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 $O([\log(1/\epsilon)]^2)$.

Index Terms—Energy Harvesting, Lyapunov Analysis, Stochastic Network, Queueing

I. 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 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 has 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 source. 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 explicitly 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 our 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 II we state our network model and the objective. In Section III we first derive an upper bound on the maximum utility. Section IV presents the ESA algorithm. The $[O(\epsilon), O(1/\epsilon)]$ performance results of the ESA algorithm are presented in Section V, for both the cases when the network randomness is i.i.d. and Markovian. We then construct the Modified-ESA algorithm (MESA) to achieve the same $O(\epsilon)$ close-to-optimal utility performance with only $O([\log(1/\epsilon)]^2)$ energy storage sizes in Section VI. Simulation results are presented in Section VII. We conclude our paper in Section VIII.

II. 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.

A. 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 $U_n^{(c)}(\overline{r}^{nc})$, where $\overline{r}^{nc}$ is the time average rate of the commodity c traffic admitted into node n, defined as $\overline{r}^{nc} = \liminf_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{R_n^{(c)}(\tau)\}$. 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 $\beta = \max_{n,c} \beta^{nc}$.

B. 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 $\mathcal{S} = (s_1, ..., s_M)$. In the following, we first assume that $\boldsymbol{S}(t)$ is i.i.d. every time slot and use π_{s_i} to denote $Pr(\boldsymbol{S}(t) = s_i)$. We will later extend our results to the case when $\boldsymbol{S}(t)$ is Markovian. 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 vectors $\boldsymbol{P}, \boldsymbol{P}' \in \mathcal{P}^{(s_i)}$, where $\boldsymbol{P}'$ is obtained by changing any single component $P_{[n,m]}$ in $\boldsymbol{P}$ to zero, we have:

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

for some finite constant $\delta > 0$.

---

[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].

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

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

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 links do not interfere with each other.

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:

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

C. 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:

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

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 $\mathcal{H} = \{\boldsymbol{h}_1, ..., \boldsymbol{h}_K\}$, and that $\boldsymbol{h}(t)$ is i.i.d. over each slot. However, components in each $\boldsymbol{h}_i$ vector may be correlated. We will later consider the case when $\boldsymbol{h}(t)$ is Markovian. In both cases, we assume that $\boldsymbol{h}(t)$ is independent of $\boldsymbol{S}(t)$. [3]

We let $\pi_{\boldsymbol{h}_i} = Pr(\boldsymbol{h}(t) = \boldsymbol{h}_i)$. We assume that there exists $h_{max} < \infty$ such that $h_n(t) \leq h_{max}$ for all n, t, and 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 energy queue is below a finite threshold of size $O(1/\epsilon)$ and drops it otherwise, thus can be implemented with finite capacity storage devices.

D. 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:

$$
\begin{aligned}
Q_n^{(c)}(t+1) \leq \; & \left[Q_n^{(c)}(t) - \sum_{b \in \mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t) \right]^+ \quad (5) \\
& + \sum_{a \in \mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(t) + R_n^{(c)}(t),
\end{aligned}
$$

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 condition is met:

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

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 *deterministically* upper bounded, thus we only need a finite energy capacity in algorithm implementation.

E. 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

---

[2]Note that in our transmission model, we did not explicitly take into account the reception power. However, this can easily be incorporated into our model at the expense of more complicated notations. All the results in this paper will still hold in this case.

[3]This is for the ease of presentation. The results in this paper still hold if they are correlated.

[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.

transmits packets accordingly, so as to maximize the utility function:

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

subject to the network stability constraint (6). Here $\overline{r} = (\overline{r}^{nc}, \forall n, c \in \mathcal{N})$ 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).

F. Discussion of the Model

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.

The main difficulty in designing an optimal scheduling policy 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 of this kind usually have to be modeled as dynamic programs [22]. However, this approach typically requires significant statistical knowledge of the network randomness, including the channel state and the energy state. Another way to utilize the harvested energy efficiently is by developing efficient sleep-wake policies, e.g., [23]. Although our model does not consider this aspect, our algorithm can also be used together with given sleep-wake policies to achieve good utility performance in that context.

III. 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 $\mathbf{r}^*$ 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.

Theorem 1: The optimal network utility $U_{tot}(\mathbf{r}^*)$ satisfies the following:

$$V U_{tot}(\mathbf{r}^*) \le \phi^*, \qquad (9)$$

where ϕ^* is the optimal value of the following optimization problem:

$$\max: \phi = V \sum_{n,c} \sum_{k=1}^{K} \vartheta_k U_n^{(c)}(r_k^{nc}) \qquad (10)$$

$$s.t. \quad \sum_{k=1}^{K} \vartheta_k r_k^{nc} + \sum_{s_i} \pi_{s_i} \sum_{k=1}^{K} \varrho_k^{(s_i)} \sum_{a \in \mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(s_i, \mathbf{P}_k^{(s_i)})$$

$$\le \sum_{s_i} \pi_{s_i} \sum_{k=1}^{K} \varrho_k^{(s_i)} \sum_{b \in \mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(s_i, \mathbf{P}_k^{(s_i)}), \forall (n,c), \qquad (11)$$

$$\sum_{s_i} \pi_{s_i} \sum_{k=1}^{K} \varrho_k^{(s_i)} \sum_{b \in \mathcal{N}_n^{(o)}} P_{k,[n,b]}^{(s_i)} \qquad (12)$$

$$= \sum_{\mathbf{h}_i} \pi_{\mathbf{h}_i} \sum_{k=1}^{K} \varphi_k^{(\mathbf{h}_i)} e_{n,k}^{(\mathbf{h}_i)}, \forall n,$$

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

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

$$0 \le r_k^{nc} \le R_{max}, \forall (n,c),$$

$$0 \le e_{n,k}^{(\mathbf{h}_i)} \le h_n^{(\mathbf{h}_i)}, \ \forall \ n, k, \mathbf{h}_i.$$

Here $K = N^2 + N + 2$. [6] $\{r_k^{nc}\}_{k=1}^K$ denotes the set of admission decisions used for each commodity flow. $\{\mathbf{P}_k^{(s_i)}\}_{k=1}^K$ denotes the set of power allocation vectors that are used when $\mathbf{S}(t) = s_i$. $\mu_{[b,n]}^{(c)}(s_i, \mathbf{P}_k^{(s_i)})$ is the rate allocated to commodity c over link $[b,n]$ under s_i and $\mathbf{P}_k^{(s_i)}$. $P_{k,[n,m]}^{(s_i)}$ is the power allocated to link $[n,m]$ under $\mathbf{P}_k^{(s_i)}$. $\{e_{n,k}^{(\mathbf{h}_i)}\}_{k=1}^K$ is the set of energy harvesting decisions of node n when the energy state is $\mathbf{h}_i$, and $h_n^{(\mathbf{h}_i)}$ is the amount of harvestable energy for node n when $\mathbf{h}(t) = \mathbf{h}_i$.

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

Note that Theorem 1 indeed holds under more general ergodic $S(t)$ and $\mathbf{h}(t)$ processes, e.g., when $S(t)$ and $\mathbf{h}(t)$ evolve according to some finite state irreducible and aperiodic Markov chains. Also note that the objective function is not of the same form as $U_{tot}(\cdot)$. However, it can be shown, using Jensen's inequality, that the optimal value of the above optimization problem remains the same if we push $\sum_k \vartheta_k$ inside the function $U_n^{(c)}$, i.e., change the objective to $V \sum_{n,c} U_n^{(c)}(\sum_{k=1}^K \vartheta_k r_k^{nc})$. Below, we first have the following lemma regarding the dual problem of (10):

Lemma 1: The dual problem of (10) is given by:

$$\min: \ g(\mathbf{v}, \boldsymbol{\nu}), \quad s.t. \ \mathbf{v} \succeq \mathbf{0}, \boldsymbol{\nu} \in \mathbb{R}^N, \qquad (13)$$

where $\mathbf{v} = (v_n^{(c)}, \forall (n,c))$, $\boldsymbol{\nu} = (\nu_n, \forall n)$ and $g(\mathbf{v}, \boldsymbol{\nu})$ is the dual function defined:

$$g(\mathbf{v}, \boldsymbol{\nu}) = \sup_{r^{nc}, \mathbf{P}^{(s_i)}, e_n^{(\mathbf{h}_j)}} \sum_{s_i} \pi_{s_i} \sum_{\mathbf{h}_j} \pi_{\mathbf{h}_j} \Big\{ V \sum_{n,c} U_n^{(c)}(r^{nc})$$

$$- \sum_n v_n^{(c)} \Big[r^{nc} + \sum_{a \in \mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(s_i, \mathbf{P}^{(s_i)}) \qquad (14)$$

$$- \sum_{b \in \mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(s_i, \mathbf{P}^{(s_i)}) \Big]$$

$$- \sum_n \nu_n \Big[\sum_{b \in \mathcal{N}_n^{(o)}} P_{[n,b]}^{(s_i)} - e_n^{(\mathbf{h}_j)} \Big] \Big\}.$$

Moreover, let $(\mathbf{v}^*, \boldsymbol{\nu}^*)$ be an optimal solution of (13), then $\phi^* \le g(\mathbf{v}^*, \boldsymbol{\nu}^*)$.

Proof: The proof uses a similar argument as the one used in [19]. Hence is omitted for brevity. ∎

---

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

Note that the dual function $g(\boldsymbol{v}, \boldsymbol{\nu})$ does not contain the terms $\vartheta_k, \varrho_k^{(s_i)}, \varphi_k^{\boldsymbol{h}_i}$. This not only simplifies the evaluation of the dual function, but also enables us to analyze the performance of our algorithm using Theorem 1. In the following, it is also useful to define the function $g_{s_i, \boldsymbol{h}_j}(\boldsymbol{v}, \boldsymbol{\nu})$ for each $(s_i, \boldsymbol{h}_j)$ pair:

$$
\begin{aligned}
g_{s_i, \boldsymbol{h}_j}(\boldsymbol{v}, \boldsymbol{\nu}) = & \sup_{r^{nc}, \boldsymbol{P}^{(s_i)}, e_n^{(\boldsymbol{h}_j)}} \bigg\{ V \sum_{n,c} U_n^{(c)}(r^{nc}) \\
& - \sum_n v_n^{(c)} \big[r^{nc} + \sum_{a \in \mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(s_i, \boldsymbol{P}^{(s_i)}) \\
& - \sum_{b \in \mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(s_i, \boldsymbol{P}^{(s_i)}) \big] \\
& - \sum_n \nu_n \big[\sum_{b \in \mathcal{N}_n^{(o)}} P_{[n,b]}^{(s_i)} - e_n^{(\boldsymbol{h}_j)} \big] \bigg\}.
\end{aligned} \tag{15}
$$

That is, $g_{s_i, \boldsymbol{h}_j}(\boldsymbol{v}, \boldsymbol{\nu})$ is the dual function of (10) when there is a single channel state s_i and a single energy state $\boldsymbol{h}_j$. It is easy to see from (14) and (15) that:

$$
g(\boldsymbol{v}, \boldsymbol{\nu}) = \sum_{s_i} \pi_{s_i} \sum_{\boldsymbol{h}_j} \pi_{\boldsymbol{h}_j} g_{s_i, \boldsymbol{h}_j}(\boldsymbol{v}, \boldsymbol{\nu}). \tag{16}
$$

IV. 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.

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:

$$
L(t) \triangleq \frac{1}{2} \sum_{n,c \in \mathcal{N}} \big[Q_n^{(c)}(t) \big]^2 + \frac{1}{2} \sum_{n \in \mathcal{N}} \big[E_n(t) - \theta_n \big]^2. \tag{17}
$$

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

$$
\Delta(t) = \mathbb{E}\big\{ L(t+1) - L(t) \mid \boldsymbol{Z}(t) \big\}. \tag{18}
$$

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. We have the following lemma regarding the drift:

Lemma 2: 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:

$$
\Delta(t) - V \mathbb{E}\big\{ \sum_{n,c} U_n^{(c)}(R_n^{(c)}(t)) \mid \boldsymbol{Z}(t) \big\} \tag{19}
$$

$$
\begin{aligned}
\leq & B - V \mathbb{E}\big\{ \sum_{n,c} U_n^{(c)}(R_n^{(c)}(t)) \mid \boldsymbol{Z}(t) \big\} \\
& - \sum_n \sum_c Q_n^{(c)}(t) \mathbb{E}\big\{ \sum_{b \in \mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t) \\
& \qquad - \sum_{a \in \mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(t) - R_n^{(c)}(t) \mid \boldsymbol{Z}(t) \big\} \\
& - \sum_{n \in \mathcal{N}} (E_n(t) - \theta_n) \mathbb{E}\big\{ \sum_{b \in \mathcal{N}_n^{(o)}} P_{[n,b]}(t) - e_n(t) \mid \boldsymbol{Z}(t) \big\}.
\end{aligned}
$$

Here $B = N^2 (\mu_{max}^2 + \frac{1}{2} R_{max}^2) + \frac{N}{2} [P_{max}^2 + h_{max}^2]$.

Proof: See Appendix A. ∎

Now denote the left-hand side (LHS) of (19) as $\Delta_V(t)$, we can rearrange the terms in (19) to get:

$$
\begin{aligned}
\Delta_V(t) \leq & B + \sum_{n \in \mathcal{N}} (E_n(t) - \theta_n) \mathbb{E}\big\{ e_n(t) \mid \boldsymbol{Z}(t) \big\} \tag{20} \\
& - \mathbb{E}\big\{ \sum_{n,c} \big[V U_n^{(c)}(R_n^{(c)}(t)) - Q_n^{(c)}(t) R_n^{(c)}(t) \big] \mid \boldsymbol{Z}(t) \big\} \\
& - \mathbb{E}\big\{ \sum_n \big[\sum_c \sum_{b \in \mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t) \big[Q_n^{(c)}(t) - Q_b^{(c)}(t) \big] \\
& \qquad + (E_n(t) - \theta_n) \sum_{b \in \mathcal{N}_n^{(o)}} P_{[n,b]}(t) \big] \mid \boldsymbol{Z}(t) \big\}.
\end{aligned}
$$

We now present the ESA algorithm. The idea of the algorithm is to approximately minimize the right-hand side (RHS) of (20) subject to the energy-availability constraint (4). In the algorithm, 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 $\boldsymbol{Q}(t)$, $\boldsymbol{E}(t)$, and $\boldsymbol{S}(t)$, 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 $e_n(t) = 0$.
- Data Admission: At every time t, choose $R_n^{(c)}(t)$ to be the optimal solution of the following optimization problem:

$$
\max: \; V U_n^{(c)}(r) - Q_n^{(c)}(t) r, \;\; s.t. \; 0 \leq r \leq R_{max}. \tag{21}
$$

- Power Allocation: At every time t, define the weight of the commodity c data over link $[n, b]$ as:

$$
W_{[n,b]}^{(c)}(t) \triangleq \big[Q_n^{(c)}(t) - Q_b^{(c)}(t) - \gamma \big]^+. \tag{22}
$$

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:

$$
G(\boldsymbol{P}(t)) \triangleq \sum_n \bigg[\sum_{b \in \mathcal{N}_n^{(o)}} \mu_{[n,b]}(t) W_{[n,b]}(t) \tag{23}
$$

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

subject to the energy availability constraint (4).

- Routing and Scheduling: For every node n, find any $c^* \in \operatorname{argmax}_c W_{[n,b]}^{(c)}(t)$. If $W_{[n,b]}^{(c^*)}(t) > 0$, set:

$$\mu_{[n,b]}^{(c^*)}(t) = \mu_{[n,b]}(t), \qquad (24)$$

  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.
- Queue Update: Update $Q_n^{(c)}(t)$ and $E_n(t)$ according to the dynamics (5) and (7), respectively.

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 obtained. 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. Also note that if all the links do not interfere with each other, then 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.

V. PERFORMANCE ANALYSIS

We now present the performance results of the ESA algorithm. In the following, we first present the results under i.i.d. network randomness and give its proof in the appendix. We later extend the performance results of ESA to the case when the network randomness is Markovian.

A. ESA under I.I.D. randomness

Here we state the performance of ESA under the case when the channel state and the energy state, i.e., $\boldsymbol{S}(t)$ and $\boldsymbol{h}(t)$, are both i.i.d.

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

(a) The data queues and the energy queues satisfy the following for all time:

$$0 \leq Q_n^{(c)}(t) \leq \beta V + R_{max}, \quad \forall \, (n,c), \qquad (25)$$
$$0 \leq E_n(t) \leq \theta_n + h_{max}, \quad \forall \, n. \qquad (26)$$

  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:

$$U_{tot}(\overline{\boldsymbol{r}}) = \sum_{n,c} U_n^{(c)}(\overline{r}^{nc}) \geq U_{tot}(\boldsymbol{r}^*) - \frac{\tilde{B}}{V}, \qquad (27)$$

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

Proof: See Appendix B. ∎

We note the following of Theorem 2: First, we will see that Part (a) is indeed a *sample path* result. Hence it holds under *arbitrary* $\boldsymbol{S}(t)$ and $\boldsymbol{h}(t)$ processes. Thus they also hold when $\boldsymbol{S}(t)$ and $\boldsymbol{h}(t)$ evolve according to some finite state

irreducible and aperiodic Markov chain. Second, by taking $\epsilon = 1/V$, we see from Part (a) 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. Third, we see from Part (a) that the energy queue size is deterministically upper bounded by some $O(1/\epsilon)$ constant. This provides an explicit characterization of the size of the energy storage device that is needed for achieving the desired utility performance. Such explicit bounds are particularly useful for system deployments.

B. ESA under Markovian randomness

We now extend our results to the more general setting where the channel state $\boldsymbol{S}(t)$ and the energy state $\boldsymbol{h}(t)$ both evolve according to some finite state irreducible and aperiodic Markov chains. Note that in this case π_{s_i} and π_{h_i} represent the steady state probability of the events $\{\boldsymbol{S}(t) = s_i\}$ and $\{\boldsymbol{h}(t) = h_i\}$, respectively. In this case, the performance results of ESA are summarized in the following theorem:

Theorem 3: Suppose that $\boldsymbol{S}(t)$ and $\boldsymbol{h}(t)$ evolve according to some finite state irreducible and aperiodic Markov chains. Then under ESA, we have: (a) the bounds (25) and (26) still hold. (b) the average utility is within $O(1/V)$ of $U_{tot}(\boldsymbol{r}^*)$, i.e., $U_{tot}(\overline{\boldsymbol{r}}) = \sum_{n,c} U_n^{(c)}(\overline{r}^{nc}) \geq U_{tot}(\boldsymbol{r}^*) - O(1/V)$.

Proof: Part (a) follows from Theorem 2, since it is indeed a sample-path result. The proof of the utility performance is similar to that in [24], and hence is omitted for brevity. ∎

VI. 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 $O([\log(1/\epsilon)]^2)$ sizes, while guaranteeing a much smaller average data queue size, i.e., $O([\log(1/\epsilon)]^2)$. Our algorithm is motivated by the following theorem, which is a modified version of Theorem 2 in [25]. In the theorem, we denote $\boldsymbol{y} = (\boldsymbol{v}, \boldsymbol{\nu})$.

Theorem 4: Suppose that $\boldsymbol{h}(t)$ and $S(t)$ both evolve according some finite state irreducible and aperiodic Markov chain, that $\boldsymbol{y}^* = (\boldsymbol{v}^*, \boldsymbol{\nu}^*)$ is finite and unique, that $\boldsymbol{\theta}$ is chosen such that $\theta_n + \nu_n^* > 0$, $\forall \, n$, and that for all $\boldsymbol{y} = (\boldsymbol{v}, \boldsymbol{\nu})$ with $\boldsymbol{v} \succeq \boldsymbol{0}, \boldsymbol{\nu} \in \mathbb{R}^N$, the dual function $g(\boldsymbol{y})$ satisfies:

$$g(\boldsymbol{y}^*) \geq g(\boldsymbol{y}) + L||\boldsymbol{y}^* - \boldsymbol{y}||, \qquad (28)$$

for some constant $L > 0$ independent of V. Then $\boldsymbol{y}^* = \Theta(V)$, and that under ESA, there exists constants $D, K, c^* = \Theta(1)$, i.e., all independent of V, such that for any $m \in \mathbb{R}_+$,

$$\mathcal{P}^{(r)}(D, Km) \leq c^* e^{-m}, \qquad (29)$$

where $\mathcal{P}^{(r)}(D, Km)$ is defined:

$$\mathcal{P}^{(r)}(D, Km) \triangleq \limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} Pr\{\mathscr{E}(\tau, m)\}, \qquad (30)$$

with $\mathscr{E}(t, m)$ being the following deviation event:

$$\mathscr{E}(t, m) = \{\exists\,(n,c), |Q_n^{(c)}(t) - v_n^{(c)*}| > D + Km\} \qquad (31)$$
$$\cup \{\exists\, n, \; |(E_n(t) - \theta_n) - \nu_n^*| > D + Km\}.$$

Proof: The proof is similar in spirit to [25] and is omitted for brevity. ∎

Note that the finiteness and uniqueness of $\boldsymbol{y}^*$ can usually be satisfied in practice, particularly when a certain "slackness" condition is met. Also note that the condition (28) can typically be satisfied in practice when the action space is finite (See [25] for further discussions of these conditions). In this case, Theorem 4 states that the queue backlog vector pair is "exponentially attracted" to the fixed point $(\boldsymbol{v}^*, \boldsymbol{\nu}^* + \boldsymbol{\theta}) = \Theta(V)$, in that the probability of deviating decreases exponentially with the deviation distance. Therefore, the probability of deviating by some $\Theta(\log(V))$ distance will be $1/V$, which will be very small when V is large. This suggests that most of the queue backlogs are kept in the queues to maintain a "proper" queue vector value to base the decisions on. If we can somehow learn the value of this vector, then we can "subtract out" a large amount of backlog from the network and reduce the required buffer sizes. Below, we present the Modified-ESA (MESA) algorithm.

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 energy harvesting, power allocation, and routing and scheduling actions so as to ensure energy availability and reduce network delay.

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

- <u>Phase I</u>: Choose a sufficiently large T. From time $t = 0, ..., T$, run ESA using $\hat{Q}(t)$ and $\hat{E}(t)$ as the data and energy queue processes. 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:

$$\mathcal{Q}_n^{(c)} = [\hat{Q}_n^{(c)}(T) - \frac{M}{2}]^+, \ \mathcal{E} = [\hat{E}_n(T) - \frac{M}{2}]^+. \quad (32)$$

- <u>Phase II</u>: Reset $t = 0$. Initialize $\hat{E}(0) = \boldsymbol{\mathcal{E}}$ and $\hat{Q}(0) = \boldsymbol{\mathcal{Q}}$. Also set $Q(0) = \boldsymbol{0}$ and $E(0) = \boldsymbol{0}$. In every time slot, first run the ESA algorithm based on $\hat{Q}(t)$, $\hat{E}(t)$, and $S(t)$, 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:
  - <u>Energy harvesting</u>: If $\hat{E}_n(t) < \mathcal{E}_n$, let $\tilde{e}_n(t) = [e_n(t) - (\mathcal{E}_n - \hat{E}_n(t))]^+$ and 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:

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

- <u>Packet Dropping</u>: 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 $\hat{E}_a(t) \in [\mathcal{E}_n + P_{max}, \mathcal{E} + M]$. 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 $Q_n^{(c)}(t)$ according to:

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

  * If $\hat{Q}_n^{(c)}(t) \geq \mathcal{Q}_n^{(c)}$, update $Q_n^{(c)}(t)$ according to:

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

- 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 $\boldsymbol{E}(t)$. Note that 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 $Q_n^{(c)}(t) = \hat{Q}_n^{(c)}(t) - \mathcal{Q}_n^{(c)}$. Our algorithm is designed to ensure that $\hat{Q}_n^{(c)}(t)$ and $\hat{E}_n(t)$ mostly stay in the "right" arrange, as shown in the following lemma.

Lemma 3: For all time t, we have the following:

$$0 \leq Q_n^{(c)}(t) \leq [\hat{Q}_n^{(c)}(t) - \mathcal{Q}_n^{(c)}]^+ + \gamma, \forall (n, c), \quad (33)$$

$$\min\big[[\hat{E}_n(t) - \mathcal{E}_n]^+, M\big] \leq E_n(t). \quad (34)$$

Proof: See Appendix C. ∎

We now summarize the performance result of MESA in the following theorem:

Theorem 5: Suppose that the conditions in Theorem 4 hold, 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:

$$\overline{\boldsymbol{Q}} \leq O([\log(V)]^2), \quad (35)$$

$$U_{tot}(\overline{\boldsymbol{r}}) \geq U_{tot}(\boldsymbol{r}^*) - O(1/V). \quad (36)$$

Proof: See Appendix D. ∎

Note that the conditions in Theorem 4 are indeed the conditions needed for proving the exponential attraction result in [25]. Thus Theorem 5 implies that if the exponential attraction result holds, which is mostly the case in practice (See [25]

for more discussions), then one can significantly reduce the energy capacity needed to achieve the $O(\epsilon)$ close-to-optimal utility performance and greatly reduce the network congestion.

VII. SIMULATION

In this section we provide simulation results of the ESA algorithm. We consider a data collection network shown in Fig. 1. Such networks typically appear 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 nodes $1, 2, 3$ sense data and deliver them to node $Sink$ via the relay of nodes $4, 5$.
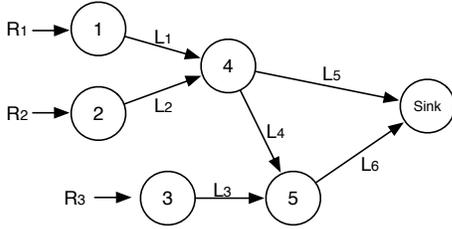
Fig. 1. A data collection network, where L_i denotes link i.

The channel state of each communication link L_i, represented by a directed edge, can be either "G=Good" or "B=Bad", and evolves according to the two-state Markov chain shown in Fig. 2 with $\rho_G = \rho_B = 0.3$. At any time, we can allocate either zero or one unit of power. 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: $U_1(r) = U_2(r) = U_3(r) = \log(1+r)$ and $U_4(r) = U_5(r) = 0$. For simplicity, we also assume that all the links do not interfere with each other.
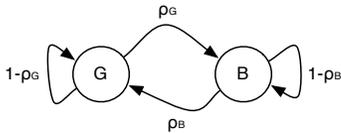

Fig. 2. A two-state Markov chain.

We also assume that for each node, the available energy $h_n(t)$ evolves according to the same two-state Markov chain in Fig. 2. When the state is good, $h_n(t) = 2$, otherwise $h_n(t) = 0$. It is easy to see that in this case, $\beta = 1$, $\delta = 2$, $\mu_{max} = 2$, $d_{max} = 2$ and $P_{max} = 2$. Using the results in Theorem 2, we set $\theta_n = \delta\beta V + P_{max} = 2V + 2$. We also see that in this case, we can use $\gamma = d_{max}\mu_{max} + R_{max} = 7$. The simulation results are plotted in Fig. 3. We see in Fig. 3 that the total network utility converges quickly to very close to the optimal value, which can be shown to be roughly 2.03. We also see that the average data queue size and the average energy queue size both grow linearly in V.

Fig. 4 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. Interestingly, we see that all the queue sizes
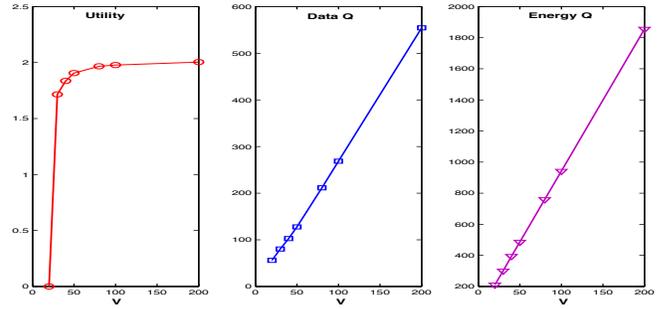

Fig. 3. Simulation results of ESA.

are "attracted" to certain fixed points. However, different from previous work, e.g., [25], we see that the queue size of $Q_1(t)$ does not approach this fixed point from below. It instead first has a "burst" in the early time slots. This is due to the fact that the systems "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.
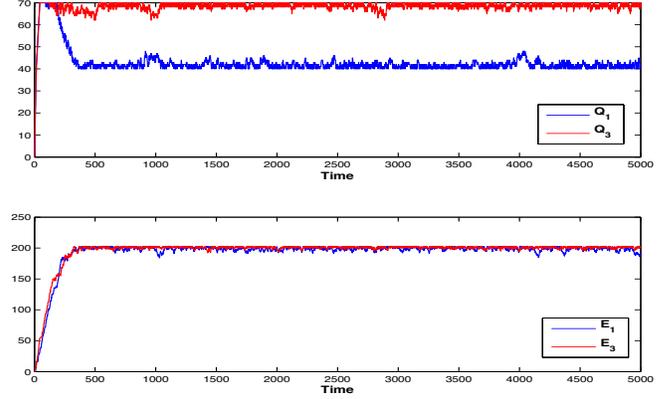

Fig. 4. Sample path queue processes.

We also simulate the MESA algorithm for the same network with the same θ value. We use $T = 50V$ in Phase I for obtaining the vectors $\mathcal{E}$ and $\mathcal{Q}$. Fig. 5 plots the performance results. We observe that extremely few packets were dropped in the simulations (at most 5 out of more than 10^5 packets were dropped 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 poly-logarithmically 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 simulation results and Theorem 5.

VIII. CONCLUSION

In this paper, we develop the Energy-limited Scheduling Algorithm (ESA) for achieving optimal utility in general energy harvesting networks equipped with only finite capacity energy storage device. We show that ESA is able to achieve an average utility that is within $O(\epsilon)$ of the optimal for any $\epsilon > 0$ using energy storage devices of $O(1/\epsilon)$ sizes, while guaranteeing that the time average network congestion is $O(1/\epsilon)$. We then
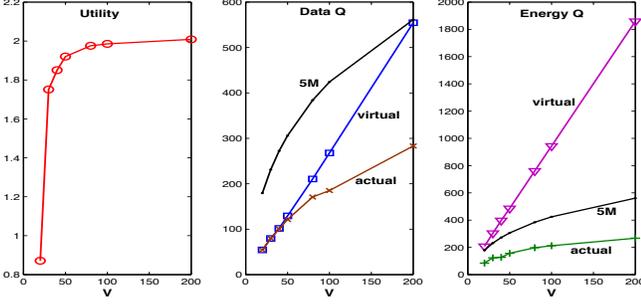
Fig. 5. Simulation results of MESA.

also develop the Modified-ESA algorithm (MESA), and show that MESA can achieve the same $O(\epsilon)$ utility performance using energy storage devices of only $O([\log(1/\epsilon)]^2)$ sizes.

APPENDIX A – PROOF OF LEMMA 2

Here we prove Lemma 2

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:

$$
\begin{aligned}
&[Q_n^{(c)}(t+1)]^2 - [Q_n^{(c)}(t)]^2 \qquad\qquad (37)\\
&\leq \Big[\sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t)\Big]^2 + \Big[\sum_{a\in\mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(t) + R_n^{(c)}(t)\Big]^2\\
&\quad - 2Q_n^{(c)}(t)\Big[\sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t) - \sum_{a\in\mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(t) - R_n^{(c)}(t)\Big].
\end{aligned}
$$

By defining $\hat{B} = \frac{3}{2}d_{max}^2\mu_{max}^2 + R_{max}^2$, we see that:

$$
\begin{aligned}
&\frac{1}{2}\big([Q_n^{(c)}(t+1)]^2 - [Q_n^{(c)}(t)]^2\big) \leq \hat{B} \qquad\qquad (38)\\
&\quad - Q_n^{(c)}(t)\Big[\sum_{b\in\mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t) - \sum_{a\in\mathcal{N}_n^{(in)}} \mu_{[a,n]}^{(c)}(t) - R_n^{(c)}(t)\Big].
\end{aligned}
$$

Using a similar approach, we get that:

$$
\begin{aligned}
&\frac{1}{2}\big([E_n(t+1) - \theta_n]^2 - [E_n(t) - \theta_n]^2\big) \qquad\qquad (39)\\
&\quad \leq \hat{B}' - [E_n(t) - \theta_n]\Big[\sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(t) - e_n(t)\Big],
\end{aligned}
$$

where $\hat{B}' = \frac{1}{2}(P_{max} + h_{max})^2$. Now by summing (38) over all (n, c) and (39) 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:

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

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. ∎

APPENDIX B – PROOF OF THEOREM 2

Here we prove Theorem 2. The proof idea is as follows: We first show that by our choice of $\boldsymbol{\theta}$, the ESA algorithm ensures the energy-availability constraint (4) even if we remove it from the algorithm. This enables us to show that ESA approximately minimizes the value of the RHS of (20) over all possible policies. We then analyze the utility performance of ESA by relating the value of the RHS of (20) under ESA to the dual function $g_{s_i, \boldsymbol{h}_j}(\boldsymbol{v}, \boldsymbol{\nu})$.

Proof: (Part (a)) We first prove (25) 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 time t, we want to show that it holds for time $t + 1$. First, if node n does not receive any commodity c data from other nodes, 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 according to the ESA algorithm, 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 (21), 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 (26).

We now show that if $E_n(t) \leq P_{max}$, then $G(\boldsymbol{P}(t))$ will be maximized by choosing $P_{[n,b]}(t) = 0$ for all $b \in \mathcal{N}_n^{(o)}$ at node n. To see this, first note that since all the actual 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 allocation vector that maximizes $G(\boldsymbol{P}(t))$ at time 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 $\boldsymbol{P}^*$. We see that $\boldsymbol{P}$ is also feasible. 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:

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

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 $W_{[n,b]}(t) \leq \beta V - d_{max}\mu_{max}$, the above implies:

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

This shows that $\boldsymbol{P}^*$ cannot have been the power vector that maximizes $G(\boldsymbol{P}(t))$ 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 (20). To see this, note from Part (A) that ESA indeed minimizes the following function at time t:

$$D(t) = \sum_{n \in \mathcal{N}} (E_n(t) - \theta_n)e_n(t) \tag{40}$$
$$- \sum_{n,c \in \mathcal{N}} \left[VU_n^{(c)}(R_n^{(c)}(t)) - Q_n^{(c)}(t)R_n^{(c)}(t) \right]$$
$$- \sum_{n \in \mathcal{N}} \left[\sum_c \sum_{b \in \mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t) \left[Q_n^{(c)}(t) - Q_b^{(c)}(t) - \gamma \right] \right.$$
$$\left. + (E_n(t) - \theta_n) \sum_{b \in \mathcal{N}_n^{(o)}} P_{[n,b]}(t) \right],$$

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:

$$\tilde{D}(t) = \sum_{n \in \mathcal{N}} (E_n(t) - \theta_n)e_n(t) \tag{41}$$
$$- \sum_{n,c \in \mathcal{N}} \left[VU_n^{(c)}(R_n^{(c)}(t)) - Q_n^{(c)}(t)R_n^{(c)}(t) \right]$$
$$- \sum_{n \in \mathcal{N}} \left[\sum_c \sum_{b \in \mathcal{N}_n^{(o)}} \mu_{[n,b]}^{(c)}(t) \left[Q_n^{(c)}(t) - Q_b^{(c)}(t) \right] \right.$$
$$\left. + (E_n(t) - \theta_n) \sum_{b \in \mathcal{N}_n^{(o)}} P_{[n,b]}(t) \right].$$

Note that $\tilde{D}(t)$ is indeed the function inside the expectation on the RHS of the drift bound (19). 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:

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

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:

$$\tilde{D}^E(t) \leq \tilde{D}^{ALT}(t) + N^2 d_{max}\mu_{max}. \tag{42}$$

That is, the value of $\tilde{D}(t)$ under ESA is no greater than its value under any other alternative policy plus a constant. Now

using the definition of $\tilde{D}(t)$, (19) can be rewritten as:

$$\Delta(t) - V\mathbb{E}\left\{ \sum_{n,c} U_n^{(c)}(R_n^{(c)}(t)) \mid \boldsymbol{Z}(t) \right\}$$
$$\leq B + \mathbb{E}\left\{ \tilde{D}^E(t) \mid \boldsymbol{Z}(t) \right\}.$$

Using (42), we get:

$$\Delta(t) - V\mathbb{E}\left\{ \sum_{n,c} U_n^{(c)}(R_n^{(c)}(t)) \mid \boldsymbol{Z}(t) \right\} \tag{43}$$
$$\leq \tilde{B} + \mathbb{E}\left\{ \tilde{D}^{ALT}(t) \mid \boldsymbol{Z}(t) \right\},$$

where $\tilde{B} = B + N^2\gamma d_{max}\mu_{max}$. Now consider the policy that minimizes $\tilde{D}(t)$ subject to only $e_n(t) \in [0, h_n(t)]$, $0 \leq R_n^{(c)}(t) \leq R_{max}$, $\boldsymbol{P} \in \mathcal{P}^{(\boldsymbol{S}(t))}$ and (3), and denote the value of $\tilde{D}(t)$ under this policy by $\tilde{D}^*(t)$. It is easy to see then $\tilde{D}^*(t)$ is obtained by minimizing each term in (41) over the constraints. Hence by comparing $\tilde{D}^*(t)$ with (15), we see that indeed, when $S(t) = s_i$ and $\boldsymbol{h}(t) = \boldsymbol{h}_j$,

$$\tilde{D}^*(t) = -g_{s_i, \boldsymbol{h}_j}(\boldsymbol{Q}(t), \boldsymbol{\theta} - \boldsymbol{E}(t)).$$

Using this fact in (43), we have under ESA that:

$$\Delta(t) - V\mathbb{E}\left\{ \sum_{n,c} U_n^{(c)}(R_n^{(c)}(t)) \mid \boldsymbol{Z}(t) \right\} \tag{44}$$
$$\leq \tilde{B} - \mathbb{E}\left\{ g_{s_i, \boldsymbol{h}_j}(\boldsymbol{Q}(t), \boldsymbol{\theta} - \boldsymbol{E}(t)) \mid \boldsymbol{Z}(t) \right\}.$$

Now using (16), i.e., $g(\boldsymbol{v}, \boldsymbol{\nu}) = \sum_{s_i} \pi_{s_i} \sum_{\boldsymbol{h}_j} \pi_{\boldsymbol{h}_j} g_{s_i, \boldsymbol{h}_j}(\boldsymbol{v}, \boldsymbol{\nu})$, the above becomes:

$$\Delta(t) - V\mathbb{E}\left\{ \sum_{n,c} U_n^{(c)}(R_n^{(c)}(t)) \mid \boldsymbol{Z}(t) \right\} \tag{45}$$
$$\leq \tilde{B} - g(\boldsymbol{Q}(t), \boldsymbol{\theta} - \boldsymbol{E}(t)).$$

By Theorem 1 and Lemma 1, we see that:

$$VU_{tot}(\boldsymbol{r}^*) \leq \phi^* \leq g(\boldsymbol{v}^*, \boldsymbol{\nu}^*) \leq g(\boldsymbol{Q}(t), \boldsymbol{\theta} - \boldsymbol{E}(t)).$$

Plug this into (45), we get:

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

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

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

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:

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

Using Jensen's inequality, we see that:

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

This completes the proof of Part (b). ∎

Here we prove Lemma 3.

Proof: We first prove (33). We first define an intermediate process $\tilde{Q}_n^{(c)}(t)$ that evolves exactly as $Q_n^{(c)}(t)$ except that it does not discard packets when $\hat{E}_n(t) < \mathcal{E}_n + P_{max}$ or $\hat{E}_n(t) > \mathcal{E}_n + M$. We see then $Q_n^{(c)}(t) \leq \tilde{Q}_n^{(c)}(t)$. Using Lemma 3 in [25], we see that: $\tilde{Q}_n^{(c)}(t) \leq [\hat{Q}_n^{(c)}(t) - \mathcal{Q}_n^{(c)}]^+ + \gamma$. Hence $Q_n^{(c)}(t) \leq [\hat{Q}_n^{(c)}(t) - \mathcal{Q}_n^{(c)}]^+ + \gamma$ and (33) follows.

We now look at (34). We see that it holds at time 0 since $0 = \hat{E}_n(0) - \mathcal{E}_n = E_n(0)$. Now suppose that it holds for $t = 0, 1, ..., k$. We will show that it holds for $t = k+1$. Since if $\hat{E}_n(k+1) \leq \mathcal{E}_n$, then (34) always holds. Below, we only consider the case when $\hat{E}_n(k+1) > \mathcal{E}_n$, i.e.,

$$[\hat{E}_n(k+1) - \mathcal{E}_n]^+ = \hat{E}_n(k+1) - \mathcal{E}_n. \tag{46}$$

Also note that since all the actions are made based on $\hat{Q}(t)$ and $\hat{E}(t)$, by Theorem 2, we always have $\hat{E}_n(t) \geq \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(t)$, thus:

$$\hat{E}_n(t+1) = \hat{E}_n(t) - \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(t) + e_n(t). \tag{47}$$

We consider the following three cases:

(I) $\hat{E}_n(k) < \mathcal{E}_n$. Since $\hat{E}_n(k+1) > \mathcal{E}_n$, we must have $\mathcal{E}_n - \hat{E}_n(k) \leq e_n(k)$. Then according to the harvesting rule,

$$
\begin{aligned}
&E_n(k+1) \\
&= \min\Big[[E_n(k) - \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(k)]^+ \\
&\qquad\qquad + e_n(k) - \mathcal{E}_n + \hat{E}_n(k), M\Big] \\
&\geq \min\Big[[\hat{E}_n(k) + E_n(k) - \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(k)]^+ \\
&\qquad\qquad + e_n(k) - \mathcal{E}_n, M\Big] \\
&\geq \min\Big[\hat{E}_n(k) - \sum_{[n,b]} P_{[n,b]}(k) + e_n(k) - \mathcal{E}_n, M\Big] \\
&= \min\Big[\hat{E}_n(k+1) - \mathcal{E}_n, M\Big] \\
&= \min\Big[[\hat{E}_n(k+1) - \mathcal{E}_n]^+, M\Big].
\end{aligned}
$$

Here the first inequality uses the property of $[\cdot]^+$, and the second inequality uses $E_n(k) \geq 0$ and $\hat{E}_n(k) \geq \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(k)$.

(II) $\hat{E}_n(k) > \mathcal{E}_n + M$. In this case, we see by the induction assumption that $E_n(k) = M$. Now by the update rule, we see that:

$$E_n(k+1) = \min\big[E_n(k) + e_n(k), M\big] = M. \tag{48}$$

Thus (34) still holds.

(III) $\mathcal{E}_n \leq \hat{E}_n(k) \leq \mathcal{E}_n + M$. We have two cases:

(III-A) If $\hat{E}_n(k+1) - \mathcal{E}_n \leq M$, then using (46) and (47),

we have:

$$
\begin{aligned}
&\min\big[[\hat{E}_n(k+1) - \mathcal{E}_n]^+, M\big] \\
&= \hat{E}_n(k) - \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(k) + e_n(k) - \mathcal{E}_n \\
&\leq \min\big[[[\hat{E}_n(k) - \mathcal{E}_n]^+ - \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(k)]^+ + e_n(k), M\big] \\
&\leq \min\big[[E_n(k) - \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(k)]^+ + e_n(k), M\big] \\
&= E_n(k+1).
\end{aligned}
$$

Here the first inequality uses the property of the operator $[\cdot]^+$, and the second inequality uses the induction that $E_n(k) \geq \min\big[[\hat{E}_n(k) - \mathcal{E}_n]^+, M\big] = [\hat{E}_n(k) - \mathcal{E}_n]^+$.

(III-B) If $\hat{E}_n(k+1) - \mathcal{E}_n > M$, then we must have $\hat{E}_n(k) \geq \mathcal{E}_n + M - \alpha_{max}$, and that $E_n(k) \geq \hat{E}_n(k) - \mathcal{E}_n \geq M - \alpha_{max}$. Using the fact that $\frac{M}{2} \geq \alpha_{max}$, we have:

$$
\begin{aligned}
&E_n(k+1) \\
&= \min\big[E_n(k) - \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(k) + e_n(k), M\big] \\
&\geq \min\big[\hat{E}_n(k) - \sum_{b\in\mathcal{N}_n^{(o)}} P_{[n,b]}(k) + e_n(k) - \mathcal{E}_n, M\big] \\
&= \min[\hat{E}_n(k+1) - \mathcal{E}_n, M],
\end{aligned}
$$

which implies $E_n(k+1) = M$. Thus (34) holds. This completes the proof of (34) and proves the lemma. ∎

Here we prove Theorem 5.

Proof: Since a steady state distribution for the queues exists under the ESA algorithm, we see that $\mathcal{P}^{(r)}(D, Km)$ is the steady state probability that event $\mathscr{E}(t, m)$ happens. Now consider a large V value that satisfies $\frac{M}{8} = \frac{1}{2}[\log(V)]^2 \geq 2D$ and $\log(V) \geq 16K$. We have:

$$\frac{\frac{1}{2}[\log(V)]^2 - D}{K} \geq \frac{\frac{1}{4}[\log(V)]^2}{K} \geq 4\log(V).$$

By using (29) and the above, we see that

$$Pr(\mathscr{E}(T, \frac{\frac{1}{2}[\log(V)]^2 - D}{K})) \leq c^* e^{-4\log(V)} = O(1/V^4).$$

Using the definition of $\mathscr{E}(t, m)$, we see that when V is large enough, with probability $1 - O(1/V^4)$, the vectors $\hat{E}(T)$ and $\hat{Q}(T)$ satisfy the following for all n, c:

$$|\hat{Q}_n^{(c)}(T) - v_n^{(c)*}| \leq \frac{M}{8}, |\hat{E}_n(T) - (\theta_n + \nu_n^*)| \leq \frac{M}{8}. \tag{49}$$

Using the fact that $\mathcal{Q}_n^{(c)} = [\hat{Q}_n^{(c)}(T) - \frac{M}{2}]^+$ and $\mathcal{E}_n = [\hat{E}_n(T) - \frac{M}{2}]^+$, (49) and the facts that $M = 4[\log(V)]^2$ and $y^* = (v^*, \nu^*) = \Theta(V)$ imply that, when V is large enough, with probability $1 - O(1/V^4)$, we have:

$$-\frac{3M}{8} \geq \mathcal{Q}_n^{(c)} - v_n^{(c)*} \geq -\frac{5M}{8}, \forall (n, c) \text{ s.t. } v_n^{(c)*} \neq 0, \tag{50}$$

$$\mathcal{Q}_n^{(c)} = v_n^{(c)*}, \forall (n, c) \text{ s.t. } v_n^{(c)*} = 0, \tag{51}$$

$$-\frac{3M}{8} \geq \mathcal{E}_n - (\theta_n + \nu_n^*) \geq -\frac{5M}{8}, \forall n. \tag{52}$$

Having established (50)-(52), (35) can now be proven using (33) in Lemma 3 and a same argument as in the proof of Theorem 4 in [25].

Now we consider (36). By Lemma 3, when $\hat{E}_n(t) \in [\mathcal{E}_n + P_{max}, \mathcal{E}_n + M]$, we have $E_n(t) \geq [\hat{E}_n(t) - \mathcal{E}_n]^+ \geq P_{max}$. Thus all the power allocations are valid under MESA. Now since at every time t, MESA performs ESA's data admission, and routing and scheduling actions, if there was no packet dropping, then MESA will achieve the same utility performance as ESA. However, since all the utility functions have bounded derivatives, to prove the utility performance of MESA, it suffices to show that the average rate of the packets dropped is $O(\epsilon) = O(1/V)$.

To prove this, we first see that packet dropping happens at time t only when the following event $\hat{\mathscr{E}}(t)$ happens, i.e.,

$$\hat{\mathscr{E}}(t) = \{\exists\, n, \hat{E}_n(t) < \mathcal{E}_n + P_{max}\} \qquad (53)$$
$$\cup\{\exists\, n, \hat{E}_n(t) > \mathcal{E}_n + M\}$$
$$\cup\{\exists\, (n,c), \hat{Q}_n^{(c)}(t) < \mathcal{Q}_n^{(c)}\}.$$

However, assuming (50)-(52) hold, the following event must happen for $\hat{\mathscr{E}}(t)$ to happen:

$$\tilde{\mathscr{E}}(t) = \{\exists n, \hat{E}_n(t) < (\theta_n + \nu_n^*) - \frac{3M}{8} + P_{max}\}$$
$$\cup\{\exists n, \hat{E}_n(t) > (\theta_n + \nu_n^*) + \frac{3M}{8}\}$$
$$\cup\{\exists (n,c), \hat{Q}_n^{(c)}(t) < \upsilon_n^{(c)*} - \frac{3M}{8}\}.$$

Therefore $\hat{\mathscr{E}}(t) \subset \tilde{\mathscr{E}}(t)$. However, it is easy to see from (31) that $\tilde{\mathscr{E}}(t) \subset \mathscr{E}(t, \tilde{m})$ with $\tilde{m} = (\frac{3M}{8} - P_{max} - D)/K = (\frac{3}{2}[\log(V)]^2 - P_{max} - D)/K$. Therefore $\hat{\mathscr{E}}(t) \subset \mathscr{E}(t, \tilde{m})$. Using (29) again, we see that:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} Pr(\hat{\mathscr{E}}(\tau)) \leq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} Pr(\mathscr{E}(\tau,\tilde{m}))$$
$$\leq c^* e^{-(\frac{3[\log(V)]^2}{2} - P_{max} - D)/K}.$$

Using the facts that $\frac{1}{2}[\log(V)]^2 \geq 2D$ and $\log(V) \geq 16K$, we see that:

$$\frac{\frac{3[\log(V)]^2}{2} - D}{K} \geq \frac{\frac{5[\log(V)]^2}{4}}{K} \geq 20\log(V).$$

Thus we conclude that:

$$\limsup_{t\to\infty} \sum_{\tau=0}^{t-1} Pr(\hat{\mathscr{E}}(\tau)) \leq \frac{c^* e^{P_{max}/K}}{V^{20}} = O(1/V^{20}).$$

Since at every time slot, the total amount of packets dropped is no more than $N(\mu_{max} + R_{max})$, we see that the average rate of packets dropped is $O(1/V)$. This completes the proof of Theorem 5. ∎

REFERENCES

[1] S. Meninger, J. O. Mur-Miranda, R. Amirtharajah, and A. Chandrakasan. 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 In 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] K. Lorincz G. Werner-Allen, 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, Article 32*, Sept. 2007.

[11] R. Srivastava and C. E. Koksal. Basic tradeoffs for energy management in rechargeable sensor networks. *arXiv tech report 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: 1010.1862v1*, 2010.

[20] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *IEEE INFOCOM Proceedings*, March 2005.

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

[23] V. Joseph, V. Sharma, and U. Mukherji. Optimal sleep-wake policies for an energy harvesting sensor node. *Proceedings of IEEE International Conference on Communications*, June 2009.

[24] L. Huang and M. J. Neely. Max-weight achieves the exact $[O(1/V), O(V)]$ utility-delay tradeoff under Markov dynamics. *arXiv:1008.0200v1*, 2010.

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