Max-min Fair Rate Allocation and Routing in Energy Harvesting Networks: Algorithmic Analysis

Jelena Marašević[1], Clifford Stein[2], Gil Zussman[1]
[1]Electrical Engineering, [2]Industrial Engineering and Operations Research Dept., Columbia University
New York, NY, 10027, USA
{jelena@ee, cliff@ieor, gil@ee}.columbia.edu

ABSTRACT

This paper considers max-min fair rate allocation and routing in energy harvesting networks where fairness is required among *both the nodes and the time slots*. Unlike most previous work on fairness, we focus on *multihop topologies* and consider *different routing methods*. We assume a predictable energy profile and focus on the design of efficient and optimal algorithms that can serve as benchmarks for distributed and approximate algorithms. We first develop an algorithm that obtains a max-min fair rate assignment for any given (time-variable or time-invariable) *unsplittable routing* or a *routing tree*. For *time-invariable unsplittable routing*, we also develop an algorithm that finds routes that maximize the minimum rate assigned to any node in any slot. For *fractional routing*, we study the joint routing and rate assignment problem. We develop an algorithm for the *time-invariable* case with constant rates. We show that the *time-variable* case is at least as hard as the 2-commodity feasible flow problem and design an FPTAS to combat the high running time. Finally, we show that finding a max-min fair unsplittable routing or a routing tree is NP-hard, even for a time horizon of a single slot. Our analysis provides insights into the problem structure and can be applied to other related fairness problems.

Categories and Subject Descriptors: C.2.1. [Computer-Communication Networks]: Network Architecture and Design — *Wireless Communication*
Keywords: Energy Harvesting; Energy Adaptive Networking; Sensor Networks; Routing; Fairness

1. INTRODUCTION

Recent advances in the development of ultra-low-power transceivers and energy harvesting devices (e.g., solar cells) will enable self-sustainable and perpetual wireless networks [12,13]. In contrast to legacy wireless sensor networks, where the available energy only decreases as the nodes sense and forward data, in energy harvesting networks the available energy can also increase through a replenishment process.

MobiHoc'14, August 11–14, 2014, Philadelphia, PA, USA.

ACM 978-1-4503-2620-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2632951.2632960 .

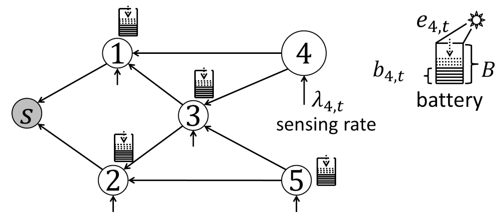


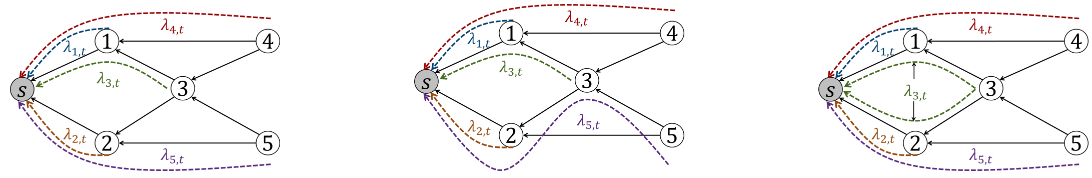
Figure 1: A simple energy harvesting network: the nodes sense the environment and forward the data to a sink s. Each node has a battery of capacity B. At time t a node i's battery level is $b_{i,t}$, it harvests $e_{i,t}$ units of energy, and senses at data rate $\lambda_{i,t}$.

As a result, the available energy is a more complex quantity, thereby posing challenges in the design of resource allocation and routing algorithms.

Two natural conditions that a network should satisfy are: (i) balanced data acquisition over all the parts of the network, and (ii) persistent operation (i.e., even when the environmental energy is not available for harvesting). Condition (i) is commonly reinforced by requiring fairness of the sensing rates over network nodes. One approach to achieving (ii) is by assigning constant sensing rates to the nodes. However, this approach can result in underutilization of the available energy. As a simple example, consider a node that harvests outdoor light energy over a 24-hour time horizon. If the battery capacity is small, then the sensing rate must be low to prevent battery depletion during the nighttime. However, during the daytime, when the harvesting rates are high, a low sensing rate prevents full utilization of the energy that can be harvested. Therefore, it is advantageous to vary the sensing rates over time. However, fairness must be required over time slots to prevent the rate assignment algorithm from assigning high rates during periods of high energy availability, and zero rates when no energy is available for harvesting.

The problems of resource allocation, scheduling, and routing in energy harvesting networks have received considerable attention [2, 4, 10, 11, 14–16, 19, 20, 25, 29]. Much of the existing work considers simple networks consisting of a single node or a link [2, 4, 11, 14, 25, 29]. Moreover, fair rate assignment has not been thoroughly studied, and most of the work either focuses on maximizing the total (or average) throughput [2, 4, 10, 16, 19, 22, 25, 29], or considers fairness either *only over nodes* [20] or *only over time* [11, 14]. An exception is [15], which requires fairness over both the nodes and the time, but is limited to *two nodes.*

In this paper, we study the max-min fair rate assignment and routing problems, requiring fairness over both nodes and



(a) Routing tree. (b) Unsplittable routing. (c) Fractional routing.

Figure 2: Routing types: (a) a routing tree, (b) unsplittable routing: each node sends its data over one path, (c) fractional routing: nodes can send their data over multiple paths. Paths are represented by dashed lines.

time slots, and with the goal of designing optimal and efficient algorithms. Following [11, 14, 15, 19, 20], we assume that the harvested energy is known for each node over a finite time horizon. Although there are settings that do not conform to this assumption, the designed algorithms can be used for: (i) determining (in an offline fashion) rate assignment and routing in a network with a highly-predictable energy profile, and (ii) benchmarking rate assignment and routing solutions in networks with unpredictable energy profiles. We consider an energy harvesting sensor network with a single sink node, and network connectivity modeled by a directed graph (Fig. 1). Each node senses some data from its surrounding (e.g., air pressure, temperature, radiation level), and sends it to the sink. The nodes spend their energy on sensing, sending, and receiving data.

We consider different routing types, which are illustrated in Fig. 2. Each routing type incurs different trade-offs between the supported sensing rates[1] and the required amount of control information. Routing types with higher number of active links require more control information to be exchanged between neighboring nodes (e.g., to maintain synchronization), and complicate the transmission and/or sleep-wake scheduling implementation. Moreover, energy consumed by the control messages can affect achievable rates significantly, due to limited energy budget, as confirmed via experiments in [13]. Below we outline the main characteristics of the routing types we consider.

Routing Tree–the simplest form of routing, in which every node sends all of the data it senses and receives to a single neighboring (parent) node. It requires minimum number of active links, yielding minimum energy consumption due to control messages. However, in general, it provides the lowest sensing rates (see more details below).

Unsplittable Routing–a single-path routing, in which every node sends all of its sensed data over a single path to the sink (a routing tree is a special case of the unsplittable routing, in which all the paths incoming into node i outgo via the same edge). There are simple cases in which unsplittable routing provides a rate assignment with the minimum sensing rate $\Omega(n)$ times higher than in a routing tree, where n is the number of nodes [23]. However, in general, it has higher number of active links than the routing tree, yielding higher energy consumption for control information.

Fractional Routing–a multi-path routing, in which each node can split its data over multiple paths to the sink (unsplittable routing is a special case of fractional routing in which every node has one path to the sink). It is the most general routing that subsumes both routing trees and unsplittable routings, and, therefore, provides the best sensing rates. However, it utilizes the highest number of links, yielding the highest energy consumption due to control messages.

[1]A metric of performance can be the minimum sensing rate that is assigned to any node in any time slot.

Time-invariable vs Time-variable Routing– A routing is *time-invariable*, if every node uses the same (set of) path(s) in each time slot to send its data to the sink. If the paths change over time, the routing is *time-variable*.[2] While there are cases in which time-variable routing provides a rate assignment with the minimum sensing rate $\Omega(n)$ times higher than in the time-invariable case [23], it requires substantial control information exchange for routing reconfigurations, yielding high energy consumption.

For the *unsplittable routing* and *routing tree*, we design a fully-combinatorial algorithm that solves the *max-min fair rate assignment* problem, both in the time-variable and time-invariable settings, when the routing is provided at the input. We then turn to *fractional routing*, considering two settings: time-variable and time-invariable. We demonstrate that in the *time-variable* setting verifying whether a given rate assignment is feasible is at least as hard as solving a feasible 2-commodity flow. This result implies that, to our current knowledge, it is unlikely that max-min fair time-variable fractional routing[3] can be solved without the use of linear programming. To combat the high running time induced by the linear programming, we develop a fully polynomial time approximation scheme (FPTAS). For the *time-invariable setting*, we provide a fully-combinatorial algorithm that determines a max-min fair routing with constant rates.

We show that *determining* a max-min fair unsplittable routing or a routing tree is NP-hard even for a single time slot. Relaxing the max-min fairness requirement, we develop an algorithm that determines a time-invariable unsplittable routing that maximizes the minimum sensing rate assigned to any node in any time slot.

The considered problems generalize classical max-min fair routing problems that have been studied outside the area of energy harvesting networks: max-min fair fractional routing [24], max-min fair unsplittable routing [18], and bottleneck routing [3]. In contrast to the problems studied in [3, 18, 24], our model allows different costs for flow generation and forwarding, and has time-variable node capacities determined by the available energies at the nodes. We note that studying networks with node capacities is as general as studying networks with capacitated edges, since there are standard methods for transforming one of these two problems into another (see, e.g., [1]). Therefore, we believe that the results can find applications in other related areas.

The rest of the paper is organized as follows. Section 2 provides the model and problem formulations, which are placed in the context of related work in Section 3. Section 4

[2]Whether the rates are constant or time-variable is independent of whether the routing is time-variable or not.

[3]We refer to a routing as max-min fair if it provides the lexicographically maximum rate assignment. The notions of max-min fairness and lexicographic maximization are defined in Section 4.

Table 1: Nomenclature.

inputs	n	Number of energy harvesting nodes
	T	Time horizon
	B	Battery capacity
	$b_{i,1}$	Initial battery level at node i
	$e_{i,t}$	Harvested energy at node i in time slot t
	c_{s}	Energy spent for sensing a unit flow
	c_{tx}	Energy spent for transmitting a unit flow
	c_{rx}	Energy spent for receiving a unit flow
variables	$\lambda_{i,t}$	Sensing rate of node i in time slot t
	$f_{ij,t}$	Flow on link (i,j) in time slot t
	$b_{i,t+1}$	Battery level at node i at the beginning of time slot $t+1$
notation	i	Node index, $i \in \{1,2,...n\}$
	t	Time index, $t \in \{1,2,...,T\}$
	c_{st}	Energy spent for jointly sensing and transmitting a unit flow: $c_{\mathrm{st}} = c_{\mathrm{s}} + c_{\mathrm{tx}}$
	c_{rt}	Energy spent for jointly receiving and transmitting a unit flow: $c_{\mathrm{rt}} = c_{\mathrm{rx}} + c_{\mathrm{tx}}$
	$f^{\Sigma}_{i,t}$	Total flow entering node i in time slot t: $f^{\Sigma}_{i,t} = \sum_{j:(j,i)\in E} f_{ji,t}$

describes the connection between max-min fairness and lexicographic maximization. Section 5 considers rate assignment in unsplittable routing, while Sections 6 and 7 study fractional routing and rate assignment in time-variable and time-invariable settings. Section 8 provides hardness results for determining unsplittable routing or a routing tree. Section 9 concludes the paper. Due to space constraints, most of the proofs are deferred to the technical report [23].

2. MODEL AND PROBLEM FORMULATION

We consider a network that consists of n energy harvesting nodes and one sink node (see Fig. 1). The sink is the central point at which all the sensed data is collected, and is assumed not to be energy constrained. In the rest of the paper, the term "sink" will be used for the sink node. The connectivity between the nodes is modeled by a directed graph $G = (V, E)$, where $|V| = n+1$ (n nodes and the sink), and $|E| = m$. The main notation is summarized in Table 1.

Each node is equipped with a rechargeable battery of finite capacity B. The time horizon is T time slots. The duration of a time slot is assumed to be much longer than the duration of a single data packet, but short enough so that the rate of energy harvesting does not change during a slot. For example, if outdoor light energy is harvested, one time slot can be at the order of a minute. In a time slot t, a node i harvests $e_{i,t}$ units of energy. The battery level of a node i at the beginning of a time slot t is $b_{i,t}$. We follow a predictable energy profile [11,14,15,19,20], and assume that the battery capacity B, initial battery levels $b_{i,1}$, and harvested energies $e_{i,t}$ are known and finite, for $i \in \{1, ..., n\}$, $t \in \{1, ..., T\}$.

A node i in slot t senses data at rate $\lambda_{i,t}$. A node forwards all the data it senses and receives towards the sink. The flow on a link (i, j) in slot t is denoted by $f_{ij,t}$. Each node spends c_{s} energy units to sense a unit flow, and c_{tx}, respectively c_{rx}, energy units to transmit, respectively receive, a unit flow.

The feasible region $\mathcal{R}$ for the sensing rates and flows is determined by the following set of linear constraints:

$$\forall i \in \{1, ..., n\}, t \in \{1, ..., T\}:$$

$$f^{\Sigma}_{i,t} + \lambda_{i,t} = \sum_{(i,j)\in E} f_{ij,t}, \quad (1)$$

$$b_{i,t+1} = \min\{B, \quad b_{i,t} + e_{i,t} - (c_{\mathrm{rt}} f^{\Sigma}_{i,t} + c_{\mathrm{st}}\lambda_{i,t})\}, \quad (2)$$

$$b_{i,t+1} \geq 0,\ \lambda_{i,t} \geq 0,\ f_{ij,t} \geq 0, \forall (i,j) \in E, \quad (3)$$

where $f^{\Sigma}_{i,t} \equiv \sum_{(j,i)\in E} f_{ji,t}$, $c_{\mathrm{st}} \equiv c_{\mathrm{s}} + c_{\mathrm{tx}}$, and $c_{\mathrm{rt}} \equiv c_{\mathrm{rx}} + c_{\mathrm{tx}}$. Eq. (1) is a classical flow conservation constraint, while (2) models battery evolution over time slots.[4]

Similar to the definition of max-min fairness in [3], we define a rate assignment $\{\lambda_{i,t}\}$, $i \in \{1, ..., n\}$, $t \in \{1, ..., T\}$, to be max-min fair if no $\lambda_{i,t}$ can be increased without either losing feasibility or decreasing some other rate $\lambda_{j,\tau} \leq \lambda_{i,t}$.

In some of the problems, the routing is provided at the input as a set of paths $\mathcal{P} = \{p_{i,t}\}$, for $i \in \{1, ..., n\}, t \in \{1, ..., T\}$. In such a case, $\mathcal{R}$ should be interpreted with respect to $\mathcal{P}$, instead with respect to the input graph G.

Considered problems (see Table 1 for inputs and variables). We examine different routing types, in time-variable and time-invariable settings, as described in the introduction. For the unsplittable routing and routing trees, we examine the problems of *determining a rate assignment* and *determining a routing* separately, as described below.

P-UNSPLITTABLE-RATES: For a given time-variable unsplittable routing $\mathcal{P} = \{p_{i,t}\}$, determine the max-min fair assignment of the rates $\{\lambda_{i,t}\}$. Note that this setting subsumes time-invariable unsplittable routing, time-invariable routing tree, and time-variable routing tree.

P-UNSPLITTABLE-FIND: Associate with each (time-invariable or time-variable) unsplittable routing $\mathcal{P}$, a set of sensing rates $\{\lambda^{\mathcal{P}}_{i,t}\}$ that optimally solves P-UNSPLITTABLE-RATES. Determine an unsplittable routing $\mathcal{P}$ that provides the lexicographically maximum[5] rate assignment $\{\lambda^{\mathcal{P}}_{i,t}\}$.

P-TREE-FIND: Let $\mathcal{T}$ denote a (time-invariable or time-variable) routing tree on the input graph G. Associate with each $\mathcal{T}$ a set of sensing rates $\{\lambda^{\mathcal{T}}_{i,t}\}$ that optimally solves P-UNSPLITTABLE-RATES. Determine $\mathcal{T}$ that provides the lexicographically maximum rate assignment $\{\lambda^{\mathcal{T}}_{i,t}\}$.

For the fractional routing, we study the following two variants of max-min fair routing, where the routing and the rate assignment are determined jointly.

P-FRACTIONAL: Determine a *time-variable* fractional routing with the max-min fair rate assignment $\{\lambda_{i,t}\}$.

P-FIXED-FRACTIONAL: Determine a *time-invariable* fractional routing with the max-min fair rate assignment $\{\lambda_{i,t}\} = \{\lambda_i\}$, when the sensing rates are constant over time.

3. RELATED WORK

Energy-harvesting Networks. Rate assignment in energy harvesting networks in the case of a single node or a link was studied in [2,4,11,14,25,29].

Resource allocation and scheduling for network-wide scenarios using Lyapunov optimization techniques was studied in [10,16,22]. While the work in [10,16,22] can support unpredictable energy profiles, it focuses on the (sum-utility of) time-average rates, which is, in general, time-unfair. The design of online algorithms for resource allocation and routing was studied in [9,19].

Max-min time-fair rate assignment for a single node or a link was studied in [11,14], while max-min fair energy allocation for single-hop and two-hop scenarios was studied in [15]. Similar to our work, [15] requires fairness over both the nodes and the time slots, but considers only two energy

[4] Eq. (2) is considered as a linear constraint, since for maximizing $\lambda_{i,t}$'s it can be replaced by $b_{i,t+1} \leq B$ and $b_{i,t+1} \leq b_{i,t} + e_{i,t} - (c_{\mathrm{rt}} f^{\Sigma}_{i,t} + c_{\mathrm{st}}\lambda_{i,t})$.

[5] Lexicographical ordering of vectors is defined in Section 4.

harvesting nodes. The work on max-min fairness in network-wide scenarios [20] is explained in more detail below.

Sensor Networks. Problems P-Fractional, P-Fixed-Fractional, and P-Unsplittable-Rates are related to the maximum lifetime routing problems (see, e.g., [6,21] and the follow-up work) in the following sense. In our model, maximization of the minimum sensing rate is equivalent to the network lifetime maximization in sensor networks, *but only if the system is observed for $T = 1$*. Namely, the nodes have the initial energy, and no harvesting happens over time.

Determining a maximum lifetime tree in sensor networks as in [5] is a special case of P-Tree-Find. We extend the NP-hardness result from [5] and provide a lower bound of $\Omega(\log n)$ for the approximation ratio (for both [5] and P-Tree-Find), where n is the number of nodes in the network.

Max-min Fair Unsplittable Routing. Rate assignment in unsplittable routing was studied extensively (see [3,7] and references therein). P-Unsplittable-Rates reduces to the problem studied in [3, 7] for $c_{\text{st}} = 0, c_{\text{rt}} > 0$, and $T = 1$. In the energy harvesting network setting, this problem has been studied in [20], for rates that are constant over time and a time-invariable routing tree. We consider a more general case than in [20], where the rates are time-variable, fairness is required over both network nodes and time slots, and the routing can be time-variable and given in a form of an unsplittable routing or a routing tree.

Determining a max-min fair unsplittable routing as studied in [18] is a special case of P-Unsplittable-Find for $c_{\text{st}} = 0, c_{\text{rt}} > 0$, and $T = 1$, and the NP-hardness results from [18] implies the NP-hardness of P-Unsplittable-Find.

Max-min Fair Fractional Routing. Max-min fair fractional routing was first studied in [24]. The algorithm from [24] relies on the property that the total values of a max-min fair flow and max flow are equal, which does not hold even in simple instances of P-Fixed-Fractional and P-Fractional. P-Fixed-Fractional and P-Fractional reduce to the problem of [24] for $c_{\text{st}} = 0, c_{\text{rt}} > 0$, and $T = 1$.

Max-min fair fractional routing in energy harvesting networks has been considered in [20]. The distributed algorithm from [20] solves P-Fixed-Fractional, but only as a heuristic. We provide a combinatorial algorithm that solves P-Fixed-Fractional optimally in a centralized manner.

A general linear programming framework for max-min fair routing was provided in [27], and extended to the setting of sensor and energy harvesting networks in [8] and [20], respectively. This framework, when applied to P-Fractional, is highly inefficient. P-Fractional reduces to [8] for $T = 1$, and to [20] when the rates are constant over time.

4. MAX-MIN FAIRNESS AND LEXICOGRAPHIC MAXIMIZATION

Recall that a rate assignment $\{\lambda_{i,t}\}$, $i \in \{1, ..., n\}$, $t \in \{1, ..., T\}$, is max-min fair if no rate $\lambda_{i,t}$ can be increased without either losing feasibility or decreasing some other rate $\lambda_{j,\tau} \leq \lambda_{i,t}$. Closely related to the max-min fairness is the notion of lexicographic maximization. The lexicographic ordering of vectors, with the relational operators denoted by $\overset{lex}{=}$, $\overset{lex}{>}$, and $\overset{lex}{<}$, is defined as follows:

Definition 4.1. *Let u and v be two vectors of the same length l, and let u_s and v_s denote the vectors obtained from u and v respectively by sorting their elements in the non-decreasing order. Then: (i) $u \overset{lex}{=} v$ if $u_s = v_s$ element-wise; (ii) $u \overset{lex}{>} v$ if there exists $j \in \{1, 2, ..., l\}$, such that $u_s(j) > v_s(j)$, and $u_s(1) = v_s(1), ..., u_s(j-1) = v_s(j-1)$ if $j > 1$; (iii) $u \overset{lex}{<} v$ if not $u \overset{lex}{=} v$ nor $u \overset{lex}{>} v$.*

It was proved in [27] that a max-min fair allocation vector exists on any convex and compact set. The results from [28] state that in a given optimization problem whenever a max-min fair vector exists, it is unique and equal to the lexicographically maximum one.

In the problems P-Unsplittable-Rates, P-Fractional, and P-Fixed-Fractional the feasible region $\mathcal{R}$ is determined by linear constraints (1)-(3), and it is therefore convex. As we are assuming that all the input values $B, e_{i,t}$, and $b_{i,1}$ are finite, it follows that the feasible region is also bounded, and therefore compact. Therefore, for the aforementioned problems, lexicographic maximization produces the max-min fair assignment of the sensing rates $\{\lambda_{i,t}\}$.

Lexicographic maximization can be implemented using the well-known water-filling framework (see, e.g., [3]):

Algorithm 1 Water-filling-Framework(G, b, e)

1: Set $\lambda_{i,t} = 0\ \forall i, t$, and mark all the rates as not fixed.
2: Increase all the rates $\lambda_{i,t}$ that are not fixed by the same maximum amount, subject to the constraints from $\mathcal{R}$.
3: Fix all the $\lambda_{i,t}$'s that cannot be further increased.
4: If all the rates are fixed, terminate. Else, go to step 2.

While the framework is well-known, it does not fully specify Steps 2 and 3. Implementation of these two steps is the main challenge in solving P-Unsplittable-Rates, P-Fractional, and P-Fixed-Fractional, for which we develop algorithms in the following sections. We refer to the algorithms that implement Steps 2 and 3 as Maximizing-the-Rates and Fixing-the-Rates, respectively.

Note: A rate $\lambda_{i,t}$ can in general get fixed in any iteration of the Water-filling-Framework; there is no rule that relates an iteration k to a node i or a time slot t.

5. RATES IN UNSPLITTABLE ROUTING

This section studies P-Unsplittable-Rates, the problem of rate assignment for an unsplittable routing provided at the input. The analysis applies to any time-invariable or time-variable unsplittable routing or a routing tree.

We assume that the routing over time $t \in \{1, ..., T\}$ is provided as a set of routing paths $\mathcal{P} = \{p_{i,t}\}$ from a node i to the sink s, for each node $i \in V \backslash \{s\}$. We say that a node j is a descendant of a node i in a time slot t if $i \in p_{j,t}$.[6]

Before describing the algorithms in detail, we need to introduce some notation. Let $F_{i,t}^k = 1$ if the rate $\lambda_{i,t}$ is not fixed at the beginning of the k^{th} iteration of Water-filling-Framework, $F_{i,t}^k = 0$ otherwise. Initially, $F_{i,t}^1 = 1$, $\forall i, t$. If a rate $\lambda_{i,t}$ is not fixed, we will say that it is "active". We will denote by $D_{i,t}^k$ the number of active descendants of the node i in the time slot t, where $D_{i,t}^1 = |\{j : i \in p_{j,t} \backslash \{j\}\}|$. Notice that $D_{i,t}^k = \sum_{j: i \in p_{j,t} \backslash \{j\}} F_{j,t}^k$. Finally, let $\lambda_{i,t}^k$ denote the value of $\lambda_{i,t}$ in the k^{th} iteration of Water-filling-Framework, and let $\lambda_{i,t}^0 = 0$, $\forall i, t$. Under this notation, the rates can be expressed as $\lambda_{i,t}^k = \sum_{l=1}^{k} F_{i,t}^l \lambda^l$, where λ^l

[6]Notice that this is consistent with the definition of a descendant in a routing tree.

denotes the common amount by which all the active rates get increased in the l^{th} iteration.

5.1 Maximizing the Rates

Maximization of the common rate λ^k in k^{th} iteration of WATER-FILLING-FRAMEWORK can be formulated as follows:

$$\begin{aligned}
\textbf{max}\ & \lambda^k \\
\textbf{s.t.}\ & \forall i \in \{1,...,n\}, t \in \{1,...,T\}: \\
& \lambda^k_{i,t} = \lambda^{k-1}_{i,t} + F^k_{i,t} \cdot \lambda^k, \\
& f^{\Sigma}_{i,t} + \lambda^k_{i,t} = \sum_{(i,j)\in E} f_{ij,t}, \\
& b_{i,t+1} = \min\{B,\ b_{i,t} + e_{i,t} - (c_{\text{rt}} f^{\Sigma}_{i,t} + c_{\text{st}}\lambda^k_{i,t})\}, \\
& b_{i,t} \ge 0,\ \lambda^k \ge 0,\ f_{ij,t} \ge 0, \forall (i,j) \in E.
\end{aligned}$$

As in each slot t every node i sends all the flow it senses over a single path, we can compute the total inflow into a node i as the sum of the flows coming from i's descendants:

$$\begin{aligned}
f^{\Sigma}_{i,t} &= \sum_{j: i \in p_{j,t}\setminus\{j\}} \sum_{l=1}^{k} F^l_{j,t} \cdot \lambda^l = \sum_{l=1}^{k} \lambda^l \sum_{j: i \in p_{j,t}\setminus\{j\}} F^l_{j,t} \\
&= \sum_{l=1}^{k} D^l_{i,t} \cdot \lambda^l.
\end{aligned}$$

Denoting the battery levels in the iteration k as $b^k_{i,t}$, the problem can now be written more compactly as:

$$\begin{aligned}
\textbf{max}\ & \lambda^k \\
\textbf{s.t.}\ & \forall i \in \{1,...,n\}, t \in \{1,...,T\}: \\
& b^k_{i,t+1} = \min\{B,\ b^k_{i,t} + e_{i,t} - \sum_{l=1}^{k} \lambda^l (c_{\text{rt}} D^l_{i,t} + c_{\text{st}} F^l_{i,t})\}, \\
& b^k_{i,t} \ge 0,\ \lambda^k \ge 0,
\end{aligned}$$

where $\forall i\, \forall k: b^k_{i,1} = b_{i,1}$.

Define the battery drop for node i in slot t and iteration k as: $\Delta b^k_{i,t} = \sum_{l=1}^{k} \lambda^l (c_{\text{rt}} D^l_{i,t} + c_{\text{st}} F^l_{i,t})$, setting $\Delta b^0_{i,t} = 0$. The intuition is: to determine the battery levels in all the time slots, we only need to know the initial battery level and how much energy ($\Delta b_{i,t}$) is spent per time slot. This results in:

$$\begin{aligned}
\textbf{max}\ & \lambda^k \\
\textbf{s.t.}\ & \forall i \in \{1,...,n\}, t \in \{1,...,T\}: \\
& \Delta b^k_{i,t} = \Delta b^{k-1}_{i,t} + \lambda^k (c_{\text{rt}} D^k_{i,t} + c_{\text{st}} F^k_{i,t}), \\
& b^k_{i,t+1} = \min\{B,\ b^k_{i,t} + e_{i,t} - \Delta b^k_{i,t}\}, \\
& b^k_{i,t} \ge 0,\ \lambda^k \ge 0.
\end{aligned}$$

Writing the problem for each node independently, we can solve the following subproblem:

$$\textbf{max}\ \lambda^k_i \tag{4}$$

$$\textbf{s.t.}\ \forall t \in \{1,...,T\}:$$

$$\Delta b^k_{i,t} = \Delta b^{k-1}_{i,t} + \lambda^k_i (c_{\text{rt}} D^k_{i,t} + c_{\text{st}} F^k_{i,t}), \tag{5}$$

$$b^k_{i,t+1} = \min\{B,\ b^k_{i,t} + e_{i,t} - \Delta b^k_{i,t}\}, \tag{6}$$

$$b^k_{i,t} \ge 0,\ \lambda^k_i \ge 0, \tag{7}$$

for each i with $\sum_{i,t} F^k_{i,t} > 0$, and determine $\lambda^k = \min_i \lambda^k_i$. Notice that we can bound each λ^k_i by the interval $[0, \lambda^k_{\max,i}]$, where $\lambda^k_{\max,i}$ is the rate for which node i spends all its available energy in the first slot τ in which its rate is not fixed:

$$\lambda^k_{\max,i} = \frac{b^{k-1}_{i,\tau} + e_{i,\tau}}{c_{\text{rt}} D^k_{i,\tau} + c_{\text{st}}}, \quad \tau = \min\{t : F^k_{i,t} = 1\}.$$

The subproblem of determining λ^k_i can now be solved by performing a binary search in the interval $[0, \lambda^k_{\max,i}]$.

Let δ denote the precision of the input variables. Note that however small, δ can usually be expressed as a constant.

Lemma 5.1. MAXIMIZING-THE-RATES *in* P-UNSPLITTABLE-FIND *can be implemented in time* $O\left(T \sum_i \log\left(\lambda^k_{\max,i}/\delta\right)\right) = O\left(nT \log\left((B + \max_{i,t} e_{i,t})/(\delta c_{st})\right)\right)$.

5.2 Fixing the Rates

Recall that the elements of the matrix F^k are such that $F^k_{i,t} = 0$ if the rate $\lambda_{i,t}$ is fixed for the iteration k, and $F^k_{i,t} = 1$ otherwise. At the end of iteration $k \ge 1$, let $F^{k+1} = F^k$, and consider the following set of rules for fixing the rates:

(F1) For all (i,t) such that $b^k_{i,t+1} = 0$ set $F^{k+1}_{i,t} = 0$.

(F2) For all (i,t) such that $b^k_{i,t+1} = 0$ determine the longest sequence $(i,t), (i,t-1), (i,t-2), ..., (i,\tau), \tau \ge 1$, with the property that $b^k_{i,s} + e_{i,s} - \Delta b^k_{i,s} \le B\ \forall s \in \{t, t-1, ..., \tau\}$, and set $F^{k+1}_{i,s} = 0\ \forall s$.

(F3) For all (i,t) for which the rules (F1) and (F2) have set $F^{k+1}_{i,t} = 0$, and for all j such that $i \in p_{j,t}$, set $F^{k+1}_{j,t} = 0$.

The correctness of the rules (F1)-(F3) is proved via the following two lemmas.

Lemma 5.2. *(Necessity) No rate fixed by the rules (F1), (F2) and (F3) can be increased in the next iteration without violating feasibility constraints.*

Proof. We first make the following two observations. When $b^k_{i,t+1} = 0$, then, from (6):

$$\begin{aligned}
b^k_{i,t+1} &= \min\{B,\ b^k_{i,t} + e_{i,t} - (c_{\text{rt}} \sum_{j: i \in p_{j,t}\setminus\{j\}} \lambda^k_{j,t} + c_{\text{st}} \lambda^k_{i,t})\} \\
&= b^k_{i,t} + e_{i,t} - (c_{\text{rt}} \sum_{j: i \in p_{j,t}\setminus\{j\}} \lambda^k_{j,t} + c_{\text{st}} \lambda^k_{i,t}) = 0. \quad (8)
\end{aligned}$$

For $b^k_{i,t+1} = 0$, let $(i,t), (i,t-1), (i,t-2), ..., (i,\tau), \tau \ge 1$, be the longest sequence with the property that: $b^k_{i,s} + e_{i,s} - \Delta b^k_{i,s} \le B\ \forall s \in \{t, t-1, ..., \tau\}$. Then $\forall s \in \{\tau, \tau+1, ..., t-1\}$:

$$b^k_{i,s+1} = b^k_{i,s} + e_{i,s} - (c_{\text{rt}} \sum_{j: i \in p_{j,t}\setminus\{j\}} \lambda^k_{j,s} + c_{\text{st}} \lambda^k_{i,s}).$$

This is a recursive relation on s, so we can write $b_{i,t+1}$ as:

$$b^k_{i,t+1} = b^k_{i,\tau} + \sum_{s=\tau}^{t} e_{i,s} - \sum_{s=\tau}^{t} (c_{\text{rt}} \sum_{j: i \in p_{j,t}\setminus\{j\}} \lambda^k_{j,s} - c_{\text{st}} \lambda^k_{i,s}). \quad (9)$$

The rest of the proof is by induction on iteration k.

The base case. In the first iteration, $\lambda^1_{i,t} = \lambda^1$, $\forall i, t$. Suppose $b^1_{i,t+1} = 0$ for some i, t. From (8), if $\lambda_{i,t}$ is increased, either $b^1_{i,t+1} < 0$, or some $\lambda_{j,t}$, with $i \in p_{j,t}\setminus\{j\}$, must be decreased. In the former, feasibility is lost. In the latter, max-min fairness does not hold. Similarly if $\lambda_{j,t}$, with $i \in p_{j,t}\setminus\{j\}$ is increased. This proves rule (F1), and rule (F3) for the descendants of node i in time slot t.

Now, for $b^1_{i,t+1} = 0$, let $(i,t), (i,t-1), (i,t-2), ..., (i,\tau), \tau \ge 1$, be the longest sequence with the property that: $b^k_{i,s} + e_{i,s} - \Delta b^k_{i,s} \le B\ \forall s \in \{t, t-1, ..., \tau\}$. From (9), if any of the rates $\lambda_{i,s}$, or $\lambda_{j,s}$, with $i \in p_{j,s}\setminus\{j\}$ is increased, either $b^1_{i,t+1} < 0$, or some other rate from (9) needs to be decreased, which violates the max-min fairness, since all the rates are equal. This proves rule (F2), and completes the proof for rule (F3).

The inductive step. Observe that:

(o1) $\lambda_{j,t} \leq \lambda_{i,t}$, $\forall j : i \in p_{j,t}$, as all the rates, until fixed, get increased by the same amount in each iteration, and once a rate gets fixed for some (i, t), by the rule (F3), it gets fixed for all the (j, t) with $i \in p_{j,t}\backslash\{j\}$. The inequality is strict only if $\lambda_{j,t}$ got fixed before $\lambda_{i,t}$.

(o2) Once fixed, a rate never becomes active again.

(o3) If a rate $\lambda_{i,t}$ gets fixed in iteration k, then $\lambda_{i,t} = \lambda^k_{i,t} = \sum_{p=1}^{k} \lambda^p = \lambda^l_{i,t}$, $\forall l \geq k$.

Suppose that $b^k_{i,t+1} = 0$ for some $i \in \{1, .., n\}, t \in \{1, ..., T\}$. If $F^k_{i,t} = 0$, then by the inductive hypothesis $\lambda_{i,t}$ cannot be further increased in any future iteration. Assume $F^k_{i,t} = 1$.

By (o1), $\lambda^k_{j,t} \leq \lambda^k_{i,t}$, $\forall j$ such that $i \in p_{j,t}\backslash\{j\}$, where the inequality holds with equality if $F^k_{j,t} = 0$. Therefore, from (8), if we increase $\lambda_{i,t}$ in some of the future iterations, either $b_{i,t+1} < 0$, or we need to decrease some $\lambda_{j,t} \leq \lambda_{i,t}$, violating the max-min fairness condition. This proves the necessity of the rule (F1). For the rule (F3), as for all (j, t) with $F^k_{j,t} = 1$, $i \in p_{j,t}\backslash\{j\}$, we have $\lambda_{j,t} = \lambda_{i,t}$, none of the i's descendants can further increase its rate in the slot t.

Now for (i, t) such that $b^k_{i,t+1} = 0$, let $(i, t), (i, t-1), (i, t-2), ..., (i, \tau), \tau \geq 1$, be the longest sequence with the property that: $b^k_{i,s} + e_{i,s} - \Delta b^k_{i,s} \leq B$ $\forall s \in \{t, t-1, ..., \tau\}$. If any of the rates appearing in (9) was fixed in some previous iteration, then it cannot be further increased by the inductive hypothesis. By the observation (o1), all the rates that are active are equal, and all the rates that are fixed are strictly lower than the active rates. Therefore, by increasing any of the active rates from (9), we either violate battery nonnegativity constraint or the max-min fairness condition. Therefore, rule (F2) holds, and rule (F3) holds for all the descendants of nodes whose rates got fixed by the rule (F2), in the corresponding time slots. □

Lemma 5.3. *(Sufficiency) If $F^{k+1}_{i,t} = 1$, then $\lambda_{i,t}$ can be further increased by a positive amount in the iteration $k+1$, $\forall i \in \{1, ..., n\}$, $\forall t \in \{1, ..., T\}$.*

Proof. Suppose that $F^{k+1}_{i,t} = 1$. Notice that by increasing $\lambda_{i,t}$ by some $\Delta\lambda_{i,t}$ node i spends an additional $\Delta b_{i,t} = c_{\rm st}\Delta\lambda_{i,t}$ energy *only in the time slot t*. As $F^{k+1}_{i,t} = 1$, by the rules (F1) and (F2), either $b_{i,t'} > 0$ $\forall t' > t$, or there is a time slot $s > t$ such that $b^k_{i,s} + e_{i,s} - \Delta b^k_{i,s} > B$ and $s < s'$, where $s' = \arg\min\{\tau > t : b_{i,\tau} = 0\}$.

If $b_{i,t'} > 0$ $\forall t' > t$, then the node i can spend $\Delta b_{i,t} = \min_{t+1\leq t'\leq T+1} b^k_{i,t'}$ energy, and keep $b_{i,t'} \geq 0$, $\forall t'$, which follows from the battery evolution equation (6).

If there is a slot $s' > t$ in which $b^k_{i,s'} = 0$, then let s be the minimum time slot between t and s', such that $b^k_{i,s} + e_{i,s} - \Delta b^k_{i,s} > B$. Decreasing the battery level at s by $(b^k_{i,s} + e_{i,s} - \Delta b^k_{i,s}) - B$ does not influence any other battery levels, as in either case $b_{i,s+1} = B$. As all the battery levels are positive in all the time slots between t and s, i can spend at least $\min\{(b^k_{i,s} + e_{i,s} - \Delta b^k_{i,s}) - B, \quad \min_{t+1\leq t'\leq s} b^k_{i,t'}\}$ energy at the time t and have $b_{i,t'} \geq 0$ $\forall t'$.

By the rule (F3), $\forall j$ such that $j \in p_{i,t}$ we have that $b_{j,t} > 0$, and, furthermore, if $\exists s' > t$ with $b_{j,s'} = 0$ then $\exists s \in \{t, s'\}$ such that $b^k_{i,s} + e_{i,s} - \Delta b^k_{i,s} > B$. By the same observations as for the node i, each $j \in p_{i,t}$ can spend some extra energy $\Delta b_{j,t} > 0$ in the time slot t and keep all the battery levels nonnegative. In other words, there is a directed path from the node i to the sink on which every node can spend some extra energy in time slot t and keep its battery levels nonnegative. Therefore, if we keep all other rates fixed, the rate $\lambda_{i,t}$ can be increased by $\Delta\lambda_{i,t} = \min\{\Delta b_{i,t}/c_{\rm st}, \min_{j\in p_{i,t}} \Delta b_{j,t}/c_{\rm rt}\} > 0$.

As each active rate $\lambda_{i,t}$ can (alone) get increased in the iteration $k + 1$ by some $\Delta\lambda_{i,t} > 0$, it follows that all the active rates can be increased simultaneously by at least $\min_{i,t} \Delta\lambda_{i,t}/(T(c_{\rm st} + nc_{\rm rt})) > 0$. □

Due to space constraints, the proofs are omitted in the rest of the paper and can instead be found in [23].

Theorem 5.4. *Fixing rules (F1), (F2) and (F3) provide necessary and sufficient conditions for fixing the sensing rates in* WATER-FILLING-FRAMEWORK.

Lemma 5.5. *The total running time of* FIXING-THE-RATES *in* P-UNSPLITTABLE-FIND *is $O(mT)$.*

Combining lemmas 5.1 and 5.5, we can compute the total running time of WATER-FILLING-FRAMEWORK for P-UNSPLITTABLE-FIND, as stated in the following lemma.

Lemma 5.6. WATER-FILLING-FRAMEWORK *with Steps 2* MAXIMIZING-THE-RATES *and 3* FIXING-THE-RATES *implemented as described in Section 5 runs in time:*

$$O\left(n^2T^2 \log\left(\frac{B + \max_{i,t} e_{i,t}}{\delta c_{st}}\right) + nmT^2\right).$$

6. FRACTIONAL ROUTING

The feasible region $\mathcal{R}$ for the rates and flows in P-FRACTIONAL can be described by the following constraints:

$$\forall i \in \{1, ..., n\}, t \in \{1, ..., T\} :$$
$$f^{\Sigma}_{i,t} + \lambda_{i,t} = \sum_{(i,j)\in E} f_{ij,t},$$
$$b_{i,t+1} = \min\{B,\ b_{i,t} + e_{i,t} - (c_{\rm rt} f^{\Sigma}_{i,t} + c_{\rm st}\lambda_{i,t})\},$$
$$b_{i,t} \geq 0,\ \lambda_{i,t} \geq 0,\ f_{ij,t} \geq 0, \forall (i, j) \in E,$$

where $f^{\Sigma}_{i,t} \equiv \sum_{(j,i)\in E} f_{ji,t}$.

Observe that we can avoid computing the values of battery levels $b_{i,t+1}$, and instead explicitly write the non-negativity constraints for each of the terms inside the min. Reordering the terms, we get the following formulation:

$$\forall i \in \{1, ..., n\}, t \in \{1, ..., T\} :$$
$$f^{\Sigma}_{i,t} + \lambda_{i,t} = \sum_{(i,j)\in E} f_{ij,t}, \tag{10}$$
$$\sum_{\tau=1}^{t} (c_{\rm rt} f^{\Sigma}_{i,\tau} + c_{\rm st}\lambda_{i,t}) \leq b_{i,1} + \sum_{\tau=1}^{t} e_{i,\tau}, \tag{11}$$
$$\sum_{\tau=s}^{t} (c_{\rm rt} f^{\Sigma}_{i,\tau} + c_{\rm st}\lambda_{i,t}) \leq B + \sum_{\tau=s}^{t} e_{i,\tau}, \quad 2 \leq s \leq t, \tag{12}$$
$$\lambda_{i,t} \geq 0,\ f_{ij,t} \geq 0, \forall (i, j) \in E. \tag{13}$$

In the $k^{\rm th}$ iteration of WATER-FILLING-FRAMEWORK we have that $\lambda^k_{i,t} = \lambda^{k-1}_{i,t} + F^k_{i,t} \cdot \lambda^k = \sum_{l=1}^{k} F^l_{i,t} \cdot \lambda^l$, where $\lambda^0_{i,t} = 0$. Let:

$$u^b_{i,t} = b_{i,1} + \sum_{\tau=1}^{t} (e_{i,\tau} - c_{\rm st}\lambda^{k-1}_{i,\tau}),\ u^B_{i,t,s} = B + \sum_{\tau=s}^{t} (e_{i,\tau} - c_{\rm st}\lambda^{k-1}_{i,\tau}).$$

Since in the iteration k all $\lambda^{k-1}_{i,t}$'s are constants, the rate maximization subproblem can be written as:

$$\textbf{max}\ \ \lambda^k \tag{14}$$
$$\textbf{s.t.}\ \ \forall i \in \{1, ..., n\}, t \in \{1, ..., T\} :$$
$$-f^{\Sigma}_{i,t} - F^k_{i,t} \cdot \lambda^k + \sum_{(i,j)\in E} f_{ij,t} = \lambda^{k-1}_{i,t}, \tag{15}$$

$$\sum_{\tau=1}^{t} (c_{\rm rt} f^{\Sigma}_{i,\tau} + F^k_{i,\tau} \cdot c_{\rm st}\lambda^k) \le u^b_{i,t}, \qquad (16)$$

$$\sum_{\tau=s}^{t} (c_{\rm rt} f^{\Sigma}_{i,\tau} + F^k_{i,\tau} \cdot c_{\rm st}\lambda^k) \le u^B_{i,t,s}, \quad 2 \le s \le t, \qquad (17)$$

$$\lambda^k \ge 0,\ f_{ij,t} \ge 0, \forall (i,j) \in E. \qquad (18)$$

Notice that in this formulation all the variables are on the left-hand side of the constraints, whereas all the right-hand sides are constant.

6.1 Relation to Multi-commodity Flow

Let $T = 2$, and observe the constraints (10)–(13). We claim that verifying whether any set of sensing rates $\lambda_{i,t}$ is feasible is at least as hard as solving a 2-commodity feasible flow problem with capacitated nodes and a single sink. To prove the claim, we first rewrite (10)–(13) as:

$$\sum_{(j,i)\in E} f_{ji,t} + \lambda_{i,t} = \sum_{(i,j)\in E} f_{ij,t}, \qquad t \in \{1,2\}$$

$$c_{\rm rt} \sum_{(j,i)\in E} f_{ji,1} \le b_{i,1} + e_{i,1} - c_{\rm st}\lambda_{i,1},$$

$$c_{\rm rt} \sum_{\tau=1}^{2} \sum_{(j,i)\in E} f_{ji,\tau} \le b_{i,1} + \sum_{\tau=1}^{2} \left(e_{i,\tau} - c_{\rm st}\lambda_{i,\tau}\right),$$

$$c_{\rm rt} \sum_{(j,i)\in E} f_{ji,2} \le B + e_{i,2} - c_{\rm st}\lambda_{i,2},$$

$$\lambda_{i,t} \ge 0, f_{ij,t} \ge 0,\ \forall i \in \{1,...,n\}, (i,j) \in E, t \in \{1,2\}.$$

Suppose that we are given any 2-commodity flow problem with capacitated nodes[7], and let:

- $\lambda_{i,t}$ denote the supply of commodity t at node i;
- $u_{i,t}$ denote the per-commodity capacity constraint at node i for commodity t;
- u_i denote the bundle capacity constraint at node i.

Given $\lambda_{i,t}$, $u_{i,t}$, and u_i, for $i \in \{1,...,n\}, t \in \{1,2\}$, choose values of $c_{\rm st}, c_{\rm rt}, B, b_{i,1}, b_{i,2}, e_{i,1}, e_{i,2}$ so that the following equalities are satisfied:

$$u_{i,1} = \left(b_{i,1} + e_{i,1} - c_{\rm st}\lambda_{i,1}\right)/c_{\rm rt},$$
$$u_{i,2} = \left(B + e_{i,2} - c_{\rm st}\lambda_{i,2}\right)/c_{\rm rt},$$
$$u_i = \left(b_{i,1} + \sum_{\tau=1}^{2}\left(e_{i,\tau} - c_{\rm st}\lambda_{i,\tau}\right)\right)/c_{\rm rt}.$$

Then feasibility of the given 2-commodity flow problem is equivalent to the feasibility of (10)–(13). Therefore, any 2-commodity feasible flow problem can be stated as an equivalent problem of verifying feasibility of sensing rates $\lambda_{i,t}$ in an energy harvesting network for $T = 2$.

For $T > 2$, (11) and (12) are general packing constraints. If a flow graph G_t in time slot t is observed as a flow of a commodity indexed by t, then for each node i the constraints (11) and (12) define capacity constraints for every sequence of consecutive commodities $s, s+1, ..., t$, $1 \le s \le t \le T$.

Therefore, to our current knowledge, it is unlikely that the general rate assignment problem can be solved exactly in polynomial time without the use of linear programming, as there have not been any combinatorial algorithms that solve feasible 2-commodity flow in directed graphs exactly.

6.2 Fractional Packing Approach

The fractional packing problem is defined as follows [26]: PACKING: *Given a convex set P for which $Ax \ge 0$ $\forall x \in P$, is there a vector x such that $Ax \le b$?* Here, A is a $p \times q$ matrix, and x is a q-length vector.

A given vector x is an ϵ-approximate solution to the PACKING problem if $x \in P$ and $Ax \le (1+\epsilon)b$. Alternatively, scaling all the constraints by $\frac{1}{1+\epsilon}$, we obtain a solution $x' = \frac{1}{1+\epsilon}x \in (\frac{1}{1+\epsilon}x_{\rm OPT}, x_{\rm OPT}] \subset ((1-\epsilon)x_{\rm OPT}, x_{\rm OPT}]$, for $\epsilon < 1$, where $x_{\rm OPT}$ is an optimal solution to the packing problem. The algorithm in [26] either provides an ϵ-approximate solution to the PACKING problem, or it proves that no such solution exists. It's running time depends on:

- The running time required to solve $\min\{cx : x \in P\}$, where $c = y^T A$, y is a given p-length vector, and $(.)^T$ denotes the transpose of a vector.
- The width of P relative to $Ax \le b$, which is defined by $\rho = \max_i \max_{x\in P} \frac{a_i x}{b_i}$, where a_i is the $i^{\rm th}$ row of A, and b_i is the $i^{\rm th}$ element of b.

For a given error parameter $\epsilon > 0$, a feasible solution to the problem $\min\{\beta : Ax \le \beta b, x \in P\}$, its dual solution y, and $C_{\mathcal{P}}(y) = \min\{cx : c = y^T A, x \in P\}$, [26] defines the following relaxed optimality conditions:

$$(1-\epsilon)\beta y^T b \le y^T Ax \qquad (\mathcal{P}1),$$

$$y^T Ax - C_{\mathcal{P}}(y) \le \epsilon(y^T Ax + \beta y^T b) \qquad (\mathcal{P}2).$$

The packing algorithm [26] is implemented through subsequent calls to the procedure IMPROVE-PACKING:

Algorithm 2 IMPROVE-PACKING(x, ϵ) [26]

```
Initialize β0 = max_i a_i x/b_i; α = 4 β0^{-1} ε^{-1} ln(2pε^{-1}); σ = ε/(4αρ).
while max_i a_i x/b_i ≥ β0/2 and x, y do not satisfy (P2) do
    For each i = 1, 2, ..., p: set y_i = (1/b_i) e^{α a_i x / b_i}.
    Find a min-cost point x̃ ∈ P for costs c = y^T A.
    Update x = (1 − σ)x + σx̃.
return x.
```

The running time of the ϵ-approximation algorithm provided in [26], for $\epsilon \in (0,1]$, equals $O(\epsilon^{-2}\rho\log(m\epsilon^{-1}))$ multiplied by the time needed to solve $\min\{cx : c = y^T A, x \in P\}$ and compute Ax (Theorem 2.5 in [26]).

6.2.1 Maximizing the Rates as Fractional Packing

We demonstrated at the beginning of this section that for the $k^{\rm th}$ iteration MAXIMIZE-THE-RATES can be stated as (14)-(18). Observe the constraints (16) and (17). Since λ^k, $f_{ij,t}$ and all the right-hand sides in (16) and (17) are nonnegative, (16) and (17) imply the following inequalities:

$\forall i \in \{1,...,n\}, t \in \{1,...,T\}$:

$$F^k_{i,\theta} \cdot c_{\rm st}\lambda^k \le u^b_{i,t}, \qquad 1 \le \theta \le t,$$

$$F^k_{i,\theta} \cdot c_{\rm st}\lambda^k \le u^B_{i,t,s}, \qquad 2 \le s \le t,\ s \le \theta \le t,$$

$$c_{\rm rt} \sum_{(j,i)\in E} f_{ji,\theta} \le u^b_{i,t} - c_{\rm st}\sum_{\tau=1}^{t} F^k_{i,\tau}\lambda^k, \qquad 1 \le \theta \le t,$$

$$c_{\rm rt} \sum_{(j,i)\in E} f_{ji,\theta} \le u^B_{i,t,s} - c_{\rm st}\sum_{\tau=s}^{t} F^k_{i,\tau}\lambda^k,\ 2 \le s \le t,\ s \le \theta \le t.$$

Therefore, we can yield an upper bound $\lambda^k_{\max}$ for λ^k:

$$\lambda^k \le \lambda^k_{\max} \equiv$$

$$\frac{1}{c_{\rm st}} \min_{i,t,s\ge 2}\{u^b_{i,t} : \sum_{\tau=1}^{t} F^k_{i,\tau} > 0, u^B_{i,t,s} : \sum_{\tau=s}^{t} F^k_{i,\tau} > 0\}.$$

[7]For the definition of a multi-commodity flow, see e.g., [1].

For a fixed λ^k, the flow entering a node i at time slot t can be bounded as:

$$\sum_{(j,i)\in E} f_{ji,t} \le u_{i,t} \equiv$$
$$\frac{1}{c_{\mathrm{rt}}} \min_{\substack{i,t_1\ge t\\ s\ge 2}} \{u^b_{i,t_1} - c_{\mathrm{st}}\sum_{\tau=1}^{t_1} F^k_{i,\tau}\lambda^k, u^B_{i,t,s} - c_{\mathrm{st}}\sum_{\tau=s}^{t_1} F^k_{i,\tau}\lambda^k\}.$$

We choose to keep only the flows $f_{ij,t}$ as variables in the Packing problem. Given a $\lambda^k \in [0, \lambda^k_{\max}]$, we define the convex set P via the following set of constrains[8]:

$$\forall i \in \{1,...,n\}, t \in \{1,...,T\}:$$
$$-\sum_{(j,i)\in E} f_{ji,t} + \sum_{(i,j)\in E} f_{ij,t} = \lambda^{k-1}_{i,t} + F^k_{i,t}\cdot\lambda^k, \quad (19)$$
$$\sum_{(j,i)\in E} f_{ji,t} \le u_{i,t}, \quad (20)$$
$$f_{ij,t} \ge 0, \qquad \forall (i,j)\in E. \quad (21)$$

Proposition 6.1. *For P described by (19) – (21) and a given vector y, problem $\min\{cf : c = y^T Af,\ f\in P\}$ reduces to T min-cost flow problems.*

The remaining packing constraints of the form $Ax \le b$ are given by (16) and (17), where $x \equiv f$.

Proposition 6.2. $Ax \ge 0\ \forall f \in P$.

Lemma 6.3. *One iteration of* Improve-Packing *for* P-Fractional *can be implemented in time*

$$O\left(nT^2 + T\cdot MCF(n,m)\right),$$

where $MCF(n,m)$ denotes the running time of a min-cost flow algorithm on a graph with n nodes and m edges.

Lemma 6.4. *Width ρ of P relative to the packing constraints (16) and (17) is $O(T)$.*

Lemma 6.5. Maximizing-the-Rates *that uses packing algorithm from [26] can be implemented in time: $\tilde{O}(T^2\epsilon^{-2}\cdot(nT+MCF(n,m)))$, where $\tilde{O}$-notation ignores poly-log terms.*

6.2.2 Fixing the Rates

As Maximizing-the-Rates described in previous subsection outputs an ϵ-approximate solution in each iteration, the objective of the algorithm is not to output a max-min fair solution anymore, but an ϵ-approximation. We consider the following notion of approximation, as in [18]:

Definition 6.6. *For a problem of lexicographic maximization, say that a feasible solution given as a vector v is an element-wise ϵ-approximate solution, if for vectors v and v_{OPT} sorted in nondecreasing order $v \ge (1-\epsilon)v_{OPT}$ component-wise, where v_{OPT} is an optimal solution to the given lexicographic maximization problem.*

Let Δ be the smallest real number that can be represented in a computer, and consider the algorithm that implements Fixing-the-Rates as stated below.

Assume that Fixing-the-Rates does not change any of the rates, but only determines what rates should be fixed in the next iteration, i.e., it only makes (global) changes to $F^{k+1}_{i,t}$. Then:

Lemma 6.7. *If the Steps 2 and 3 in the* Water-filling-Framework *are implemented as* Maximizing-the-Rates *and* Fixing-the-Rates *from this section, then the solution output by the algorithm is an element-wise ϵ-approximate solution to the lexicographic maximization of $\lambda_{i,t}\in\mathcal{R}$.*

[8] P is determined by linear equalities and inequalities, which implies that it is convex.

Algorithm 3 Fixing-the-Rates

Solve the following linear program:
max $\sum_{i=1}^n F^k_{i,t}\lambda^k_{i,t}$
s.t. $\forall i \in \{1,...,n\}, t\in\{1,...,T\}$:
  $\lambda^k_{i,t} \ge \lambda^{k-1}_{i,t} + F^k_{i,t}\cdot\lambda^k$,
  $\lambda^k_{i,t} \le \lambda^{k-1}_{i,t} + F^k_{i,t}\cdot\left(\epsilon\lambda^{k-1}_{i,t} + (1+\epsilon)\lambda^k + \Delta\right)$,
  $f^{\Sigma}_{i,t} + \lambda^k_{i,t} = \sum_{(i,j)\in E} f_{ij,t}$,
  $b_{i,t+1} = \min\left\{B,\ b_{i,t} + e_{i,t} - \left(c_{\mathrm{rt}} f^{\Sigma}_{i,t} + c_{\mathrm{st}}\lambda^k_{i,t}\right)\right\}$,
  $b_{i,t}\ge 0, \quad \lambda^k_{i,t}\ge 0, \quad f_{ij,t}\ge 0.$
Let $F^{k+1}_{i,t} = F^k_{i,t}$, $\forall i,t$.
If $\lambda^k_{i,t} < (1+\epsilon)(\lambda^{k-1}_{i,t} + F^k_{i,t}\cdot\lambda^k) + \Delta$, set $F^{k+1}_{i,t} = 0$.

Lemma 6.8. *An FPTAS for* P-Fractional *can be implemented in time:*

$$\tilde{O}(nT(T^2\epsilon^{-2}\cdot(nT + MCF(n,m)) + LP(mT,nT))),$$

where $LP(mT,nT)$ denotes the running time of a linear program with mT variables and nT constraints, and $MCF(n,m)$ denotes the running time of a min-cost flow algorithm run on a graph with n nodes and m edges.

Note: A linear programming framework as in [8, 20, 27] when applied to P-Fractional would yield a running time equal to $O(n^2T^2\cdot LP(mT,nT))$. As the running time of an iteration in our approach is dominated by $LP(mT,nT)$, the improvement in running time is at least $O(nT)$-fold, at the expense of providing an ϵ-approximation.

7. FIXED FRACTIONAL ROUTING

Suppose that we want to solve lexicographic maximization of the rates keeping both the routing and the rates constant over time. Observe that, as both the routing and the rates do not change over time, the energy consumption per time slot of each node i is also constant over time and equal to $\Delta b_i = c_{\mathrm{st}}\lambda_i + c_{\mathrm{rt}}\sum_{(j,i)\in E} f_{ji}$.

Proposition 7.1. *Maximum constant energy consumption Δb_i can be determined in time $O(T\log(\frac{b_{i,1}+e_{i,1}}{\delta}))$ for each node $i \in V\backslash\{s\}$, for the total time of $O(nT\log(\frac{b_{i,1}+e_{i,1}}{\delta}))$.*

Similarly as in previous sections, let $F^k_i = 0$ if the rate i is fixed at the beginning of iteration k, and $F^k_i = 1$ if it is not. Initially: $F^1_i = 1$, $\forall i$. Rate maximization can then be implemented as follows:

Algorithm 4 Maximizing-the-Rates(G, F^k, b, e, k)

$\lambda^k_{\max} = \frac{1}{c_{\mathrm{st}}}\min_i\{\Delta b_i - c_{\mathrm{st}}\lambda^{k-1}_i : F^k_i = 1\}$.
repeat for $\lambda^k \in [0, \lambda^k_{\max}]$, via binary search:
  Set the supply of node i to $d_i = \lambda^{k-1} + F^k_i\lambda^k$, capacity of node i to $u_i = \frac{1}{c_{\mathrm{rt}}}(\Delta b_i - c_{\mathrm{st}}\lambda^k)$, for each i.
  Set the demand of the sink to $\sum_i d_i$.
  Solve feasible flow problem on G.
until λ^k takes maximum value for which the flow problem is feasible on G.

The remaining part of the algorithm is to determine which rates should be fixed at the end of iteration k. We note that in each iteration k, the maximization of the rates produces a flow f in the graph G^k with the supply rates λ^k_i. Instead of having capacitated nodes, we can modify the input graph by a standard procedure of splitting each node i into two nodes i' and i'', and assigning the capacity of i to the edge (i', i''). This allows us to obtain a residual graph $G^{r,k}$ for the given flow. We claim the following:

Lemma 7.2. *The rate λ_i of a node $i \in G$ can be further increased in the iteration $k+1$ if and only if there is a directed path from i to the sink node in $G^{r,k}$.*

Lemma 7.3. WATER-FILLING-FRAMEWORK *for* P-FIXED-FRACTIONAL *can be implemented in time*

$$O\left(n \log\left(\frac{b_{i,1} + e_{i,1}}{\delta}\right) \min(T, MF(n,m))\right),$$

where $MF(n,m)$ denotes the running time of a max-flow algorithm for a graph with n nodes and m edges.

8. DETERMINING A ROUTING

In this section we demonstrate that solving P-UNSPLITTABLE-FIND is NP-hard, and design an efficient combinatorial algorithm for a relaxed version of this problem–it determines a time-invariable unsplittable routing that maximizes the minimum rate. For P-TREE-FIND we show that it is NP-hard to obtain an approximation ratio better than $\Omega(\log n)$.

8.1 Unsplittable Routing

A simple extension of the NP-hardness proof for unsplittable routing studied in [18] can be used to show NP-hardness of P-UNSPLITTABLE-FIND for a single time slot, and we omit the proof due to space constraints. This result implies that P-UNSPLITTABLE-FIND remains NP-hard for $T > 1$, regardless of whether the routing structure is allowed to change over time slots or not.

However, determining a time-invariable unsplittable routing that guarantees the maximum value of the minimum sensing rate over all time-invariable unplittable routings is solvable in polynomial time, and we provide a combinatorial algorithm that solves it below.

We first observe that in any time-invariable unsplittable routing, if all the nodes are assigned the same sensing rate λ, then every node i spends a fixed amount of energy Δb_i per time slot equal to the energy spent for sensing and sending own flow and for forwarding the flow coming from the descendant nodes: $\Delta b_i = \lambda\,(c_{\rm st} + c_{\rm rt} D_{i,t})$.

The next property we use follows from the integrality of the max flow problem with integral capacities (see, e.g., [1]). This property was stated as a theorem in [17] for single-source unsplittable flows, and we repeat it here for the equivalent single-sink unsplittable flow problem:

Theorem 8.1. [17] *Let $G = (N, E)$ be a given graph with the predetermined sink node s. If the supplies of all the nodes in the network are from the set $\{0, \lambda\}$, $\lambda > 0$, and the capacities of all the edges/nodes are integral multiples of λ, then: if there is a fractional flow of value f, there is an unsplittable flow of value at least f. Moreover, this unsplittable flow can be found in polynomial time.*

Note: For the setting of theorem 8.1, any augmenting-path or push-relabel based max flow algorithm produces a flow that is unsplittable, as a consequence of the integrality of the solution produced by these algorithms. We will assume that the used max-flow algorithm has this property.

The last property we need is that our problem can be formulated in the setting of theorem 8.1. We observe that for a given sensing rate λ, each node spends $c_{\rm st}\lambda$ units of energy for sensing, whereas the remaining energy can be used for routing the flow originating at other nodes. Therefore, for a given λ, we can set the supply of each node i to λ, set its capacity to $u_i = (\Delta b_i - c_{\rm st}\lambda)/c_{\rm rt}$ (making sure that $\Delta b_i - c_{\rm st}\lambda \geq 0$), and observe the problem as the feasible flow problem. For any feasible *unsplittable* flow solution with all the supplies equal to λ, we have that flow through every edge/node equals the sum flow of all the routing paths that contain that edge/node. As every path carries a flow of value λ, the flow through every edge/node is an integral multiple of λ. Therefore, to verify whether it is feasible to have a sensing rate of λ at each node, it is enough to downround all the nodes' capacities to the nearest multiple of λ: $u_i = \lambda \cdot \lfloor (\Delta b_i - c_{\rm st}\lambda)/(c_{\rm rt}\lambda) \rfloor$, and apply the theorem 8.1.

An easy upper bound for λ is $\lambda_{\max} = \min_i \Delta b_i / c_{\rm st}$, which follows from the battery nonnegativity constraint. The algorithm becomes clear now:

Algorithm 5 MAXMIN-UNSPLITTABLE-ROUTING(G, b, e)

1: Perform a binary search for $\lambda \in [0, \lambda_{\max}]$.
2: For each λ chosen by the binary search set node supplies to λ and node capacities to $u_i = \lambda \cdot \lfloor (\Delta b_i - c_{\rm st}\lambda)/(c_{\rm rt}\lambda) \rfloor$. Solve feasible flow problem.
3: Return the maximum feasible λ.

Lemma 8.2. *The running time of* MAXMIN-UNSPLITTABLE-ROUTING *is $O(\log(\min_i(b_{i,1} + e_{i,1})/(c_{st}\delta)) \cdot MF(n+1, m))$, where $MF(n,m)$ is the running time of a max-flow algorithm on an input graph with n nodes and m edges.*

8.2 Routing Tree

If it was possible to find the (either time variable or time-invariable) max-min fair routing tree in polynomial time for any time horizon T, then the same result would follow for $T = 1$. It follows that if P-TREE-FIND NP-hard for $T = 1$, it is also NP-hard for any $T > 1$. Therefore, we restrict our attention to $T = 1$. In such a setting, determining a tree with the maximum value of the minimum sensing rate is equivalent to the maximum lifetime tree problem from [5]. The instance used in [5] for showing the NP-hardness of the problem has the property that on that instance, at the optimum, P-TREE-FIND produces $\lambda_1 = \lambda_2 = ... = \lambda_n = \lambda$. Therefore, P-TREE-FIND is also NP-hard.

We state the following lower bound result without a proof, and instead provide it in [23].

Theorem 8.3. *The lower bound on the approximation ratio of* P-TREE-FIND *is $\Omega(\log n)$.*

9. CONCLUSIONS AND FUTURE WORK

This paper presents a comprehensive algorithmic study of the max-min fair rate assignment and routing problems in energy harvesting networks with predictable energy profile. We develop algorithms for the WATER-FILLING-FRAMEWORK implementation under various routing types. The running times of the developed algorithms are summarized in Table 2. The algorithms provide important insights into the structure of the problems, and can serve as benchmarks for evaluating distributed, online, and approximate algorithms.

The WATER-FILLING-FRAMEWORK does not specify how to implement the MAXIMIZING-THE-RATES and FIXING-THE-RATES steps. Although general algorithms that implement water-filling (e.g., [8, 20, 27]) can be adapted to solve P-

Table 2: Running times of the algorithms for the Water-filling-Framework implementation.

	Maximizing-the-Rates	Fixing-the-Rates	**Total**
P-Unsplittable-Find	$O\left(nT\log\left(\frac{B+\max_{i,t} e_{i,t}}{\delta c_{\rm st}}\right)\right)$	$O(mT)$	$O\left(n^2T^2\log\left(\frac{B+\max_{i,t} e_{i,t}}{\delta c_{\rm st}}\right)+nmT^2\right)$
P-Fixed-Fractional	$O\left(n\log\left(\frac{b_{i,1}+e_{i,1}}{\delta}\right)\min(T, MF(n,m))\right)$	$O(m)$	$O\left(n\log\left(\frac{b_{i,1}+e_{i,1}}{\delta}\right)\min(T, MF(n,m))\right)$
P-Fractional	$\tilde{O}(T^2\epsilon^{-2}\cdot(nT+MCF(n,m)))$	$LP(mT,nT)$	$\tilde{O}(nT(T^2\epsilon^{-2}\cdot(nT+MCF(n,m))+LP(mT,nT)))$

Unsplittable-Rates, P-Fractional, and P-Fixed-Fractional, their implementation would require solving a large number of linear programs (LPs), each with a high number of variables and constraints. This would result in a very high running time. Moreover, such algorithms do not provide insights into the problem structure.

Our algorithms exploit the problem structure and in most cases do not use linear programming. The only exception is the algorithm for P-Fractional, which solves $O(nT)$ LPs (an adaptation of [8, 20, 27] would need to solve $O(n^2T^2)$ LPs). Furthermore, each LP in our solution searches over much smaller space (only within the ϵ-region of the starting point, for any ϵ-approximation).

Overall, the results reveal interesting trade-offs between different routing types. For example, in simple routing types (routing tree and unsplittable routing), it is relatively simple to determine the max min-fair rate assignment whenever the routing is provided at the input. However, determining a good routing (the one that provides lexicographically maximum assignment of rates among all the routings of the same type) is hard even for a single time slot.

There are several directions for future work. For example, extending the model to incorporate the energy consumption due to the control messages exchange would provide a more realistic setting. Moreover, designing algorithms for unpredictable energy profiles that can be implemented in an online and/or distributed manner is of high practical importance.

10. ACKNOWLEDGEMENTS

We are grateful to Prof. Mihalis Yannakakis for useful discussions. This research was supported in part by NSF grants CCF-1349602, CCF-09-64497, and CNS-10-54856 and the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement n^o[PIIF-GA-2013-629740].11.

11. REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[2] B. Bacinoglu and E. Uysal-Biyikoglu. Finite-horizon online transmission rate and power adaptation on a communication link with markovian energy harvesting. *CoRR*, abs/1305.4558, 2013.

[3] D. Bertsekas and R. Gallager. *Data networks (2nd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[4] P. Blasco, D. Gunduz, and M. Dohler. A learning theoretic approach to energy harvesting communication system optimization. In *Proc. IEEE Globecom Workshops'12*, 2012.

[5] C. Buragohain, D. Agrawal, and S. Suri. Power aware routing for sensor databases. In *Proc. IEEE INFOCOM'05*, 2005.

[6] J.-H. Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(4):609–619, 2004.

[7] A. Charny, D. D. Clark, and R. Jain. Congestion control with explicit rate indication. In *Proc. IEEE ICC'95*, 1995.

[8] S. Chen, Y. Fang, and Y. Xia. Lexicographic maxmin fairness for data collection in wireless sensor networks. *IEEE Trans. Mobile Comput.*, 6(7):762–776, July 2007.

[9] S. Chen, P. Sinha, N. Shroff, and C. Joo. A simple asymptotically optimal energy allocation and routing scheme in rechargeable sensor networks. In *Proc. IEEE INFOCOM'12*, 2012.

[10] M. Gatzianas, L. Georgiadis, and L. Tassiulas. Control of wireless networks with rechargeable batteries. *IEEE Trans. Wireless Commun.*, 9(2):581–593, 2010.

[11] M. Gorlatova, A. Bernstein, and G. Zussman. Performance evaluation of resource allocation policies for energy harvesting devices. In *Proc. WiOpt'11*, 2011.

[12] M. Gorlatova, P. Kinget, I. Kymissis, D. Rubenstein, X. Wang, and G. Zussman. Challenge: ultra-low-power energy-harvesting active networked tags (EnHANTs). In *Proc. ACM MobiCom'09*, 2009.

[13] M. Gorlatova, R. Margolies, J. Sarik, G. Stanje, J. Zhu, B. Vigraham, M. Szczodrak, L. Carloni, P. Kinget, I. Kymissis, and G. Zussman. Energy harvesting active networked tags (EnHANTs): Prototyping and experimentation. Technical Report 2012-07-27, Columbia University, July 2012.

[14] M. Gorlatova, A. Wallwater, and G. Zussman. Networking low-power energy harvesting devices: Measurements and algorithms. *IEEE Trans. Mobile Comput.*, 12(9):1853–1865, 2013.

[15] B. Gurakan, O. Ozel, J. Yang, and S. Ulukus. Energy cooperation in energy harvesting two-way communications. In *Proc. IEEE ICC'13*, 2013.

[16] L. Huang and M. Neely. Utility optimal scheduling in energy-harvesting networks. *IEEE/ACM Trans. Netw.*, 21(4):1117–1130, 2013.

[17] J. Kleinberg. Single-source unsplittable flow. In *Proc. IEEE FOCS'96*, 1996.

[18] J. Kleinberg, Y. Rabani, and E. Tardos. Fairness in routing and load balancing. In *Proc. IEEE FOCS'99*, 1999.

[19] L. Lin, N. Shroff, and R. Srikant. Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources. *IEEE/ACM Trans. Netw.*, 15(5):1021–1034, 2007.

[20] R.-S. Liu, K.-W. Fan, Z. Zheng, and P. Sinha. Perpetual and fair data collection for environmental energy harvesting sensor networks. *IEEE/ACM Trans. Netw.*, 19(4):947–960, Aug. 2011.

[21] R. Madan and S. Lall. Distributed algorithms for maximum lifetime routing in wireless sensor networks. *IEEE Trans. Wireless Commun.*, 5(8):2185–2193, 2006.

[22] Z. Mao, C. Koksal, and N. Shroff. Near optimal power and rate control of multi-hop sensor networks with energy replenishment: Basic limitations with finite energy and data storage. *IEEE Trans. Autom. Control*, 57(4):815–829, 2012.

[23] J. Marašević, C. Stein, and G. Zussman. Max-min fair rate allocation and routing in energy harvesting networks: Algorithmic analysis. *CoRR*, abs/1406.3671, June 2014.

[24] N. Megiddo. Optimal flows in networks with multiple sources and sinks. *Mathematical Programming*, 7(1):97–107, 1974.

[25] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener. Transmission with energy harvesting nodes in fading wireless channels: Optimal policies. *IEEE J. Sel. Areas Commun.*, 29(8):1732–1743, 2011.

[26] S. Plotkin, D. Shmoys, and E. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Math. of O.R.*, 20(2):257–301, 1995.

[27] B. Radunović and J.-Y. L. Boudec. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Trans. Netw.*, 15(5):1073–1083, Oct. 2007.

[28] S. Sarkar and L. Tassiulas. Fair allocation of discrete bandwidth layers in multicast networks. In *Proc. IEEE INFOCOM'00*, 2000.

[29] R. Srivastava and C. Koksal. Basic performance limits and tradeoffs in energy-harvesting sensor nodes with finite data and energy storage. *IEEE/ACM Trans. Netw.*, 21(4):1049–1062, 2013.