Utility Optimal Scheduling in Energy Harvesting Networks

Longbo Huang, Michael J. Neely [*†]

ABSTRACT

In this paper, we show how to achieve close-to-optimal utility performance in energy harvesting networks with only *finite* capacity energy storage devices. In these networks, nodes are capable of harvesting energy from the environment. The amount of energy that can be harvested is time varying and evolves according to some probability law. We develop an *online* algorithm, called the Energy-limited Scheduling Algorithm (ESA), which jointly manages the energy and makes power allocation decisions for packet transmissions. ESA only has to keep track of the amount of energy left at the network nodes and *does not require any knowledge* of the harvestable energy process. We show that ESA achieves a utility that is within $O(\epsilon)$ of the optimal, for any $\epsilon > 0$, while ensuring that the network congestion and the required capacity of the energy storage devices are *deterministically* upper bounded by bounds of size $O(1/\epsilon)$. We then also develop the Modified-ESA algorithm (MESA) to achieve the same $O(\epsilon)$ close-to-utility performance, with the average network congestion and the required capacity of the energy storage devices being only $O([\log(1/\epsilon)]^2)$.

Keywords

Energy Harvesting, Lyapunov Analysis, Stochastic Network, Queueing

1. INTRODUCTION

Recent developments in hardware design have enabled many general wireless networks to support themselves by harvesting energy from the environment. For instance, by converting mechanical vibration into energy [1], by using solar panels [2], by utilizing thermoeletric generators [3], or by converting ambient radio power into energy [4]. Such harvesting methods are also referred to as "recycling" energy [5]. This energy harvesting ability is crucial for many network design problems. It frees the network devices from having an "always on" energy source and provides a way of operating the network with a potentially infinite lifetime. These two advantages are particularly useful for networks that work autonomously, e.g., wireless sensor networks that perform monitoring tasks in dangerous fields [6], tactical networks [7], or wireless handheld devices that operate over a longer period [8], etc.

However, to take full advantage of the energy harvesting technology, efficient scheduling algorithms must consider the finite capacity for energy storage at each network node. In this paper, we consider the problem of constructing utility optimal scheduling algorithms in a discrete stochastic network, where the communication links have time-varying qualities, and the nodes are powered by *finite* capacity energy storage devices but are capable of harvesting energy. Every time slot, the network decides how much new data to admit and how much power to allocate over each communication link for data transmission. The objective of the network is to maximize the aggregate traffic utility subject to the constraint that the average network backlog is finite, and the "energy-availability" constraint is met, i.e., at all time, the energy consumed is no more than the energy stored. We see that the "energy-availability" constraint greatly complicates the design of an efficient scheduling algorithm, due to the fact that the current energy expenditure decision may cause energy outage in the future and thus affect the future decisions. Such problems can in principle be formulated as dynamic programs (DP) and be solved optimally. However, the DP approach typically requires substantial statistical

[*]Longbo Huang, (http://www-scf.usc.edu/∼longbohu) and Michael J. Neely (http://www-rcf.usc.edu/∼mjneely) are with the Dept. of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA. This material is supported in part by one or more of the following: the NSF Career grant CCF-0747525, NSF grant 0540420, the Network Science Collaborative Technology Alliance sponsored by the U.S. Army Research Laboratory W911NF-09-2-0053.

knowledge of the harvestable energy process and the channel state process, and often runs into the "curse-of-dimensionality" problem when the network size is large.

There have been many previous works developing algorithms for such energy harvesting networks. [9] develops algorithms for a single sensor node for achieving maximum capacity and minimizing delay when the rate-power curve is linear. [10] considers the problem of optimal power management for sensor nodes, under the assumption that the harvested energy satisfies a leaky-bucket type property. [11] looks at the problem of designing energy-efficient schemes for maximizing the decay exponent of the queue length. [12] develops scheduling algorithms to achieve close-to-optimal utility for energy harvesting networks with time varying channels. [13] develops an energy-aware routing scheme that approaches optimal as the network size increases. Outside the energy harvesting context, [14] considers the problem of maximizing the lifetime of a network with finite energy capacity and constructs a scheme that achieves a close-to-maximum lifetime. [15] and [16] develop algorithms for minimizing the time average network energy consumption for stochastic networks with "always on" energy sources. However, most of the existing results focus on single-hop networks and often require sufficient statistical knowledge of the harvestable energy, and results for multihop networks often do not give explicit queueing bounds and do not provide explicit characterizations of the needed energy storage capacities.

We tackle this problem using the Lyapunov optimization technique developed in [15] and [17], combined with the idea of weight perturbation, e.g., [18] and [19]. The idea of this approach is to construct the algorithm based on a quadratic Lyapunov function, but carefully perturb the weights used for decision making, so as to "push" the target queue levels towards certain nonzero values to avoid underflow (in our case, the target queue levels are the energy levels at the nodes). Based on this idea, we construct the Energy-limited Scheduling Algorithm (ESA) for achieving optimal utility in general multihop energy harvesting networks powered by finite capacity energy storage devices. ESA is an *online* algorithm which makes greedy decisions every time slot *without requiring any knowledge of the harvestable energy* and *without requiring any statistical knowledge of the channel qualities*. We show that the ESA algorithm is able to achieve an average utility that is within $O(\epsilon)$ of the optimal for any $\epsilon > 0$, and only requires energy storage devices that are of $O(1/\epsilon)$ sizes. We also explicitly compute the required storage capacity and show that ESA also guarantees that the network backlog is deterministically bounded by $O(1/\epsilon)$. Furthermore, we develop the Modified-ESA algorithm (MESA) to achieve the same $O(\epsilon)$ close-to-optimal utility performance with energy storage devices that are only of $O([\log(1/\epsilon)]^2)$ sizes. We note that the approach of using perturbation in Lyapunov algorithms is novel. It not only allows us to resolve the energy outage problem easily, but also enables an easy analysis of the algorithm performance.

Our paper is mostly related to the recent work [12], which considers a similar problem. [12] uses a similar Lyapunov optimization approach (without perturbation) for algorithm design, and achieves a similar $[O(\epsilon), O(1/\epsilon)]$ utility-backlog performance using energy storage sizes of $O(1/\epsilon)$ for single-hop networks. Multihop networks are also considered in [12]. However, the performance bounds for multihop networks are given in terms of unknown parameters. In our paper, we compute the explicit $O(1/\epsilon)$ capacity requirements for the data buffers and energy storage devices for general multihop networks for achieving the $O(\epsilon)$ close-to-optimal utility performance. We then also develop a scheme to achieve the same utility performance with only $O([\log(1/\epsilon)]^2)$ energy storage capacities.

Our paper is organized as follows: In Section 2 we state our network model and the objective. In Section 3 we first derive an upper bound on the maximum utility. Section 4 presents the ESA algorithm. The $[O(\epsilon), O(1/\epsilon)]$ performance results of the ESA algorithm are presented in Section 5. We then construct the Modified-ESA algorithm (MESA) in Section 6. Simulation results are presented in Section 7.

2. THE NETWORK MODEL

We consider a general interconnected network that operates in slotted time. The network is modeled by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N} = \{1, 2, ..., N\}$ is the set of the N nodes in the network, and $\mathcal{L} = \{[n, m],\ n, m \in \mathcal{N}\}$ is the set of communication links in the network. For each node n, we use $\mathcal{N}_n^{(o)}$ to denote the set of nodes b with $[n, b] \in \mathcal{L}$, and use $\mathcal{N}_n^{(in)}$ to denote the set of nodes a with $[a, n] \in \mathcal{L}$. We then define $d_{max} \triangleq \max_n |\mathcal{N}_n^{(in)}|$ to be the maximum in-degree that any node $n \in \mathcal{N}$ can have.

2.1 The Traffic and Utility Model

At every time slot, the network decides how many packets destined for node c to admit at node n. We call these traffic the *commodity* c data and use $R_n^{(c)}(t)$ to denote the amount of new commodity c data admitted. We assume that $0 \leq R_n^{(c)}(t) \leq R_{max}$ for all n, c with some finite R_{max} at all time. [1] We assume that each commodity is associated with a utility function

[1]Note that this setting implicitly assumes that nodes always have packets to admit. The case when the number of packets available is random can also be incorporated into our model and solved by introducing auxiliary variables, as in [20]. Also note this traffic admission model can be viewed as "shaping" the arrivals from some external sending nodes. One future extension of our model is to also consider the backlogs at these sending nodes.

$U_n^{(c)}(\overline{r}^{nc})$, where $\overline{r}^{nc}$ is the time average rate of the commodity c traffic admitted into node n, defined as $\overline{r}^{nc} = \liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\big\{R_n^{(c)}(\tau)\big\}$. Each $U_n^{(c)}(r)$ function is assumed to be increasing, continuously differentiable, and strictly concave in r with a bounded first derivative and $U_n^{(c)}(0) = 0$. We use β^{nc} to denote the maximum first derivative of $U_n^{(c)}(r)$, i.e., $\beta^{nc} = (U_n^{(c)})'(0)$ and denote $\beta = \max_{n,c}\beta^{nc}$.

2.2 The Transmission Model

In order to deliver the data to their destinations, each node needs to allocate power to each link for data transmission at every time slot. To model the effect that the transmission rates typically also depend on the link conditions and that the link conditions may be time varying, we let $\boldsymbol{S}(t)$ be the network *channel state*, i.e., the N-by-N matrix where the (n,m) component of $\boldsymbol{S}(t)$ denotes the channel condition between nodes n and m. We assume that $\boldsymbol{S}(t)$ takes values in some finite set $\mathcal{S} = (s_1, ..., s_M)$. We will assume in the following that the pair energy state (defined later) and $\boldsymbol{S}(t)$ is i.i.d. every slot. At every time slot, if $\boldsymbol{S}(t) = s_i$, then the power allocation vector $\boldsymbol{P}(t) = (P_{[n,m]}(t), [n,m] \in \mathcal{L})$, where $P_{[n,m]}(t)$ is the power allocated to link $[n,m]$ at time t, must be chosen from some feasible power allocation set $\mathcal{P}^{(s_i)}$. We assume that $\mathcal{P}^{(s_i)}$ is compact for all s_i, and that every power vector in $\mathcal{P}^{(s_i)}$ satisfies the constraint that for each node n, $0 \leq \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(t) \leq P_{max}$ for some $P_{max} < \infty$. Also, we assume that setting any $P_{[n,m]}$ in a vector $\boldsymbol{P} \in \mathcal{P}^{(s_i)}$ to zero yields another power vector that is still in $\mathcal{P}^{(s_i)}$. Given the channel state $\boldsymbol{S}(t)$ and the power allocation vector $\boldsymbol{P}(t)$, the transmission rate over the link $[n,m]$ is given by the rate-power function $\mu_{[n,m]}(t) = \mu_{[n,m]}(\boldsymbol{S}(t), \boldsymbol{P}(t))$. For each s_i, we assume that the function $\mu_{[n,m]}(s_i, \boldsymbol{P})$ satisfies the following properties:

PROPERTY 1. *For any $\boldsymbol{P}, \boldsymbol{P'} \in \mathcal{P}^{(s_i)}$, where $\boldsymbol{P'}$ is obtained by changing any single component $P_{[n,m]}$ in $\boldsymbol{P}$ to zero, we have for some finite constant $\delta > 0$ that:*

$$\mu_{[n,m]}(s_i, \boldsymbol{P}) \leq \mu_{[n,m]}(s_i, \boldsymbol{P'}) + \delta P_{[n,m]}. \tag{1}$$

PROPERTY 2. *If $\boldsymbol{P'}$ is obtained by setting the entry $P_{[n,b]}$ in $\boldsymbol{P}$ to zero, then:*

$$\mu_{[a,m]}(s_i, \boldsymbol{P}) \leq \mu_{[a,m]}(s_i, \boldsymbol{P'}),\ \forall\,[a,m] \neq [n,b]. \tag{2}$$

Property 1 states that the rate obtained over a link $[n,m]$ is upper bounded by some linear function of the power allocated to it, whereas Property 2 states that reducing the power over any link does not reduce the rate over any other links. We see that Property 1 and 2 can usually be satisfied by most rate-power functions, e.g., when the rate function is differentiable and has finite directional derivatives with respect to power [15], and the link rates do not improve with increased interference.

We also assume that there exists some finite constant μ_{max} such that $\mu_{[n,m]}(t) \leq \mu_{max}$ for all time under any power allocation vector and any channel state $\boldsymbol{S}(t)$. [2] In the following, we also use $\mu_{[n,b]}^{(c)}(t)$ to denote the rate allocated to the commodity c data over link $[n,b]$ at time t. It is easy to see that at any time t, we have:

$$\sum_c \mu_{[n,b]}^{(c)}(t) \leq \mu_{[n,b]}(t), \forall\ [n,b]. \tag{3}$$

2.3 The Energy Queue Model

We now specify the energy model. Every node in the network is assumed to be powered by a *finite* capacity energy storage device, e.g., a battery or ultra-capacitor [9]. We model such a device using an *energy queue*. We use the energy queue size at node n at time t, denoted by $E_n(t)$, to measure the amount of the energy left in the storage device at node n at time t. We assume that at every time, the nodes are capable of tracking its current energy level $E_n(t)$. In any time slot t, the power allocation vector $\boldsymbol{P}(t)$ must satisfy the following "energy-availability" constraint: [3]

$$\sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(t) \leq E_n(t), \quad \forall\ n. \tag{4}$$

That is, the consumed power must be no more than what is available. Each node in the network is assumed to be capable of harvesting energy from the environment, using, for instance, solar panels [9]. However, the amount of harvestable energy in a time slot is typically not fixed and varies over time. We use $h_n(t)$ to denote the amount of harvestable energy by node n at time t, and denote $\boldsymbol{h}(t) = (h_1(t), ..., h_N(t))$ the harvestable energy vector at time t, called the *energy state*. We assume that $\boldsymbol{h}(t)$ takes values in some finite set $\mathcal{H} = \{\boldsymbol{h}_1, ..., \boldsymbol{h}_K\}$. We assume that the pair $[\boldsymbol{h}(t), \boldsymbol{S}(t)]$ is i.i.d. over slots, with marginal distributions $\pi_{\boldsymbol{h}_i}$ and $\pi_{\boldsymbol{s}_j}$, respectively.

We assume that there exists $h_{max} < \infty$ such that $h_n(t) \leq h_{max}$ for all n, t. The energy harvested at time t is assumed to be available for use in time $t+1$. In the following, it is convenient for us to assume that each energy queue has infinite capacity, and that each node can decide whether or not to harvest energy on each slot. We model this harvesting decision by using $e_n(t) \in [0, h_n(t)]$ to denote the amount of energy that is actually harvested at time t. We will show later that our algorithm always harvests energy when the en-

[2]Note that in our transmission model, we did not explicitly take into account the reception power. We can easily incorporate that into our model at the expense of more complicated notations. In that case, our algorithm will also optimize over the reception power consumption, and the results in this paper still hold.

[3]We measure time in unit size "slots," so that our power $P_{[n,b]}(t)$ has units of energy/slot, and $P_{[n,b]}(t) \times (1\,\text{slot})$ is the resulting energy use in one slot. For simplicity, we suppress the implicit multiplication by 1 slot when converting between power and energy.

ergy queue is below a finite threshold of size $O(1/\epsilon)$ and drops it otherwise, thus can be implemented with finite capacity storage devices.

2.4 Queueing Dynamics

Let $\boldsymbol{Q}(t) = (Q_n^{(c)}(t), n, c \in \mathcal{N}),\ t = 0, 1, 2, ...$ be the data queue backlog vector in the network, where $Q_n^{(c)}(t)$ is the amount of commodity c data queued at node n. We assume the following queueing dynamics:

$$Q_n^{(c)}(t+1) \leq \big[Q_n^{(c)}(t) - \sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t)\big]^+ + \sum_{a\in\mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(t) + R_n^{(c)}(t), \quad (5)$$

with $Q_n^{(c)}(0) = 0$ for all $n, c \in \mathcal{N}$, $Q_c^{(c)}(t) = 0\ \forall\, t$, and $[x]^+ = \max[x, 0]$. The inequality in (5) is due to the fact that some nodes may not have enough commodity c packets to fill the allocated rates. In this paper, we say that the network is *stable* if the following is met:

$$\overline{\boldsymbol{Q}} \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{n,c}\mathbb{E}\big\{Q_n^{(c)}(\tau)\big\} < \infty. \quad (6)$$

Similarly, let $\boldsymbol{E}(t) = (E_n(t), n \in \mathcal{N})$ be the vector of the energy queue sizes. Due to the energy availability constraint (4), we see that for each node n, the energy queue $E_n(t)$ evolves according to the following: [4]

$$E_n(t+1) = E_n(t) - \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(t) + e_n(t), \quad (7)$$

with $E_n(0) = 0$ for all n. [5] Note again that by using the queueing dynamic (7), we start by assuming that each energy queue has infinite capacity. Later we will show that under our algorithms, all the $E_n(t)$ values are *determinstically* upper bounded, thus we only need a finite energy capacity in algorithm implementation.

2.5 Utility Maximization with Energy Management

The goal of the network is thus to design a joint flow control, routing and scheduling, and energy management algorithm that at every time slot, admits the right amount of data $R_n^{(c)}(t)$, chooses power allocation vector $\boldsymbol{P}(t)$ subject to (4), and transmits packets accordingly, so as to maximize the utility function:

$$U_{tot}(\overline{\boldsymbol{r}}) = \sum_{n,c} U_n^{(c)}(\overline{r}^{nc}), \quad (8)$$

subject to the network stability constraint (6). Here $\overline{\boldsymbol{r}} = (\overline{r}^{nc}, \forall\, n, c \in \mathcal{N})$ is the vector of the average expected admitted rates. Below, we will refer to this problem as the *Utility Maximization with Energy Management problem* (UMEM).

2.6 Discussion of the Model

(I) Our model is quite general and can be used to model many networks where nodes are powered by finite capacity batteries. For instance, a field monitoring sensor network [6], or many mobile ad hoc networks [21]. Also, our model allows the harvestable energy to be correlated among network nodes. This is particularly useful, as in practice, nodes that are collocated may have similar harvestable energy conditions.

(II) Although our model looks similar to the utility maximization model considered in [17] and [22], the problem considered in this paper is much more complicated. The main difficulty here is imposed by the constraint (4). Indeed, *(4) couples the current power allocation action and the future actions, in that a current action may cause the energy queue to be empty and hence block some power allocation actions in the future.* Problems involving such "no-underflow" constraints, e.g., [23], usually have to be modeled as dynamic programs (DP) [24]. However, DP typically suffers from a curse of dimensionality, and requires significant knowledge of the network probabilities. The work in [12] overcomes this "no-underflow" requirement by enforcing a positive drift constraint on the harvested energy and using Lyapunov optimization with this new constraint. Our approach is different and uses a modified Lyapunov function, which simplifies analysis and provides more explicit performance guarantees for the multi-hop case. Our MESA algorithm also fundamentally improves the resulting buffer size tradeoffs from $O(1/\epsilon)$ to $O([\log(1/\epsilon)]^2)$.

(III) Finally, note that although we assume $\boldsymbol{S}(t)$ and $\boldsymbol{h}(t)$ are i.i.d., we have extended our results to the case when $\boldsymbol{S}(t)$ and $\boldsymbol{h}(t)$ are Markovian in [25]. Moreover, our algorithm can also be shown to perform well under arbitrary $\boldsymbol{S}(t)$ and $\boldsymbol{h}(t)$ processes using the universal scheduling technique developed in [26].

3. UPPER BOUNDING THE OPTIMAL NETWORK UTILITY

In this section, we first obtain an upper bound on the optimal utility. This upper bound will be useful for our later analysis. The result is presented in the following theorem, in which we use $\boldsymbol{r}^*$ to denote the optimal solution of the UMEM problem, subject to the constraint that the network nodes are powered by finite capacity energy storage devices. The V parameter in the theorem can be any positive constant that is greater or equal to 1, and is included for our later analysis.

[4] Note that we do not explicitly consider energy leakage due to the imperfectness of the energy storage devices. This is a valid assumption if the rate of energy leakage is very small compared to the amount spent in each time slot.

[5] We can also pre-store energy in the energy queue and initialize $E_n(0)$ to any finite positive value up to its capacity. The results in the paper will not be affected.

Theorem 1. *The optimal network utility $U_{tot}(\boldsymbol{r}^*)$ satisfies: $VU_{tot}(\boldsymbol{r}^*) \leq \phi^*$, where ϕ^* is obtained over the class of stationary and randomized policies that have the following structure: allocate constant admission rates r^{nc} every slot; when $\boldsymbol{S}(t) = s_i$, choose a power vector $\boldsymbol{P}_k^{(s_i)}$ and allocate service rate $\mu_{[n,b]}^{(c)}(s_i, \boldsymbol{P}_k^{(s_i)})$ to node n with probability $\varrho_k^{(s_i)}$; and harvest energy $e_{n,k}^{(\boldsymbol{h}_i)}$ with probability $\varphi_k^{(\boldsymbol{h}_i)}$ when $\boldsymbol{h}(t) = \boldsymbol{h}_i$, subject to (3), (5) and (7), without regard to the energy availability constraint (4), to satisfy:*

$$\max: \ \phi = V \sum_{n,c} U_n^{(c)}(r^{nc}) \tag{9}$$

$$\text{s.t.} \quad r^{nc} + \mathbb{E}\big\{\sum_{k=1}^{K} \varrho_k^{(s_i)} \sum_{a\in\mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(s_i, \boldsymbol{P}_k^{(s_i)})\big\}$$

$$\leq \mathbb{E}\big\{\sum_{k=1}^{K} \varrho_k^{(s_i)} \sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(s_i, \boldsymbol{P}_k^{(s_i)})\big\}, \forall\,(n,c), \tag{10}$$

$$\mathbb{E}\big\{\sum_{k=1}^{K} \varrho_k^{(s_i)} \sum_{b\in\mathcal{N}_n^{(o)}} P_{k,[n,b]}^{(s_i)}\big\} = \mathbb{E}\big\{\sum_{k=1}^{K} \varphi_k^{(\boldsymbol{h}_i)} e_{n,k}^{(\boldsymbol{h}_i)}\big\}, \forall n, \tag{11}$$

$$\boldsymbol{P}_k^{(s_i)} \in \mathcal{P}^{(s_i)}, 0 \leq \varrho_k^{(s_i)}, \varphi_k^{(\boldsymbol{h}_i)} \leq 1, \forall s_i, k, \boldsymbol{h}_i,$$

$$\sum_{k=1}^{K} \varrho_k^{(s_i)} = 1, \sum_{k=1}^{K} \varphi_k^{(\boldsymbol{h}_i)} = 1, \forall s_i, \boldsymbol{h}_i,$$

$$0 \leq r^{nc} \leq R_{max}, \forall\,(n,c), 0 \leq e_{n,k}^{(\boldsymbol{h}_i)} \leq h_n^{(\boldsymbol{h}_i)}, \ \forall\, n, k, \boldsymbol{h}_i.$$

Here the expectation is taken over the random channel states s_i and energy states $\boldsymbol{h}_i$, and $K = N^2 + N + 2$. [6]

Proof. The proof argument is similar to the one used in [19], hence is omitted for brevity. □

In the theorem, (10) says that the rate of incoming data to node n is no more than the transmission rate out, and the equality constraint (11) says that the rate of harvested energy is equal to the energy consumption rate. We note that Theorem 1 indeed holds under more general ergodic $\boldsymbol{S}(t)$ and $\boldsymbol{h}(t)$ processes, e.g., when $\boldsymbol{S}(t)$ and $\boldsymbol{h}(t)$ evolve according to some finite state irreducible and aperiodic Markov chains.

4. ENGINEERING THE QUEUES

In this section, we present our Energy-limited Scheduling Algorithm (ESA) for the UMEM problem. ESA is designed based on the Lyapunov optimization technique developed in [19] and [17]. The idea of ESA is to construct a Lyapunov scheduling algorithm with *perturbed* weights for determining the energy harvesting, power allocation, routing and scheduling decisions. We will show that, by carefully perturbing the weights, one can ensure that whenever we allocate power to the links, there is always enough energy in the energy queues.

[6] The number K is due to the use of Caratheodory's Theorem in the proof argument used in [19].

4.1 The ESA Algorithm

To start, we first choose a *perturbation* vector $\boldsymbol{\theta} = (\theta_n, n \in \mathcal{N})$ (to be specified later). We then define a *perturbed* Lyapunov function as follows:

$$L(t) \triangleq \frac{1}{2} \sum_{n,c\in\mathcal{N}} \big[Q_n^{(c)}(t)\big]^2 + \frac{1}{2} \sum_{n\in\mathcal{N}} \big[E_n(t) - \theta_n\big]^2. \tag{12}$$

The intuition behind the use of the $\boldsymbol{\theta}$ vector is that: by keeping the Lyapunov function value small, we indeed "push" the $E_n(t)$ value towards θ_n. Thus by carefully choosing the value of θ_n, we can ensure that the energy queues always have enough energy for transmission.

Now denote $\boldsymbol{Z}(t) = (\boldsymbol{Q}(t), \boldsymbol{E}(t))$, and define a one-slot conditional Lyapunov drift as follows:

$$\Delta(t) \triangleq \mathbb{E}\big\{L(t+1) - L(t) \mid \boldsymbol{Z}(t)\big\}. \tag{13}$$

Here the expectation is taken over the randomness of the channel state and the energy state, as well as the randomness in choosing the data admission action, the power allocation action, the routing and scheduling action, and the energy harvesting action. For notation simplicity, we also define:

$$\Delta_V(t) \triangleq \Delta(t) - V\mathbb{E}\big\{\sum_{n,c} U_n^{(c)}(R_n^{(c)}(t)) \mid \boldsymbol{Z}(t)\big\}. \tag{14}$$

We have the following lemma regarding the drift:

Lemma 1. *Under any feasible data admission action, power allocation action, routing and scheduling action, and energy harvesting action that can be implemented at time t, we have:*

$$\begin{aligned}\Delta_V(t) \leq B &+ \sum_{n\in\mathcal{N}} (E_n(t) - \theta_n)\mathbb{E}\big\{e_n(t) \mid \boldsymbol{Z}(t)\big\} \\ &- \mathbb{E}\big\{\sum_{n,c} \big[VU_n^{(c)}(R_n^{(c)}(t)) - Q_n^{(c)}(t)R_n^{(c)}(t)\big] \mid \boldsymbol{Z}(t)\big\} \\ &- \mathbb{E}\Big\{\sum_{n} \Big[\sum_{c} \sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t)\big[Q_n^{(c)}(t) - Q_b^{(c)}(t)\big] \\ &\qquad + (E_n(t) - \theta_n) \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(t)\Big] \mid \boldsymbol{Z}(t)\Big\}.\end{aligned} \tag{15}$$

Here $B = N^2(\frac{3}{2}d_{max}^2\mu_{max}^2 + R_{max}^2) + \frac{N}{2}(P_{max} + h_{max})^2$, and d_{max} is defined in Section 2 as the maximum in-degree of any node in the network.

Proof. See Appendix A. □

We now present the ESA algorithm. The idea of the algorithm is to approximately minimize the right-hand side (RHS) of (15) subject to the energy-availability constraint (4). In ESA, we use a parameter $\gamma \triangleq R_{max} + d_{max}\mu_{max}$, which is used in the link weight definition to allow deterministic upper bounds on queue sizes.

Energy-limited Scheduling Algorithm (ESA): Initialize $\boldsymbol{\theta}$. At every slot, observe $\boldsymbol{Q}(t)$, $\boldsymbol{E}(t)$, $\boldsymbol{S}(t)$, and do:

- Energy Harvesting: At time t, if $E_n(t) - \theta_n < 0$, perform energy harvesting and store the harvested energy, i.e., $e_n(t) = h_n(t)$. Else set $e_n(t) = 0$. Note that this decision on $e_n(t)$ indeed minimizes the $(E_n(t) - \theta_n)\mathbb{E}\{e_n(t) \mid \boldsymbol{Z}(t)\}$ term in (15).
- Data Admission: At every time t, choose $R_n^{(c)}(t)$ to be the optimal solution of the following optimization problem:

$$\max: \ VU_n^{(c)}(r) - Q_n^{(c)}(t)r, \ \ s.t. \ 0 \leq r \leq R_{max}. \quad (16)$$

  Note that this decision minimizes the terms involving $R_n^{(c)}(t)$ in the RHS of (15).
- Power Allocation: At every time t, define the weight of the commodity c data over link $[n, b]$ as:

$$W_{[n,b]}^{(c)}(t) \triangleq \big[Q_n^{(c)}(t) - Q_b^{(c)}(t) - \gamma\big]^+. \quad (17)$$

  Then define the link weight $W_{[n,b]}(t) = \max_c W_{[n,b]}^{(c)}(t)$, and choose $\boldsymbol{P}(t) \in \mathcal{P}^{(s_i)}$ to maximize:

$$G(\boldsymbol{P}(t)) \triangleq \sum_n \bigg[\sum_{b \in \mathcal{N}_n^{(o)}} \mu_{[n,b]}(t) W_{[n,b]}(t) \quad (18)$$

$$+ (E_n(t) - \theta_n) \sum_{b \in \mathcal{N}_n^{(o)}} P_{[n,b]}(t) \bigg],$$

  subject to the energy availability constraint (4).
- Routing and Scheduling: For every node n, find any $c^* \in \text{argmax}_c W_{[n,b]}^{(c)}(t)$. If $W_{[n,b]}^{(c^*)}(t) > 0$, set:

$$\mu_{[n,b]}^{(c^*)}(t) = \mu_{[n,b]}(t), \quad (19)$$

  that is, allocate the full rate over the link $[n, b]$ to any commodity that achieves the maximum positive weight over the link. Use idle-fill if needed. If $W_{[n,b]}^{(c^*)}(t) = 0$, we set $\mu_{[n,b]}^{(c)}(t) = 0$ for all c over link $[n, b]$. [7]
- Queue Update: Update $Q_n^{(c)}(t)$ and $E_n(t)$ according to the dynamics (5) and (7), respectively.

The combined Power Allocation and Routing Scheduling step would have minimized the terms involving $\mu_{[n,b]}^{(c)}(t)$ and $\boldsymbol{P}(t)$ in the RHS of (15) if we had defined $\gamma = 0$. However, we have included a non-zero γ in the differential backlog definition (17), resulting in a decision that comes within an *additive constant* of minimizing the RHS of (15). The advantage of using this γ is that it leads to a deterministic bound on all queue sizes, as we show in the next section.

Note that in the energy harvesting step of ESA, node n will perform energy harvesting only when the energy volume is less than θ_n. This feature is very important because it allows one to implement ESA with finite energy storage capacity. More importantly, we will show that it provides us with a very easy way to size our energy storage devices if we want to achieve a utility that is within $O(\epsilon)$ of the optimal, i.e., use energy storage devices of size $O(1/\epsilon)$. In practice, once the right energy storage capacity is determined, we can always modify ESA by having the nodes perform energy harvesting in every time slot, in which case nodes always have more energy than that under ESA and the same utility performance can be achieved.

[7] Note that we will still use the same power allocation $P_{[n,b]}(t)$ (can be nonzero) in this case, although all the rates $\mu_{[n,b]}^{(c)}(t)$ are zero. We will show that doing this still yields performance that can be pushed arbitrarily close to optimal. In the actual implementation, however, we can always save the power $P_{[n,b]}(t)$ when $\mu_{[n,b]}^{(c)}(t) = 0 \ \forall c$. Similar performance results can also be obtained.

4.2 Implementation of ESA

(I) First we note that ESA only requires the knowledge of the *instant* channel state $\boldsymbol{S}(t)$ and the queue sizes $\boldsymbol{Q}(t)$ and $\boldsymbol{E}(t)$. *It does not even require any knowledge of the energy state process* $\boldsymbol{h}(t)$. This is very useful in practice when the knowledge of the energy source is difficult to obtain. ESA is also very different from previous algorithms for energy harvesting network, e.g., [9] [10], where statistical knowledge of the energy source is often required.

(II) Note that the implementation of ESA involves maximizing (18). Thus ESA's complexity is the same as the widely used max-weight algorithms, which in general requires centralized control and can be NP-hard [17]. However, in cases when the links do not interfere with each other, ESA can easily be implemented in a distributed manner, where each node only has to know about the queue sizes at its neighbor nodes and can decide on the power allocation locally. Moreover, one can look for constant factor approximation solutions of (18), e.g., [27] and Section 4.7 and 5.2.1 in [17]. Such approximation results can usually be found in a distributed manner in polynomial time, and ESA can be shown to achieve a utility that is at least a constant factor of $U_{tot}(\boldsymbol{r}^*)$ under these solutions.

5. PERFORMANCE ANALYSIS

We now present the performance results of the ESA algorithm in the following theorem. Note that we have also extended the theorem to the case when $\boldsymbol{S}(t)$ and $\boldsymbol{h}(t)$ evolve according to general finite state irreducible and aperiodic Markov chains in [25].

Theorem 2. *Under the ESA algorithm with $\theta_n \triangleq \delta\beta V + P_{max}$, $\forall\ n$, we have the following:*

(a) The data queues and the energy queues satisfy the following for all time under any arbitrary $\boldsymbol{S}(t)$ and $\boldsymbol{h}(t)$ processes:

$$0 \leq Q_n^{(c)}(t) \leq \beta V + R_{max}, \ \ \forall\ (n, c), \quad (20)$$

$$0 \leq E_n(t) \leq \theta_n + h_{max}, \ \ \forall\ n. \quad (21)$$

Moreover, when a node n allocates nonzero power to any of its outgoing links, $E_n(t) \geq P_{max}$.

(b) Let $\overline{\boldsymbol{r}} = (\overline{r}^{nc}, \forall\,(n,c))$ be the time average admitted rate vector achieved by ESA, then:

$$U_{tot}(\overline{\boldsymbol{r}}) = \sum_{n,c} U_n^{(c)}(\overline{r}^{nc}) \geq U_{tot}(\boldsymbol{r}^*) - \frac{\tilde{B}}{V}, \quad (22)$$

where $\boldsymbol{r}^*$ is an optimal solution of the UMEM problem, and $\tilde{B} = B + N^2\gamma d_{max}\mu_{max} = \Theta(1)$, i.e., independent of V.

Proof. See Appendix B. $\square$

We note the following of Theorem 2: (I) Part (a) is a *sample path* result. Hence it holds even under *non-stationary* $\boldsymbol{S}(t)$ and $\boldsymbol{h}(t)$ processes. (II) By taking $\epsilon = 1/V$, Part (a) implies that the average data queue size is $O(1/\epsilon)$. Combining this with Part (b), we see that that ESA achieves an $[O(\epsilon), O(1/\epsilon)]$ utility-backlog tradeoff for the UMEM problem. (III) we see from Part (a) that the energy queue size is deterministically upper bounded by a constant of size $O(1/\epsilon)$. This provides an explicit characterization of the size of the energy storage device needed for achieving the desired utility performance. Such explicit bounds are particularly useful for system deployments.

6. REDUCING THE BUFFER SIZE

In this section, we show that it is possible to achieve the same $O(\epsilon)$ close-to-optimal utility performance guarantee using energy storage devices with only $O([\log(\frac{1}{\epsilon})]^2)$ sizes, while guaranteeing a much smaller average data queue size, i.e., $O([\log(1/\epsilon)]^2)$. Our algorithm is motivated by the "exponential attraction" result developed in [22], which states that the probability for the network backlog vector to deviate from some fixed point typically decreases exponentially with the deviation distance. Using this result, we will develop the Modified -ESA (MESA) algorithm. We emphasize that, although MESA looks similar to the algorithms developed in [22], it only uses *finite* energy storage capacities. This feature makes it very different and requires a new analysis for its performance (see [25] for the analysis of MESA).

To start, for a given ϵ, we let $V = 1/\epsilon$, and define $M = 4[\log(V)]^2$. We then associate with each node n a *virtual* energy queue process $\hat{E}_n(t)$ and a set of *virtual* data queues $\hat{Q}_n^{(c)}(t)\ \forall\, c$. We also associate with each node n an *actual* energy queue with size M. We assume that V is chosen to be such that $\frac{M}{2} > \alpha_{max} \triangleq \max[P_{max}, h_{max}]$. MESA consists of two phases: Phase I runs the system using the virtual queue processes, to discover the "attraction point" values of the queues (as explained below). Phase II then uses these values to carefully perform the actions so as to ensure energy availability and reduce network delay.

Modified-ESA (MESA): Initialize $\boldsymbol{\theta}$. Perform:

- Phase I: Choose a sufficiently large T. From time $t = 0, ..., T$, run ESA using $\hat{\boldsymbol{Q}}(t)$ and $\hat{\boldsymbol{E}}(t)$ as the data and energy queues. Obtain the two vectors $\boldsymbol{\mathcal{Q}} = (\mathcal{Q}_n^{(c)}, \forall\,(n,c))$ and $\boldsymbol{\mathcal{E}} = (\mathcal{E}_n, \forall\, n)$ by having: $\mathcal{Q}_n^{(c)} = [\hat{Q}_n^{(c)}(T) - \frac{M}{2}]^+$ and $\mathcal{E}_n = [\hat{E}_n(T) - \frac{M}{2}]^+$.
- Phase II: Reset $t = 0$. Initialize $\hat{\boldsymbol{E}}(0) = \boldsymbol{\mathcal{E}}$ and $\hat{\boldsymbol{Q}}(0) = \boldsymbol{\mathcal{Q}}$. Also set $\boldsymbol{Q}(0) = \boldsymbol{0}$ and $\boldsymbol{E}(0) = \boldsymbol{0}$. In every time slot, first run the ESA algorithm based on $\hat{\boldsymbol{Q}}(t)$, $\hat{\boldsymbol{E}}(t)$, and $\boldsymbol{S}(t)$, to obtain the action variables, i.e., the corresponding $e_n(t)$, $R_n^{(c)}(t)$, and $\mu_{[n,b]}^{(c)}(t)$ values. Perform Data Admisson, Power Allocation, and Routing and Scheduling exactly as ESA, plus the following:
  - Energy harvesting: If $\hat{E}_n(t) < \mathcal{E}_n$, let $\tilde{e}_n(t) = [e_n(t) - (\mathcal{E}_n - \hat{E}_n(t))]^+$. Harvest $\tilde{e}(t)$ amount of energy, i.e., update $E_n(t)$ as follows:

$$E_n(t+1) = \big([E_n(t) - \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(t)]^+ + \tilde{e}_n(t)\big) \wedge M.$$

    Here $a \wedge b = \min[a, b]$. Else if $\hat{E}_n(t) > \mathcal{E}_n + M$, do not spend any power and update $E_n(t)$ according to:

$$E_n(t+1) = \min\big[E_n(t) + e_n(t), M\big].$$

    Else update $E_n(t)$ according to:

$$E_n(t+1) = \big([E_n(t) - \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(t)]^+ + e_n(t)\big) \wedge M.$$

  - Packet Dropping: For any node n with $\hat{E}_n(t) < \mathcal{E}_n + P_{max}$ or $\hat{E}_n(t) > \mathcal{E}_n + M$, drop all the packets that should have been transmitted, i.e., change the input into any $Q_n^{(c)}(t)$ to:

$$A_n^{(c)}(t) = R_n^{(c)}(t) + \sum_{a\in\mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(t) 1_{[F_a(t)]}.$$

    Here $1_{[\cdot]}$ is the indicator function and $F_a(t)$ is the event that $\hat{E}_a(t) \in [\mathcal{E}_a + P_{max}, \mathcal{E}_a + M]$. Then further modify the routing and scheduling action under ESA as follows:
    - If $\hat{Q}_n^{(c)}(t) < \mathcal{Q}_n^{(c)}$, let $\tilde{A}_n^{(c)}(t) = \big[A_n^{(c)}(t) - [\mathcal{Q}_n^{(c)} - \hat{Q}_n^{(c)}(t)]\big]^+$, update $Q_n^{(c)}(t)$ by:

$$Q_n^{(c)}(t+1) \leq \big[Q_n^{(c)}(t) - \sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t)\big]^+ + \tilde{A}_n^{(c)}(t).$$

    - If $\hat{Q}_n^{(c)}(t) \geq \mathcal{Q}_n^{(c)}$, update $Q_n^{(c)}(t)$ by:

$$Q_n^{(c)}(t+1) \leq \big[Q_n^{(c)}(t) - \sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t)\big]^+ + A_n^{(c)}(t).$$

  - Update $\hat{\boldsymbol{E}}(t)$ and $\hat{\boldsymbol{Q}}(t)$ using (7) and (5).

Note here we have used the $[\cdot]^+$ operator for updating $E_n(t)$ in the energy harvesting part. This is due to the fact that the power allocation decisions are now made *based on* $\hat{\boldsymbol{E}}(t)$ *but not* $\boldsymbol{E}(t)$. If $\hat{E}_n(t)$ never gets below $\mathcal{E}_n$ or above $\mathcal{E}_n + M$, then we always have $E_n(t) = \hat{E}_n(t) - \mathcal{E}_n$. Similarly, if $\hat{Q}_n^{(c)}(t)$ is always above $\mathcal{Q}_n^{(c)}$ and $\hat{E}_n(t)$ is always in $[\mathcal{E}_n + P_{max}, \mathcal{E}_n + M]$, then we always have $Q_n^{(c)}(t) = \hat{Q}_n^{(c)}(t) - \mathcal{Q}_n^{(c)}$. MESA is designed to ensure that $\hat{Q}_n^{(c)}(t)$ and $\hat{E}_n(t)$ mostly stay in these "right" ranges. We now summarize the performance results of MESA in the following theorem. In the theorem, we use $g(\boldsymbol{\upsilon}, \boldsymbol{\nu})$ to denote the dual function of the problem (9), which is shown in [25] to be:

$$g(\boldsymbol{\upsilon}, \boldsymbol{\nu}) = \sup_{r^{nc}, \boldsymbol{P}^{(s_i)}, e_n^{(\boldsymbol{h}_j)}} \sum_{s_i} \pi_{s_i} \sum_{\boldsymbol{h}_j} \pi_{\boldsymbol{h}_j} \Big\{ V \sum_{n,c} U_n^{(c)}(r^{nc}) \\ - \sum_n \upsilon_n^{(c)} \big[r^{nc} + \sum_{a \in \mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(s_i, \boldsymbol{P}^{(s_i)}) \\ - \sum_{b \in \mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(s_i, \boldsymbol{P}^{(s_i)}) \big] - \sum_n \nu_n \big[\sum_{b \in \mathcal{N}_n^{(o)}} P_{[n,b]}^{(s_i)} - e_n^{(\boldsymbol{h}_j)} \big] \Big\}. \quad (23)$$

We also write $g(\boldsymbol{\upsilon}, \boldsymbol{\nu})$ as a function of $\boldsymbol{y} = (\boldsymbol{\upsilon}, \boldsymbol{\nu})$ and use $\boldsymbol{y}^*$ to denote an optimal solution of $g(\boldsymbol{y})$.

Theorem 3. *Suppose that $\boldsymbol{y}^* = (\boldsymbol{\upsilon}^*, \boldsymbol{\nu}^*)$ is finite and unique, that $\boldsymbol{\theta}$ is chosen such that $\theta_n + \nu_n^* > 0$, $\forall\ n$, and that for all $\boldsymbol{y} = (\boldsymbol{\upsilon}, \boldsymbol{\nu})$ with $\boldsymbol{\upsilon} \succeq \boldsymbol{0}, \boldsymbol{\nu} \in \mathbb{R}^N$, the dual function $g(\boldsymbol{y})$ satisfies:*

$$g(\boldsymbol{y}^*) \geq g(\boldsymbol{y}) + L||\boldsymbol{y}^* - \boldsymbol{y}||, \quad (24)$$

for some constant $L > 0$ independent of V, that the system is in steady state at time T, and that a steady state distribution for the queues exists under ESA. Then under MESA with a sufficiently large V, with probability $1 - O(\frac{1}{V^4})$, we have:

$$\overline{\boldsymbol{Q}} \leq O([\log(V)]^2), \quad (25)$$

$$U_{tot}(\overline{\boldsymbol{r}}) \geq U_{tot}(\boldsymbol{r}^*) - O(1/V). \quad (26)$$

Furthermore, the fraction of packets dropped in the packet dropping step is $O(\frac{1}{V^{\log(V)}})$.

Proof. See [25]. $\square$

Note that (24) is indeed the condition needed for proving the exponential attraction result in [22]. It has been observed, e.g., in [22] that (24) typically holds in practice, particularly when the network action set is finite, in which case the dual function $g(\boldsymbol{y})$ is polyhedral in $\boldsymbol{y}$ (see [22] for more discussions). Theorem 3 then shows that under this condition, one can significantly reduce the energy capacity needed to achieve the $O(\epsilon)$ close-to-optimal utility performance and greatly reduce the network congestion.

7. SIMULATION

In this section we provide simulation results of our algorithms. We consider a data collection network shown in Fig. 1. Such a network typically appears in the sensor network scenario where sensors are used to sense data and forward them to the sink. In this network, there are 6 nodes. The node S represents the sink node, the nodes $1, 2, 3$ sense data and deliver them to node S via the relay of nodes $4, 5$.

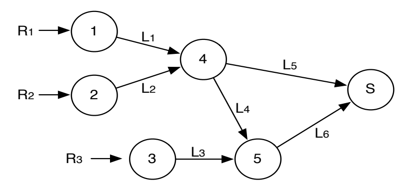

Figure 1: A data collection network.

The channel state of each communication link, represented by a directed edge, is i.i.d. every time slot and can be either "G=Good" or "B=Bad" with equal probabilities. One unit of power can serve two packets over a link when the channel state is good, but can only serve one when the channel is bad. We assume $R_{max} = 3$ and the utility functions are given by: $U_1^{(S)}(r) = U_2^{(S)}(r) = U_3^{(S)}(r) = \log(1 + r)$ and $U_4^{(S)}(r) = U_5^{(S)}(r) = 0$. For simplicity, we also assume that all the links do not interfere with each other. We assume that for each node, the available energy $h_n(t)$ is i.i.d. and $h_n(t) = 2/0$ with equal probabilities.

It is easy to see that in this case, we can use $\beta = 1$, $\delta = 2$, $\mu_{max} = 2$, $d_{max} = 2$, $P_{max} = 2$, and $\gamma = d_{max}\mu_{max} + R_{max} = 7$. Using Theorem 2, we set $\theta_n = \delta\beta V + P_{max} = 2V + 2$. We simulate $V \in \{20, 30, 40, 50, 80, 100, 200\}$. Each simulation is run for 10^6 slots. The simulation results are plotted in Fig. 2. We see that the total network utility converges quickly to very close to the optimal value, which can be shown to be roughly 2.03, and that the average data queue size and the average energy queue size both grow linearly in V.

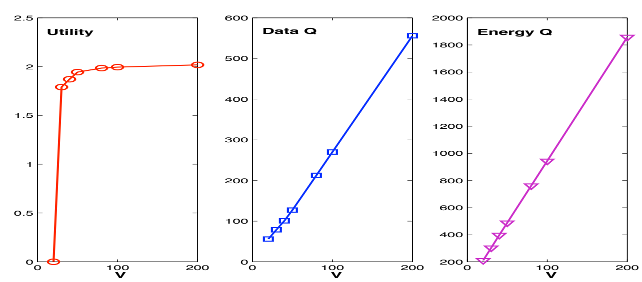

Figure 2: Simulation results of ESA.

Fig. 3 also shows two sample-path data queue processes and two energy queue processes under $V = 100$. It can be verified that all the queue sizes satisfy the queueing bounds in Theorem 2. We also observe the "exponential attraction" behavior of the queues, as shown in [22]. However, different from the simulation results in previous works, e.g., [22], we see that the queue size of $Q_1^{(S)}(t)$ does not approach the fixed point from below. It instead first has a "burst" in the early time slots. This is due to the fact that the system "waits" for $E_1(t)$ to come close enough to its fixed point. Such an effect can be mitigated by storing an initial energy of size θ in the energy queue.

We also simulate the MESA algorithm for the same

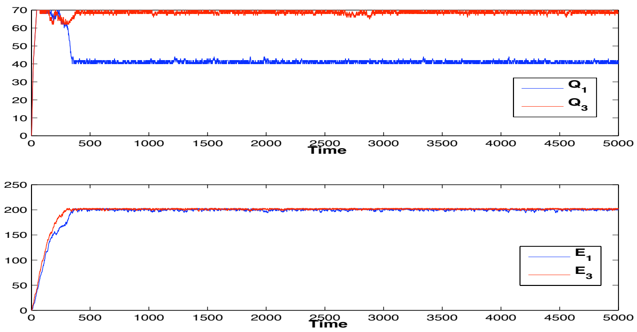

Figure 3: Sample path queue processes.

network with the same $\boldsymbol{\theta}$ vector. We use $T = 50V$ in Phase I for obtaining the vectors $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\mathcal{Q}}$. Fig. 4 plots the performance results. We observe that no packet was dropped throughout the simulations under any V values. The utility again quickly converges to the optimal as V increases. We also see from the second and third plots that the actual queues only grow polylogarithmically in V, i.e., $O([\log(V)]^2)$, while the virtual queues, which are the same as the actual queues under ESA, grows linearly in V. This shows a good match between the simulations and Theorem 3.

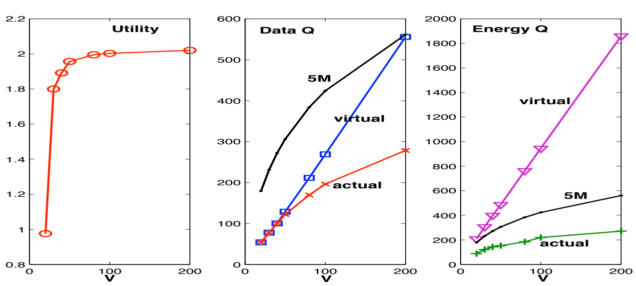

Figure 4: Simulation results of MESA. $5M$ is the total network energy buffer size.

Appendix A – Proof of Lemma 1

Here we prove Lemma 1.

Proof. First by squaring both sides of (5), and using the fact that for any $x \in \mathbb{R}$, $([x]^+)^2 \leq x^2$, we have:

$$[Q_n^{(c)}(t+1)]^2 - [Q_n^{(c)}(t)]^2 \tag{27}$$
$$\leq [\sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t)]^2 + [\sum_{a\in\mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(t) + R_n^{(c)}(t)]^2$$
$$-2Q_n^{(c)}(t)[\sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t) - \sum_{a\in\mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(t) - R_n^{(c)}(t)].$$

Multiplying both sides by $\frac{1}{2}$, and defining $\hat{B} = \frac{3}{2}d_{max}^2\mu_{max}^2 + R_{max}^2$, we have:

$$\frac{1}{2}([Q_n^{(c)}(t+1)]^2 - [Q_n^{(c)}(t)]^2) \leq \hat{B} \tag{28}$$
$$-Q_n^{(c)}(t)[\sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t) - \sum_{a\in\mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(t) - R_n^{(c)}(t)].$$

Using a similar approach, we get that:

$$\frac{1}{2}([E_n(t+1) - \theta_n]^2 - [E_n(t) - \theta_n]^2) \tag{29}$$
$$\leq \hat{B}' - [E_n(t) - \theta_n][\sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(t) - e_n(t)],$$

where $\hat{B}' = \frac{1}{2}(P_{max} + h_{max})^2$. Now by summing (28) over all (n, c) and (29) over all n, and by defining $B = N^2\hat{B} + N\hat{B}' = N^2(\frac{3}{2}d_{max}^2\mu_{max}^2 + R_{max}^2) + \frac{1}{2}N(P_{max} + h_{max})^2$, we have:

$$L(t+1) - L(t) \leq B - \sum_{n,c} Q_n^{(c)}(t)[\sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t)$$
$$- \sum_{a\in\mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(t) - R_n^{(c)}(t)]$$
$$-\sum_n [E_n(t) - \theta_n][\sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(t) - e_n(t)].$$

Taking expectations on both sides over the random channel and energy states and the randomness over actions conditioning on $\boldsymbol{Z}(t)$, subtracting from both sides the term $V\mathbb{E}\{\sum_{n,c} U_n^{(c)}(R_n^{(c)}(t)) \mid \boldsymbol{Z}(t)\}$, and rearranging the terms, we see that the lemma follows. $\square$

Appendix B – Proof of Theorem 2

Here we prove Theorem 2.

Proof. (Part (a)) We first prove (20) using a similar argument as in [15]. It is easy to see that it holds for $t = 0$, since $Q_n^{(c)}(0) = 0$ for all (n, c). Now assume that $Q_n^{(c)}(t) \leq \beta V + R_{max}$ for all (n, c) at t, we want to show that it holds for time $t + 1$. First, if node n does not receive any new commodity c data, then $Q_n^{(c)}(t) \leq Q_n^{(c)}(t+1) \leq \beta V + R_{max}$. Second, if node n receives endogenous commodity c data from any other node b, then we must have:

$$Q_n^{(c)}(t) \leq Q_b^{(c)}(t) - \gamma \leq \beta V + R_{max} - \gamma.$$

However, since any node can receive at most γ commodity c packets, we have $Q_n^{(c)}(t+1) \leq \beta V + R_{max}$. Finally, if node n receives exogenous packets from outside the network, then according to (16), we must have $Q_n^{(c)}(t) \leq \beta V$. Hence $Q_n^{(c)}(t+1) \leq \beta V + R_{max}$.

Now it is also easy to see from the energy storage part of ESA that $E_n(t) \leq \theta_n + h_{max}$, which proves (21).

We now show that if $E_n(t) < P_{max}$, then $G(t)$ will be maximized by choosing $P_{[n,m]}(t) = 0$ for all $m \in \mathcal{N}_n^{(o)}$ at node n. To see this, first note that since all the data queues are upper bounded by $\beta V + R_{max}$, we have: $W_{[n,b]}(t) \leq \beta V - d_{max}\mu_{max}$ for all $[n, b]$ and for all time.

Now let the power vector that maximizes $G(t)$ be $\boldsymbol{P}^*$ and assume that there exists some $P_{[n,m]}^*$ that is positive. We now create a new power allocation vector $\boldsymbol{P}$ by setting only $P_{[n,m]}^* = 0$ in $\boldsymbol{P}^*$. Then we have the following, in which we have written $\mu_{[n,m]}(\boldsymbol{S}(t), \boldsymbol{P}(t))$ only as a function of $\boldsymbol{P}(t)$ to simplify notation:

$$\begin{aligned} & G(\boldsymbol{P}^*) - G(\boldsymbol{P}) \\ = & \sum_n \sum_{b\in\mathcal{N}_n^{(o)}} \big[\mu_{[n,b]}(\boldsymbol{P}^*) - \mu_{[n,b]}(\boldsymbol{P})\big] W_{[n,b]}(t) \\ & \qquad\qquad + (E_n(t) - \theta_n) P^*_{[n,m]} \\ \leq & \big(\mu_{[n,m]}(\boldsymbol{P}^*) - \mu_{[n,m]}(\boldsymbol{P})\big) W_{[n,m]}(t) + (E_n(t) - \theta_n) P^*_{[n,m]}. \end{aligned}$$

Here in the last step we have used (2) in Property 2 of $\mu_{[n,m]}(\cdot, \boldsymbol{P})$, which implies that $\mu_{[n,b]}(\boldsymbol{P}^*) - \mu_{[n,b]}(\boldsymbol{P}) \leq 0$ for all $b \neq m$. Now suppose $E_n(t) < P_{max}$. We see then $E_n(t) - \theta_n < -\delta\beta V$. Using Property 1 and the fact that $W_{[n,m]}(t) \leq \beta V - d_{max}\mu_{max}$, the above implies:

$$\begin{aligned} G(\boldsymbol{P}^*) - G(\boldsymbol{P}) &< (\beta V - d_{max}\mu_{max})\delta P^*_{[n,m]} - \delta\beta V P^*_{[n,m]} \\ &< 0. \end{aligned}$$

This shows that $\boldsymbol{P}^*$ cannot have been the power vector that maximizes $G(t)$ if $E_n(t) < P_{max}$. Therefore $E_n(t) \geq P_{max}$ whenever node n allocates any nonzero power over any of its outgoing links. Hence all the power allocation decisions are feasible. This shows that the constraint (4) is indeed *redundant* in ESA and completes the proof of Part (a).

(Part (b)) We now prove Part (b). We first show that ESA approximately minimizes the RHS of (15). To see this, note from Part (A) that ESA indeed minimizes the following function at time t:

$$\begin{aligned} D(t) = & \sum_{n\in\mathcal{N}} (E_n(t) - \theta_n) e_n(t) \\ & - \sum_{n,c\in\mathcal{N}} \big[V U_n^{(c)}(R_n^{(c)}(t)) - Q_n^{(c)}(t) R_n^{(c)}(t)\big] \\ & - \sum_{n\in\mathcal{N}} \bigg[\sum_c \sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t) \big[Q_n^{(c)}(t) - Q_b^{(c)}(t) - \gamma\big] \\ & \qquad + (E_n(t) - \theta_n) \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(t)\bigg], \end{aligned} \tag{30}$$

subject to only the constraints: $e_n(t) \in [0, h_n(t)]$, $R_n^{(c)}(t) \in [0, R_{max}]$, $\boldsymbol{P}(t) \in \mathcal{P}^{(s_i)}$ and (3), i.e., without the energy-availability constraint (4). Now define $\tilde{D}(t)$ as follows:

$$\begin{aligned} \tilde{D}(t) = & \sum_{n\in\mathcal{N}} (E_n(t) - \theta_n) e_n(t) \\ & - \sum_{n,c\in\mathcal{N}} \big[V U_n^{(c)}(R_n^{(c)}(t)) - Q_n^{(c)}(t) R_n^{(c)}(t)\big] \\ & - \sum_{n\in\mathcal{N}} \bigg[\sum_c \sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t) \big[Q_n^{(c)}(t) - Q_b^{(c)}(t)\big] \\ & \qquad + (E_n(t) - \theta_n) \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(t)\bigg]. \end{aligned} \tag{31}$$

Note that $\tilde{D}(t)$ is indeed the function inside the expectation on the RHS of the drift bound (15). It is easy to see from the above that:

$$D(t) = \tilde{D}(t) + \sum_n \sum_c \sum_{[n,b]\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t)\gamma.$$

Since ESA minimizes $D(t)$, we see that:

$$\begin{aligned} & \tilde{D}^E(t) + \sum_n \sum_c \sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)E}(t)\gamma \\ & \qquad \leq \tilde{D}^{ALT}(t) + \sum_n \sum_c \sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)ALT}(t)\gamma, \end{aligned}$$

where the superscript E represents the ESA algorithm, and ALT represents any other alternate policy. Since

$$0 \leq \sum_n \sum_c \sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t)\gamma \leq N^2\gamma d_{max}\mu_{max},$$

we have:

$$\tilde{D}^E(t) \leq \tilde{D}^{ALT}(t) + N^2\gamma d_{max}\mu_{max}. \tag{32}$$

That is, the value of $\tilde{D}(t)$ under ESA is no greater than its value under *any* other alternative policy plus a constant, including the ones that ignore the energy availability constraint (4). Further, Part (a) shows that the energy availability constraint (4) is naturally satisfied under ESA without explicitly being enforced. Now using the definition of $\tilde{D}(t)$, (15) can be rewritten as:

$$\begin{aligned} & \Delta(t) - V\mathbb{E}\big\{\sum_{n,c} U_n^{(c)}(R_n^{(c)}(t)) \mid \boldsymbol{Z}(t)\big\} \\ & \qquad\qquad \leq B + \mathbb{E}\big\{\tilde{D}^E(t) \mid \boldsymbol{Z}(t)\big\}. \end{aligned}$$

Using (32), we get:

$$\begin{aligned} & \Delta(t) - V\mathbb{E}\big\{\sum_{n,c} U_n^{(c)}(R_n^{(c)}(t)) \mid \boldsymbol{Z}(t)\big\} \\ & \qquad\qquad \leq \tilde{B} + \mathbb{E}\big\{\tilde{D}^{ALT}(t) \mid \boldsymbol{Z}(t)\big\}, \end{aligned} \tag{33}$$

where $\tilde{B} = B + N^2\gamma d_{max}\mu_{max}$. Now plugging into (33) the policy in Theorem 1, which by comparing (9) and (31) can easily be shown to result in $\mathbb{E}\big\{\tilde{D}^{ALT}(t) \mid \boldsymbol{Z}(t)\big\} = \phi^*$, and using the fact that $\phi^* \geq VU_{tot}(\boldsymbol{r}^*)$, we have:

$$\Delta(t) - V\mathbb{E}\big\{\sum_{n,c} U_n^{(c)}(R_n^{(c)}(t)) \mid \boldsymbol{Z}(t)\big\} \leq \tilde{B} - VU_{tot}(\boldsymbol{r}^*).$$

Taking expectations over $\boldsymbol{Z}(t)$ and summing the above over $t = 0, ..., T-1$, we have:

$$\begin{aligned} & \mathbb{E}\big\{L(T) - L(0)\big\} - V\sum_{t=0}^{T-1} \mathbb{E}\big\{\sum_{n,c} U_n^{(c)}(R_n^{(c)}(t))\big\} \\ & \qquad\qquad \leq T\tilde{B} - TVU_{tot}(\boldsymbol{r}^*). \end{aligned}$$

Rearranging the terms, using the facts that $L(t) \geq 0$ and $L(0) = 0$, dividing both sides by VT, and taking the liminf as $T \to \infty$, we get:

$$\liminf_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\big\{\sum_{n,c} U_n^{(c)}(R_n^{(c)}(t))\big\} \geq U_{tot}(\boldsymbol{r}^*) - \tilde{B}/V.$$

Using Jensen's inequality, we see that:

$$\sum_{n,c} U_n^{(c)}\big(\liminf_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big\{R_n^{(c)}(t)\big\}\big) \geq U_{tot}(\boldsymbol{r}^*) - \tilde{B}/V.$$

This completes the proof of Part (b). $\square$

8. REFERENCES

[1] S. Meninger, J. O. Mur-Miranda, R. Amirtharajah, A. Chandrakasan, and J. H. Lang. Vibration-to-eletric energy conversion. *IEEE Trans. on VLSI, Vol. 9, No.1*, Feb. 2001.

[2] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. B. Srivastava. Design considerations for solar energy harvesting wireless embedded systems. *Proc. of IEEE IPSN*, April 2005.

[3] S. Chalasani and J. M. Conrad. A survey of energy harvesting sources for embedded systems. *IEEE Southeastcon*, 2008.

[4] M. Gorlatova, P. Kinget, I. Kymissis, D. Rubenstein, X. Wang, and G. Zussman. Challenge: Ultra-low-power energy-harvesting active networked tags (EnHANTs). *Proceedings of MobiCom*, Sept. 2009.

[5] Power from thin air. *Economist*, June 10, 2010.

[6] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. *7th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2006.

[7] T. R. Halford and K. M. Chugg. Barrage relay networks. *Information Theory and Applications Workshop (ITA)*, 2010.

[8] D. Graham-Rowe. Wireless power harvesting for cell phones. *MIT Technology Review*, June, 2009.

[9] V. Sharma, U. Mukherji, V. Joseph, and S. Gupta. Optimal energy management policieis for energy harvesting sensor nodes. *IEEE Trans. on Wireless Communication, Vol.9, Issue 4.*, April 2010.

[10] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava. Power management in energy harvesting sensor networks. *ACM Trans. on Embedded Computing Systems, Vol.6, Issue 4*, Sept. 2007.

[11] R. Srivastava and C. E. Koksal. Basic tradeoffs for energy management in rechargeable sensor networks. *ArXiv Techreport arXiv: 1009.0569v1*, Sept. 2010.

[12] M. Gatzianas, L. Georgiadis, and L. Tassiulas. Control of wireless networks with rechargeable batteries. *IEEE Trans. on Wireless Communications, Vol. 9, No. 2*, Feb. 2010.

[13] L. Lin, N. B. Shroff, and R. Srikant. Asymptotically optimal power-aware routing for multihop wireless networks with renewable energy sources. *Proceedings of INFOCOM*, 2005.

[14] L. Lin, N. B. Shroff, and R. Srikant. Energy-aware routing in sensor networks: A large system appraoch. *Ad Hoc Networks, Vol. 5, Issue 6, 818-831*, 2007.

[15] M. J. Neely. Energy optimal control for time-varying wireless networks. *IEEE Transactions on Information Theory 52(7): 2915-2934*, July 2006.

[16] M. J. Neely. Optimal energy and delay tradeoffs for multi-user wireless downlinks. *IEEE Transactions on Information Theory vol. 53, no. 9, pp. 3095-3113*, Sept. 2007.

[17] L. Georgiadis, M. J. Neely, and L. Tassiulas. *Resource Allocation and Cross-Layer Control in Wireless Networks*. Foundations and Trends in Networking Vol. 1, no. 1, pp. 1-144, 2006.

[18] M. J. Neely and L. Huang. Dynamic product assembly and inventory control for maximum profit. *IEEE Conference on Decision and Control (CDC), Atlanta, Georgia*, Dec. 2010.

[19] L. Huang and M. J. Neely. Utility optimal scheduling in processing networks. *ArXiv Techreport arXiv: 1010.1862v1*, 2010.

[20] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *IEEE/ACM Trans. on Networking, vol. 16, no. 2, pp. 396-409*, April 2008.

[21] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine, Vol. 44, Issue 11*, Nov. 2006.

[22] L. Huang and M. J. Neely. Delay reduction via Lagrange multipliers in stochastic network optimization. *IEEE Transactions on Automatic Control*, to appear.

[23] D. I. Shuman and M. Liu. Energy-efficient transmission scheduling for wireless media streaming with strict underflow constraints. *WiOpt*, 2008.

[24] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vols. I and II.* Boston: Athena Scientific, 2005 and 2007.

[25] L. Huang and M. J. Neely. Utility optimal scheduling in energy harvesting networks. *ArXiv technical report, arXiv:1012.1945v1*, Dec 2010.

[26] M. J. Neely. Universal scheduling for networks with arbitrary traffic, channels, and mobility. *Proc. IEEE Conf. on Decision and Control (CDC)*, Atlanta, GA, Dec 2010.

[27] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer congestion control in wireless networks. *IEEE/ACM Transactions on Networking, vol. 14, no. 2, pp302-315*, April 2006.