packet of class i assigned to path j is blocked, and $\mathcal{F}_n$ to be the event that energy queue at node n full. The blocking probability of class i packets assigned to path j is

$$\begin{aligned}
P_{B,ij} &= \mathbf{Pr}\{\mathcal{B}_{ij}\} \\
&= \mathbf{Pr}\{\bigcup_{n \in R_{ij}} \mathcal{F}_n\} \\
&\le \sum_{n=n_1}^{n_{m(i,j)}} P_{K,ij}^n \\
&\le \sum_{n=n_1}^{n_{m(i,j)}} \tilde{P}_{K,ij}^n \\
&\le M \max_{n \in R_{ij}} \tilde{P}_{K,ij}^n. \qquad (16)
\end{aligned}$$

From the above formula, to show $P_{B,ij} \rightarrow 0$, it suffices to show that $\tilde{P}^n_{K,ij} \rightarrow 0$. Note that $\tilde{P}^n_{K,ij}$ is the blocking probability of a single $M/G/1/K$ queue with multiple classes of Poisson arrivals. From PASTA [17], $\tilde{P}^n_{K,ij} = \tilde{P}^n_K$, where $\tilde{P}^n_K$ is the probability of queue being full, as observed at an arbitrary time. Therefore, to calculate the blocking probability $\tilde{P}^n_K$, the queue can be viewed as a $M/G/1/K$ queue with a single class arrivals. Consider the corresponding $M/G/1/\infty$ 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.

A sample path of the workload in $M/G/1/K$ queue can be constructed from a sample path of $M/G/1/\infty$ queue by removing all the time intervals when the workload is above K [8]. It is thus clear that

$$\tilde{P}^n_K \leq \tilde{P}^n_{K,\infty}. \tag{17}$$

Now we have a $M/G/1$ queue with infinite buffer, and we want to show its blocking probability $\tilde{P}^n_{K,\infty}$ goes to zero, as $K \rightarrow \infty$. The arrival rate to this $M/G/1$ queue is

$$\tilde{\lambda}_n = \sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \lambda_i p_{ij} H^n_{ij}. \tag{18}$$

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

$$\frac{1}{\tilde{\mu}_n} = \frac{\sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \frac{1}{\mu^n_{ij}} \lambda_i p_{ij} H^n_{ij}}{\sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \lambda_i p_{ij} H^n_{ij}}. \tag{19}$$

From (18) and (19), the overall load is

$$\tilde{\rho}_n = \tilde{\lambda}_n \frac{1}{\tilde{\mu}_n} = \sum_{i=1}^{I} \sum_{j=1}^{\theta(i)} \frac{1}{\mu^n_{ij}} \lambda_i p_{ij} H^n_{ij}. \tag{20}$$

Recall that $(p_{ij})_{ij}$ is the solution to optimization problem (6). It follows that $\tilde{\rho}_n < 1$. We then invoke Pollaczek-Khinchine formulas [17], and the expected queue length is

$$\mathbf{E}\{Q_n\} = \left(\frac{\tilde{\rho}_n}{1 - \tilde{\rho}_n}\right) [1 - \frac{\tilde{\rho}_n}{2}(1 - \mu_n^2 \sigma_n^2)], \tag{21}$$

where σ_n^2 is the variance of the service time distribution. Since $\tilde{\rho}_n < 1$, $\mathbf{E}\{Q_n\}$ is finite. From Markov Inequality,

$$\tilde{P}^n_{K,\infty} = \mathbf{Pr}\{Q_n \geq K\} \leq \frac{\mathbf{E}\{Q_n\}}{K}. \tag{22}$$

Therefore, $\tilde{P}^n_{K,\infty} \rightarrow 0$, as $K \rightarrow \infty$. It follows from (17) that $\tilde{P}^n_K \rightarrow 0$, as $K \rightarrow \infty$.

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] M. L. Balinski and P. Wolfe. *Nondifferentiable Optimization*. North-Holland Publishing Company, Amsterdam, 1975.

[4] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, Englewood Cliffs, NJ, 1989.

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

[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'03*, 2003.

[8] F. Kelly. Notes on effective bandwidths. *Stochastic Networks: Theory and Applications*, pages 141–168, 1996.

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

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

[11] 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.

[12] 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.

[13] 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.

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

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

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

[17] M. Schwartz. *Telecommunication Networks, Protocols, Modeling and Analysis*. Addison-Wesley Publishing Company, November 1988.

[18] 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.

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

[20] 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.

[21] 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.