A

Distributed Optimal Lexicographic Max-min Rate Allocation in Solar Powered Wireless Sensor Networks

Shusen Yang and Julie McCann, Imperial College London

Understanding the optimal usage of fluctuating renewable energy in Wireless Sensor Networks (WSNs) is complex. Lexicographic Max-min (LM) rate allocation is a good solution, but is non-trivial for multi-hop WSNs, as both fairness and sensing rates have to be optimized through the exploration of all possible forwarding routes in the network. All current optimal approaches to this problem are centralized and off-line, suffering from low scalability and large computational complexity; typically solving $O(N^2)$ linear programming problems for N-node WSNs. This paper presents the *first optimal distributed solution to this problem with much lower complexity*. We apply it to Solar Powered WSNs (SP-WSNs) to achieve both LM optimality and sustainable operation. Based on realistic models of both time-varying solar power and photovoltaic-battery hardware, we propose an optimization framework that integrates a local power management algorithm with a global distributed LM rate allocation scheme. The optimality, convergence, and efficiency of our approaches are formally proven. We also evaluate our algorithms via experiments on both solar-powered MicaZ motes and extensive simulations using real solar energy data and practical power parameter settings. The results verify our theoretical analysis and demonstrate how our approach outperforms both the state-of-the-art centralized optimal and distributed heuristic solutions.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Design, Algorithms, Theory, Performance

Additional Key Words and Phrases: Solar Powered Wireless Sensor Networks, Power Management, Lexicographic optimality, Distributed multi-objective optimization, Max-min fairness

1. INTRODUCTION

The top two challenges facing wireless sensor networks (WSNs) are that of network-wide longevity and maintenance after deployment. The ability to conserve energy is core to both these challenges. Therefore harvesting energy from the environment brings a step change that ensures the viability of WSN for real world deployments [Sudevalayam and Kulkarni 2011; Sharma et al. 2009]. However, the ability of the WSN to meet their requirements while maximizing longevity will be compromised if the system is not able to exploit renewable energy optimally. Various environmental energy sources exist, such as solar, thermal, wind and vibrational. Of these, solar energy has been more widely considered and Solar Powered WSNs (SP-WSNs) have been attracting a growing interest in various research fields including hardware system design (e.g. [Taneja et al. 2008]), node-centric power management schemes (e.g. [Kansal et al. 2007]), as well as network-wide algorithms (e.g. [Liu et al. 2011]).

Renewable solar energy provides an opportunity to achieve so-called Energy Neutral Operation (ENO) [Kansal et al. 2007]; that is, the solar-powered node achieves theoretically perpetual operation (it always has energy). The main power management goals for solar-powered sensor nodes, therefore, are to achieve both ENO while maximizing capacity or

workload, rather than to simply maximize their lifetime (e.g. [Moser et al. 2010; Niyato et al. 2007]). To facilitate this, both the time-varying nature of solar power and the behaviors of photovoltaic-battery hardware system (e.g. battery recharging inefficiencies) should be realistically modeled.

In SP-WSNs, each senor node collects environmental data then forwards sensor data to a sink, or to multiple sinks, in a multi-hop fashion. Besides ENO at each sensor node, SP-WSNs aim at achieving the following two nework-wide objectives: (1) high network throughput for better solar energy resource utilization; (2) fair sensing rate assignment for all nodes across the network. However, there is a fundamental trade-off between network throughput and fairness for given solar energy resources. On one hand, if we only maximize network throughput, the sensors that are farthest from the sinks, or those which have poor solar harvesting opportunities, will be allocated much lower sensing rates than those closer to the sink, resulting in potentially unacceptable bias in the readings coming from network. On the other hand, absolute fairness (i.e. enforce equal sensing rates for all nodes) would lead to significant reduction in network throughput and inefficient solar energy usage (e.g. energy may be lost due to the battery overcharging).

Max-min fairness [Bertsekas and Galager 1992] is a well-recognized approach to balance the tradeoff between network throughput and fairness, and is widely adopted in rate allocation and control schemes in WSNs [Liu et al. 2011; Rangwala et al. 2006; Sridharan and Krishnamachari 2009]. In the context of WSNs, we describe a sensing rate allocation as max-min fair if no sensor can be allocated a higher rate without reducing the rate of another sensor that has equal or lower rate. Classic max-min rate allocation assumes that underlying end-to-end data traffic routes [Bertsekas and Galager 1992; Liu et al. 2011; Rangwala et al. 2006; Sridharan and Krishnamachari 2009] are predetermined, and purely adjusts the sensing rates at the transport layer. Therefore, there exists a huge number of classic max-min rate allocations for a given network, because the number of all possible underlying routes is of exponential order of the number of nodes in the network.

In this paper, we focus on a generalization of max-min rate allocation, the Lexicographic Max-min (LM) rate allocation [Chen et al. 2007; Hou et al. 2008; Liu et al. 2011; Radunovic and Boudec 2007], for SP-WSNs with arbitrary topologies. LM rate allocation jointly optimizes the end-to-end routes and the sensing rate allocation at the network and transport layers respectively, i.e. it computes the optimal max-min rate assignment by exploring all possible routes and energy resources in SP-WSNs. Therefore, LM rate allocation is optimal over all possible classic max-min allocations for a given network.

Theoretically, it is proven that the LM vector is *uniquely* optimal over any given convex and compact set [Radunovic and Boudec 2007]. This means that as long as the network constraints (energy in SP-WSNs) form an unique convex and compact set for all possible rate allocations, the LM rate allocation will be the unique optimal solution. To compute the LM vector, an approach called max-min programming is proposed in [Radunovic and Boudec 2007]. All current optimal solutions [Chen et al. 2007; Hou et al. 2008; Liu et al. 2011] to the LM rate allocation problem in WSNs can be considered as specific implementations of the max-min programming approach. However, all these solutions are centralized and suffer from a large computational complexity; of the order of solving N^2 Linear Programming (LP) problems for a N-node WSN. Therefore, although they are solvable in polynomial-time, their centralized nature and complexity still prohibits their use in practical applications. Furthermore, such centralized approaches may lead to loops in the computed routes associated with the LM rate allocation, leading to large end-to-end delays and unnecessary network resource costs.

It is recognized that distributed approaches are more suitable for WSNs as they allow the system to both scale and be resilient to change and failure. Recently, two distributed algorithms, DLEX and DLEX-DAG [Liu et al. 2011], were proposed to compute the max-min fair rate allocation, based on predetermined routing tree (single-path routing) and directed

acyclic graphs (DAG, i.e. multi-path routing) respectively. Furthermore, both of them are designed only for WSNs with a single sink, which restricts their application in large-scale WSNs where multiple sinks exist (e.g. [Shah-Mansouri et al. 2009]). In contrast, LM rate allocation considers all possible routes over arbitrary network topologies with either a single sink or multiple sinks. To the best of our knowledge, there exists no distributed solution to LM rate allocation problem yet.

In this paper, we focus on fully distributed solution to the LM rate allocation. In doing so, we demonstrate that designing a distributed solution for LM rate allocation is much more difficult than the traditional distributed Network Utility Maximization (NUM)-based rate control scheme; well-studied over the past decade [Chiang et al. 2008]. The core reason for this is that LM rate allocation is inherently a multi-objective optimization problem (e.g. [Huang 2007; Salles and Barria 2008]), while NUM-based rate control algorithms normally solve single-objective convex optimization problems only (i.e. they typically maximize aggregated concave utility functions such as α-fairness [Mo and Walrand 2000; Lan et al. 2010]).

The main contributions of this paper are summarized as follows:

(1) We present a systematic approach to LM rate allocation in SP-WSNs with arbitrary topologies and multiple sinks (or a single sink). This is formalized as a joint power management, routing, and rate allocation problem. The formalization considers both the time-varying nature of the solar power and realistic photovoltaic-battery hardware behaviors; such as battery capacity, recharging inefficiencies, and energy leakage. We decompose the formalized problem into two sub-problems: a LP problem for Local Power Management (LPM) that optimizes power for each node, and a multi-objective optimization problem for the network-wide LM rate allocation.

(2) An efficient Local Power Management (LPM) algorithm is proposed to compute the maximum feasible energy consumption budget for each sensor node to ensure ENO. Compared with solving LP at runtime, our LPM algorithm remains optimal with a much lower complexity (similar to a sorting operation), which is suitable for sensor nodes with limited computation resources.

(3) We develop *the first distributed optimal* approach to the global LM rate allocation problem. It operates by iterating through two distributed algorithms: a dual-decomposition-based algorithm, namely the Distributed Maximum Common Rate (DMCR) and the LM rate Determination (LMD); a graph-theoretic scheme. Our DMCR-LMD approach is not only fully distributed, but also achieves a worst-case complexity of $O(N)$ LPs to compute LM rate allocation for a N-node SP-WSN, which is much more efficient than current centralized approaches requiring $O(N^2)$ LPs.

(4) We present theoretical proofs of the optimality, convergence, and efficiency for both the individual algorithms and the whole system. Furthermore, we also demonstrate several nice properties of our approaches such as the loop-free optimal routing with respect to the LM rate allocation.

(5) The LPM algorithm is evaluated on a solar-powered MicaZ mote. Using a realistic power model and parameter settings, we also constructed simulations to evaluate the performance of our DMCR-LMD approach, in terms of optimality, overheads, convergence, and scalability. Simulation results verify our theoretical analysis and demonstrate that our approach manages to achieve much better fairness than the state-of-the-art distributed algorithms DLEX and DLEX-DAG, and with much lower complexity compared with the centralized approaches. In addition, we also study important practical issues such as how to overcome errors in solar power prediction to ensure ENO for realistic scenarios, through LPM parameter adjustments.

The remainder of this paper is organized as follows: the next section presents system model. We present the LPM algorithm in Section 3. Sections 4 and 5 respectively discuss the DMCR and LMD in detail. Section 6 provides rigorous theoretical analysis of the proposed

algorithms. In Section 7, we discuss practical issues of implementing DMCR-LMD approach in real SP-WSNs. The evaluation of our approach is presented in Section 7. Section 8 discusses the related work and we finally we conclude the paper in Section 9.

2. SYSTEM MODEL

We consider a multi-hop SP-WSN that consists of several sensor nodes, and one or multiple sinks. The SP-WSN can be represented as a directed graph $\mathcal{G}(\mathcal{V} \cup \mathcal{S}, \mathcal{L})$ where $\mathcal{V}$ is the set of all sensor nodes, $\mathcal{S}$ is the set of all sinks, and $\mathcal{L}$ is the set of all logical links. For each node $x \in \mathcal{V} \cup \mathcal{S}$, define N_x as the set of all one-hop neighbors of x excluding x.

Sensor node x collects environmental data (e.g. temperature and humidity) at a sensing rate $r_x \geq 0$. A data packet is sent in a multi-hop manner to any sink in $\mathcal{S}$. Let $f_{x,y} \geq 0$ be the transmission rate at which node x transmit sensor data to node y, $(x, y) \in \mathcal{L}$.

2.1. Energy Model

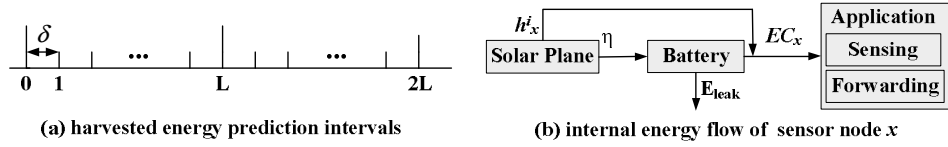(a) harvested energy prediction intervals   (b) internal energy flow of sensor node x

Fig. 1. Prediction interval and energy flow.

To model the time-varying solar power, time is divided into identical discrete slots with duration δ (e.g. [Challen et al. 2010; Moser et al. 2010]). Due to the predicability of solar power, a prediction algorithm (e.g. [Bergonzini et al. 2010]) is assumed to be used to estimate the harvesting profile of every non-overlapping prediction interval consisting of L slots, as shown in Figure 1 (a). At the beginning of every prediction interval, harvested energy h_x^i in every future slot $i \in I$ is estimated, where $I=\{1, 2, ..., L\}$. Based on the predicted harvesting profile of every prediction interval, our distributed algorithms calculate the optimal LM rate allocation and corresponding routes for this prediction interval.

Let the long-term average energy cost (Joule per bit) of sensing, receiving and transmitting be E_s, E_r, and E_t respectively. Then, the total energy consumption EC_x, $x \in \mathcal{V}$ of every slot in a given prediction interval is set to be same and is represented as:

$$EC_x = E_s r_x + E_r \sum_{y \in N_x} f_{y,x} + E_t \sum_{y \in N_x} f_{x,y} \tag{1}$$

$$E_{min} \leq EC_x \leq E_{max} \tag{2}$$

Where E_{min} and E_{max} are the lower and upper bounds of energy consumption. For instance, we measured a MicaZ mote [Mic] with the fixed power level 0DB. When turn the CC2420 transceiver off and keep Micro-Controller Unit (MCU) idle, we get $E_{min} \approx 13.7\delta \times 10^{-3}$ J; when turn the CC2420 transceiver on and keep Micro-controller unit (MCU) active, we get $E_{max} \approx (78.4 + P_{max}^{sensor})\delta \times 10^{-3}$ J, where P_{max}^{sensor} is the maximum power consumption of sensor depending on the specific sensor hardware. In addition, E_{min} can also be defined by application requirements.

Figure 1.b shows a sensor node's internal energy flow model, which considers a realistic rechargeable battery model with battery capacity B_{max}, recharging efficiency $\eta < 1$ and a constant leakage E_{leak} for each slot. Similar power system models are also used in realistic power management algorithms [Kansal et al. 2007; Moser et al. 2010]. h_x^i can be either stored in the battery or be directly consumed. Let B_x^i be the residual energy of sensor

node x's battery at the beginning of every slot i. If EC_x is larger than h_x^i in slot i, which means that $EC_x - h_x^i$ amount of energy will be discharged from the battery; Otherwise, $\eta(h_x^i - EC_x)$ amount of energy will be recharged into the battery, according to the inefficient recharging process. In summary, the following constraints represent the proposed battery model:

$$B_x^{i+1} = B_x^i + \eta|h_x^i - EC_x|_+ - |EC_x - h_x^i|_+ - E_{\text{leak}} \qquad (3)$$

$$0 \leq B_x^{i+1} \leq B_{\text{max}} \qquad (4)$$

Where $|x|_+ = x$, if $x > 0$, and $|x|_+ = 0$, otherwise.

Finally, we also consider the so-called final state constraint [Moser et al. 2010]

$$B_x^{\text{L}+1} \geq \varphi \qquad (5)$$

The parameter $\varphi < B_{\text{max}}$ ensures that there is enough initial energy for next prediction interval, which influences the long-term performance of the system and feasible solution of LPM problem (see Proposition 3.2 in subsection 3.1). Given the existence of solar prediction errors in practice, a smaller φ results in a more aggressive power system behavior and a high risk to deplete battery. On the other hand, a larger φ may lead to battery overcharging (i.e. lose solar harvesting opportunity) and poor performance. In this paper, we treat φ as a constant protocol parameter witch can be set according to application scenarios. Dynamically adjusting φ over different prediction intervals such as [Moser et al. 2010] is outside of the scope of this paper.

2.2. Wireless Data Forwarding Model

Due to the law of flow conservation, we have for all sensor node $x \in \mathcal{V}$

$$r_x + \sum_{y \in N_x} f_{y,x} - \sum_{y \in N_x} f_{x,y} = 0 \qquad (6)$$

Constraint (6) means that the sum of forwarding rates of i's all outgoing links must be equal to that of its all incoming links plus its sensing rate.

We define the packet reception ratio (PRR) over a wireless link (x, y), $PRR_{x,y}$, as the probability of successfully transmitting a data packet from node x to y. We define

$$c_{x,y} = c^{\text{max}} PRR_{x,y}$$

as the logical link-layer capacity of a wireless link (x, y), where c^{max} is the data rate of the wireless radio (e.g. 250 kbps for IEEE 802.15.4 radio). Consider the link capacity constant, we have transmitting a data packet from node x to y. We define

$$0 \leq f_{x,y} \leq c_{x,y} \qquad (7)$$

Here $c_{x,y}$ represents the average link capacity over a prediction interval. Due to the dynamic nature of wireless channel quality, instantaneous $PRR_{x,y}$ and $c_{x,y}$ changes over time, but can be accurately estimated at runtime in practical WSNs [Gnawali et al. 2009; Moeller et al. 2010]. However, since SP-WSN is statically deployed, the Euclidean distance between two any pair of nodes is fixed. As a result, the average Signal-to-Noise Ratio (SNR) at the receiver of each wireless link is fixed. Therefore, it is clear that the average link capacity $c_{x,y}$, $(x, y) \in \mathcal{L}$ for different prediction intervals are nearly the same. Therefore, the average link capacity $c_{x,y}$ of a prediction interval can be predicted by using schemes such as EWMA [Cox 1961] with a high degree of accuracy.

Due to the limited size of solar panel, energy is the dominant bottleneck resource of SP-WSNs rather than wireless capacity [Liu et al. 2011; Hou et al. 2008; Sudevalayam and

Kulkarni 2011]. To focus on energy aspect, we use above simple channel model. However, the essence of our solution to the LM rate allocation problem is nonetheless still preserved, and our approach can be extended to more complex wireless models.

2.3. Problem Formalization

We first define a feasible sensing rate allocation as follows:

Definition 2.1 (*Feasibility Condition*). A rate allocation is represented as a $|\mathcal{V}|$-dimension vector $R = (r_1, r_2, \cdots, r_{|\mathcal{V}|})$. R is feasible if under R there exists a power management and a routing scheme such that all the constraints (1)–(7) can be guaranteed.

The objective of this paper, the optimal LM rate allocation, is defined as follows:

Definition 2.2 (*Lexicographic Optimality*). Let $R = (r_1, r_2, \cdots, r_{|\mathcal{V}|})$ be a feasible rate allocation which is sorted in non-descending order. Any two such vectors R and $R^{'}$ have the following relationships: 1) If $r_i = r_i^{'}$ for any $i = 1, 2, \cdots |\mathcal{V}|$, then R is lexicographically equal to $R^{'}$; 2) If there exist a prefix $(r_1, r_2, \cdots, r_i)$ of R and a prefix $(r_1^{'}, r_2^{'}, \cdots, r_i^{'})$ of $R^{'}$ such that $r_i > r_i^{'}$, and $r_j = r_j^{'}$, for $1 \leq j \leq i - 1$, Then R is lexicographically greater than $R^{'}$. R is lexicographically optimal if it is lexicographically greater than all other feasible rate allocations.

The lexicographically ordering takes into account the objectives of providing both high throughput and fairness. Let LM^* be the lexicographically optimal rate allocation, then the LM rate allocation problem is:

Objective $\qquad\qquad\qquad\qquad LM^*$ $\qquad\qquad\qquad\qquad$ (8)

Subject to $\qquad\qquad$ Constraints $(1) - (7)$

It worth noting that although all constraints (1)–(7) is linear, *the LM rate allocation problem (8) is inherently a multi-objective optimization problem rather than a LP problem such as the classic Maximum-Flow Problem.*

Problem (8) can be naturally decoupled into two sub-problems by introducing an auxiliary variable EC_x^{max}, which is the maximum feasible EC_x, $x \in \mathcal{V}$:

$$EC_x^{max} = \max_{<\text{Constraints } (1)-(5)>} \{EC_x\} \qquad\qquad (9)$$

The first sub-problem (9) is the LPM problem which can be solved locally on each node, because all constraints (1)-(5) only require parameters of each sensor node's internal power system. Based on the energy budget EC_x^{max}, the second sub-problem is

Objective $\qquad\qquad\qquad\qquad LM^*$ $\qquad\qquad\qquad\qquad$ (10)

Subject to $\qquad\qquad$ Constraints (6) and (7)

$$\text{E}_\text{s} r_x + \text{E}_\text{r} \sum_{y \in N_x} f_{y,x} + \text{E}_\text{t} \sum_{y \in N_x} f_{x,y} \leq EC_x^{max} \qquad\qquad (11)$$

The second sub-problem has two sets of variables r_x and $f_{x,y}$, $x \in \mathcal{V}$, $y \in N_x$. Consequently, the optimal solution of this problem produces the optimal rate vector LM^* and the optimal $f_{x,y}$ over every link, i.e the optimal routes corresponding to LM^*. Theorem 2.3 shows the uniqueness of LM^*.

THEOREM 2.3. *Optimal LM rate allocation LM^* in our system model is unique.*
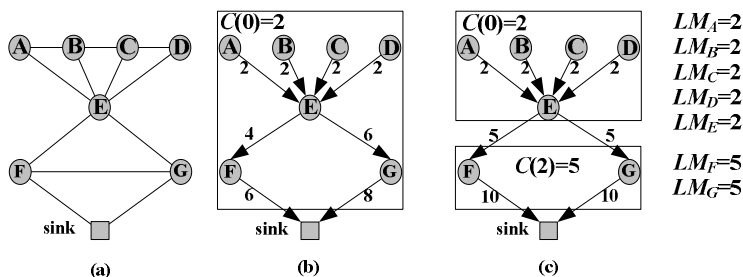
Fig. 2. LM^* for a SP-WSN with seven sensor nodes and a sink. Assume $E_s = E_r = E_t = 1$ unit, link capacity is 50 unit, and the maximum feasible energy consumption of all sensor nodes are 20 units. (a) The network topology. (b) the first maximum common rate and corresponding routes (date forwarding directions and rates). (c) LM^* and corresponding optimal routes.

PROOF. Firstly, the solution to LPM problem (9), EC_x^{max}, is unique according to Proposition 3.3 in Section 3. Therefore, constraints (6), (7), and (11) define the unique feasible region of a rate allocation $R = (r_1, r_2, \cdots, r_{|\mathcal{V}|})$, say $\mathcal{X}$, which is a convex and compact set (i.e. a polyhedron), since EC_x^{max} is unique. Define a function $\phi(R) = R$ over $\mathcal{X}$. Obviously $\phi(R)$ is a continuous and increasing function in R. According to Theorem 1 and Proposition 3 in [Radunovic and Boudec 2007], there exists a unique optimal maximin vector $(\phi(r_1), \phi(r_2), \cdots, \phi(r_{|\mathcal{V}|}))$. Therefore, the optimal LM rate allocation LM^* in our system model is unique. □

2.4. Overview of Distributed Solution

Define LM_x as the LM rate of a sensor node $x \in \mathcal{V}$ (i.e. LM_x is unique, and is an entry of vector LM^*). Let $\mathcal{V}(r) = \{x | LM_x \leq r, x \in \mathcal{V}\}$ be the set of sensor nodes whose LM rates are not larger than a given real number r. For instance, in Figure 2.4[1], $\mathcal{V}(0) = \varnothing$, $\mathcal{V}(2) = \{A, B, C, D, E\}$, and $\mathcal{V}(5) = \{A, B, C, D, E, F, G\}$.

Let the sensors in $\mathcal{V}(r)$ take their LM rates, and all sensors in $\mathcal{V} - \mathcal{V}(r)$ take a common rate, Define $C(r)$ as the maximum feasible common rate of all the sensor nodes in $\mathcal{V} - \mathcal{V}(r)$:

$$C(r) = \max_{<\text{constraints (6),(7), } and \text{ (11)}, r_z = LM_z, \forall z \in \mathcal{V}(r), r_x = r_y, \forall x, y \in \mathcal{V} - \mathcal{V}(r))>} \{r_x\}$$

For instance, in Figure 2.4, $C(0) = 2$ restricted by bottleneck node E, and $C(2) = 5$, restricted by bottleneck nodes F and G. Note that both $\mathcal{V}(r)$ and $C(r)$ are functions of r and will be commonly used in our latter discussions.

Current centralized solution [Chen et al. 2007; Hou et al. 2008; Liu et al. 2011] to LM rate allocation can be considered as specific implementations of the max-min programming approach[Radunovic and Boudec 2007]. In our context, max-min programming can compute LM^* by iteratively solving two kinds of LP problems: Maximum Common Rate (MCR) and Maximum Single Rate (MSR). In each MCR-MSR cycle, MCR computes $C(r)$ for all sensor nodes in $\mathcal{V} - \mathcal{V}(r)$ and then MSR checks whether $LM_x = C(r)$ for each node $x \in \mathcal{V} - \mathcal{V}(r)$. The iteration rule of r for each MCR-MSR cycle is

$$r^n = C(r^{n-1}) \quad r^0 = 0,$$

Hence, LM^* in non-descending order has the following structure:

$$(C(r^0), ..., C(r^0), C(r^1), ..., C(r^1), ..., C(r^n), ..., C(r^n))$$

─────────
[1]Note that although Figure 2.4(a) is illustrated as a bidirectional graph for brevity, all the links in it is directional.

In Figure 2.4, for instance, LM^* in non-descending order is $(2, 2, 2, 2, 2, 5, 5)$ and it needs two MCR-MSR cycles to compute LM^* and corresponding optimal routes. It worth noting that although LM^* is unique, the routes corresponding to LM^* may not be unique. For instance, $f_{A,B} = f_{B,A} = 0$ in Figure 2.4(c). However, we can also get a feasible optimal routing corresponding to LM^*, by resetting $f_{A,B} = f_{B,A} = 1$ and keep all other data forwarding rate constant. However, this route contains a loop.

It can be seen that MCR-MSR approach are not only centralized but also suffer a very large overhead, i.e. solving $O(|\mathcal{V}|)$ MCR and $O(|\mathcal{V}|^2)$ MSR problems. In addition, the routes corresponding to the LM rate allocation computed by above centralized approach may exist loops, which leads to large end-to-end delay and unnecessary network resource costs.
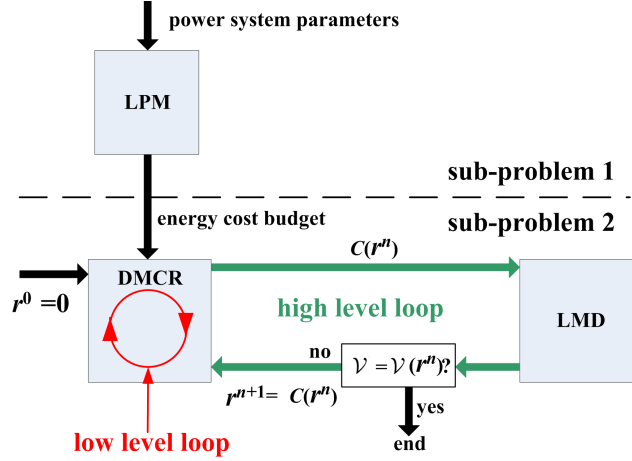

Fig. 3. Logical flow of our distributed DMCR-LMD approach.

Due to its optimality, our approach, DMCR-LMD iterations, adopts the similar logical flow of max-min programming, but computes LM^* in a fully distributed way and with much lower complexity. Figure 3 shows the logical flow of our distributed approach. After solving LPM locally, nodes run two level of iterations to compute LM^*. The high level is the global DMCR-LMD cycles, and the low level is the iterations of each DMCR.

At each DMCR-LMD cycles, the dual-decomposition based DMCR method calculates the maximum common rate $C(r)$. Then the graph-theoretic scheme LMD determines the LM rates of all nodes in $\mathcal{V}-\mathcal{V}(r)$ simultaneously rather than solving $|\mathcal{V}-\mathcal{V}(r)|$ MSR problems one by one. Our LMD scheme requires nearly no calculations and at most 1 control packet per node for the *whole* procedure of LM^* calculation. Further more, our distributed approach can guarantee loop-free routes (Lemma 6.2 in Section 6).

3. LOCAL POWER MANAGEMENT

This section focuses on the the first sub-problem (9), the local power management problem, which aims to maximize the average energy consumption EC_x for every $x \in \mathcal{V}$ in a given prediction interval.

3.1. Properties of the LPM problem

Due to our practical power model (e.g. finite battery capacity B_{max}, recharging inefficiency, and all possible solar power $h_x^i \geq 0$, $i \in I$), it is difficult to solve LPM problem (9) directly. In addition, not arbitrary given parameters (e.g. B_x^1 and E_{leak}) would result in a feasible solution of the LPM problem, i.e. $EC_x^{max} \in (E_{min}, E_{max})$. For instance, if the given initial

battery level and estimated solar power are too small, the final batter level B_x^{L+1} would not be able to reach a large φ even when $EC_x = \mathrm{E_{min}}$. Consequently, before presenting the algorithm, we first introduce some very important properties of LPM problem and discuss feasible parameter settings in practice.

PROPOSITION 3.1 (MONOTONICITY). $B_x^{i+1}(EC_x)$ is a monotonic non-increasing function of EC_x for any given slot $i \in I$.

PROOF. See the appendix. □

Two sufficient conditions to grantee that LPM always has a feasible solution are shown in proposition 3.2.

PROPOSITION 3.2 (FEASIBILITY). Two sufficient conditions to grantee that LPM has a feasible solution are

$$\forall i \in I, h_x^i - \mathrm{E_{min}} \geq \mathrm{E_{leak}}/\eta \tag{12}$$

$$B_x^1 + \eta \sum_{i \in I}(h_x^i - \mathrm{E_{min}}) - \mathrm{E_{leak}}L \geq \varphi \tag{13}$$

PROOF. See the Appendix. □

In practice, condition (12) and (13) nearly always holds for daytime of sunny days with carefully selected final state parameter φ. We test this based on a solar powered sensor node with a $9 \times 3.8\mathrm{cm}^2$ solar panel. For night or bad-weather days, neither constraints (12) nor (13) would be guaranteed. However, (12) and (13) are sufficient conditions for feasibility of LPM problem but not necessary conditions to ensure ENO (i.e. battery is not depleted for every slot). We can set φ carefully to store energy during daytime to avoid battery exhausting during night.

Finally, the optimal solution EC_x^{max} has the following property.

PROPOSITION 3.3 (OPTIMALITY AND UNIQUENESS). If there exist two energy consumption levels EC_x^1 and $EC_x^2 \in [E_{min}, E_{max}]$. EC_x^1 is the maximal EC_x that satisfies $B_x^{i+1}(EC_x) \geq 0$, $\forall i \in I$, and EC_x^2 is the maximal EC_x that satisfies the $B_i^{L+1}(EC_x) = \varphi$. The optimal solution of sub-problem (9) is $EC_x^{max} = \max(\mathrm{E_{min}}, \min(EC_x^1, EC_x^2, \mathrm{E_{max}}))$.

PROOF. From proposition 3.1, $EC_x^{max} = \min(EC_x^1, EC_x^2)$ can satisfy both the constraints (4) and (5). Consider $EC_x \in [\mathrm{E_{min}}, \mathrm{E_{max}}]$, Proposition 3.3 obviously holds. □

3.2. LPM Algorithm

Based on the properties 3.1 and 3.3, we design the LMP algorithm. The pseudo-code is shown in Algorithm 1.

The main idea of the LPM algorithm is to gradually decease EC_x from a theoretical upper bound (line 1) until both the two constraints $\forall i \in I, B_x^{i+1} \geq 0$ and $B_x^{L+1} \geq \varphi$ are just guaranteed. The slot pointer i represents the current slot and j records slot 1 or the last slot in which the battery level reaches $\mathrm{B_{max}}$. L_1 and L_2 represent the number of slots between j and i, in which EC_x is smaller than and larger than the harvested solar energy respectively. For instance, Figure 4.a shows the value of a EC_x and solar energy from slot j to a slot k ($B_x^{k+1} < 0$) and Figure 4.b shows the corresponding battery level in all slots before $k + 1$. In Figure 4, if $i = k$, then $L_1 = 6, L_2 = 8$. E represents

$$E = B_x^j + \sum_{EC_x \geq h_x^t} h_x^t + \sum_{EC_x < h_x^t} \eta h_x^t, t \in [j, i]$$

ALGORITHM 1: The operation of a node $x \in \mathcal{V}$ at the beginning of each prediction interval (LPM algorithm).

Input: Initial battery level B_x^1, battery capacity $\mathrm{B_{max}}$, final state constraint parameter φ, recharging efficiency η, leakage $\mathrm{E_{leak}}$, and harvesting profile h_x^i, $\forall i \in I$.

Output: The maximum feasible energy consumption EC_x^{max}

1:   $EC_x \leftarrow \min(\mathrm{E_{max}}, \sum_{i \in I}(h_x^i + B_x^1 - \varphi)/\mathrm{L} - \mathrm{E_{leak}})$;
2:   $i \leftarrow 1; j \leftarrow 1; B_x^j \leftarrow B_x^1; L_1 \leftarrow 0; L_2 \leftarrow 0; E \leftarrow B_x^1$;
3:   **while** $i \leq \mathrm{L}$ **do**
4:     **if** $h_x^i > EC_x$ **then**   /* recharging in slot i */
5:        $B_x^{i+1} = B_x^i + \eta(h_x^i - EC_x) - \mathrm{E_{leak}}$;
6:        $L_1 \leftarrow L_1 + 1$;
7:        $E \leftarrow E + \eta h_x^i - \mathrm{E_{leak}}$;
8:     **else**   /* discharging in slot i */
9:        $B_x^{i+1} = B_x^i + h_x^i - EC_x - \mathrm{E_{leak}}$;
10:       $L_2 \leftarrow L_2 + 1$;
11:       $E \leftarrow E + h_x^i - \mathrm{E_{leak}}$;
12:    **end if**
13:    **if** $B_x^{i+1} > \mathrm{B_{max}}$ **then** /* battery is overcharged*/
14:       $B_x^{i+1} \leftarrow \mathrm{B_{max}}$;
15:       $E \leftarrow \mathrm{B_{max}}$;
16:       $j \leftarrow i + 1$;
17:       $B_x^j \leftarrow \mathrm{B_{max}}$;
18:       $L_1 \leftarrow 0; L_2 \leftarrow 0$;
19:       $i \leftarrow i + 1$;
20:    **else if** $B_x^{i+1} < 0$ **then**   /* decrease EC_x*/
21:       $EC_x \leftarrow E/(\eta L_1 + L_2)$;
22:       $i \leftarrow j$;
23:       $B_x^i \leftarrow B_x^j; E \leftarrow B_x^j; L_1 \leftarrow 0; L_2 \leftarrow 0$;
24:    **else if** $B_x^{i+1} < \varphi \wedge i = L$ **then** /* decrease EC_x*/
25:       $EC_x \leftarrow (E - \varphi)/(\eta L_1 + L_2)$;
26:       $i \leftarrow j$;
27:       $B_x^i \leftarrow B_x^j; E \leftarrow B_x^j; L_1 \leftarrow 0; L_2 \leftarrow 0$;
28:    **else**
29:       $i \leftarrow i + 1$;
30:    **end if**
31: **end while**
32: **return** $\max(EC_x, \mathrm{E_{min}})$

With a EC_x, there may be a slot when the battery level reaches $\mathrm{B_{max}}$(line 13). j always records such latest slot. L_1, L_2, and E are updated to record the corresponding values after slot j. EC_x is only updated by lines 21 and 25.

LEMMA 3.4. *Let EC_x' be the updated EC_x, thus EC_x' must be smaller than EC_x.*

PROOF. For line 21, since $B_x^{k+1}(EC_x) < 0$, we have

$$EC_x > (B_x^j + \sum_{EC_x > h_x^t} h_x^t + \sum_{EC_x < h_x^t} \eta h_x^t - (L_1 + L_2)\mathrm{E_{leak}})/(\eta L_1 + L_2)$$

$$= E/(\eta L_1 + L_2) = EC_x'$$

For line 25, similarly, we can obtain $B_x^{\mathrm{L}+1} < \varphi \Leftrightarrow EC_x' < EC_x$. $\square$

Since EC_x is feasible for all slots before $k+1$, EC_x' is therefore feasible for all slots before $k+1$, according to Proposition 3.1 and Lemma 3.4. Consequently, having calculated EC_x', the LPM algorithm only rechecks the feasibility of EC_x' in slot $k+1$. To this end, the LPM algorithm recalculates the battery levels from slot j to slot $k+1$ (lines 22 and 26), because B_x^j remains $\mathrm{B_{max}}$ or B_x^1 for any $EC_x' < EC_x$, according to Proposition 3.1.

For a calculated EC_x', let the corresponding numbers of slots between j and k be L_1' and L_2' respectively, in which harvesting energy is smaller and larger than EC_x'. For instance, in Figure 4, $L_1' = L_1 + 4$ and $L_2' = L_2 - 4$. When the energy consumption level drops from EC_x to EC_x', the following three cases would happen.
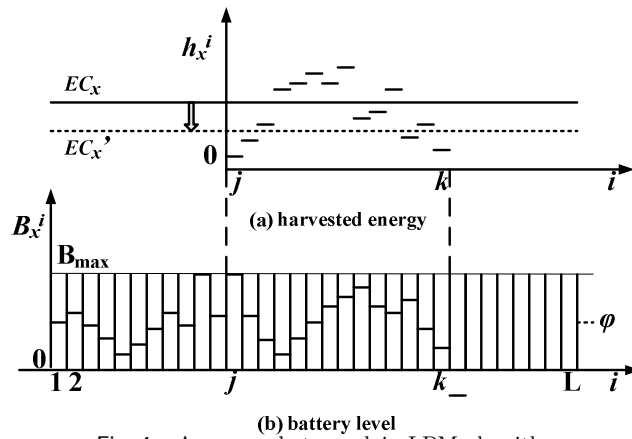

(a) harvested energy

(b) battery level
Fig. 4. An example to explain LPM algorithm.

LEMMA 3.5. *For any given* $1 < k \le \mathrm{L}$ *such that* $B_x^{k+1}(EC_x) < 0$ *if* $k < L$*; or* $B_x^{k+1}(EC_x) < \varphi$*, otherwise the last updated* EC_x *is the maximum feasible energy consumption for all slots before* k*.*

PROOF. See the Appendix. $\square$

THEOREM 3.6. *The output of the LPM algorithm is the optimal solution of sub-problem (9).*

PROOF. According to Lemma 3.5, the last updated EC_x is always the maximum feasible energy consumption for all slots before $k+1$, and the algorithm ends when $k = L$ and $B_x^{k+1}(EC_x) \ge \varphi$. $\square$

Now we discuss the computation overhead of the LPM algorithm. For a given j and k, both the cases 2 and 3 happen at most $k - j + 1$ times before k is updated (because there are $k - j + 1$ slots between slots k and j). Consequently, in above two worst cases, our LPM needs at most $\mathrm{O(L^2)}$ simple arithmetic calculations to compute EC_x^{max}. In practice, we can choose large $\mathrm{B_{max}}$ and δ such that cases 2 and 3 rarely happen, then the number of simple arithmetic calculations reduces to $\mathrm{O(L)}$. In summary, the overhead of our LPM is similar to a sorting operation for L elements, which is suitable for sensor nodes with limited computing capacity.

4. DISTRIBUTED MCR

Let the maximum common rate computed in the last DMCR-LMD cycle be $r \geq 0$, then the current DMCR aims to compute $C(r)$ for all nodes in $\mathcal{V} - \mathcal{V}(r)$ (please see Figure 3 for the logical flow of our DMCR-LMD approach). For instance, if the current DMCR is in the first DMCR-LMD cycle, then $r = 0$. $C(r)$ can be calculated by solving the following problem: $\forall x, y \in \mathcal{V} - \mathcal{V}(r), \; z \in N_x, \; p \in \mathcal{V}(r)$

Maximize
$$\sum_{x \in \mathcal{V} - \mathcal{V}(r)} r_x \tag{14}$$

Subject to
$$r_x - r_y = 0 \tag{15}$$
$$r_x - r \geq 0 \tag{16}$$
$$r_p - LM_p = 0 \tag{17}$$
$$0 \leq f_{x,y} \leq c_{x,y} \tag{18}$$
$$r_x + \sum_{z \in N_x} f_{z,x} - \sum_{z \in N_x} f_{x,z} = 0 \tag{19}$$
$$EC_x^{max} - \mathrm{E_s} r_x - \mathrm{E_r} \sum_{z \in N_x} f_{z,x} - \mathrm{E_t} \sum_{z \in N_x} f_{x,z} \geq 0 \tag{20}$$

Constraint (15) enforces all r_x to be equal, according to the objective of the MCR problem. Constraints (19) and (20) refer to flow conservation law and energy constraints respectively. Also, since $x \in \mathcal{V} - \mathcal{V}(r)$, constraint (16) ensures that the lower bound of r_x is r. Further, constraint (17) highlights that every node $p \in \mathcal{V}(r)$ should keep their sensing rate as LM_p which has been determined by previous DMCR-LMD cycles.

The objective of problem (14) is to calculate the maximum common rate $C(r)$ of sensor nodes in $\mathcal{V} - \mathcal{V}(r)$ (i.e. $r_x, x \in \mathcal{V} - \mathcal{V}(r)$), as well as the corresponding optimal routes ($f_{x,z}, z \in N_x$). DMCR is based on dual-decomposition which is commonly used in distributed network optimization. In contrast to existing approaches, however, DMCR deals with two novel problems as follows:

Heterogeneous Decomposition. From the problem formalization (14)-(20) we can see that not only all sensor nodes in $\mathcal{V}$ but also every node $x \in \mathcal{V} - \mathcal{V}(r)$ and every node $z \in N_x$ are part of the current DMCR calculation. A simple example is shown in Figure 5, nodes 2, 6, and 7 who are in $\mathcal{V}(r)$, will not involve in current DMCR, because their LM rates and the corresponding optimal routes (represented as directional solid lines) have been determined in previous DMCR-LMD cycles. Since the current DMCR calculation is to compute the maximum common rate $C(r)$ of the nodes in $\mathcal{V} - \mathcal{V}(r)$ (i.e. nodes 1, 3, 4, 8, 10, 11, and 12), as well as the routes (flows over links that represented as dotted lines), not only the nodes in $\mathcal{V} - \mathcal{V}(r)$, but also the nodes 5 and 9 who are the neighbors of nodes 4, 8, 10, 11 are part of the current DMCR calculation. Nodes 5 and 9 only calculate the optimal flows over links (5, 4), (5, 8), (9, 10), and (9, 11). Our DMCR manages to decouple the original problem (14) into different subproblems for each node $x \in \mathcal{V} - \mathcal{V}(r)$ and each node $z \in N_x$.

Disconnected Network Topology. Current dual-decomposition-based schemes always deal with the connected network. In our case, however, with absence of nodes in $\mathcal{V}(r)$, the sub-graph consists of only nodes in $\mathcal{V} - \mathcal{V}(r)$ would not be connected. As shown in Figure 5, for instance, nodes in $\mathcal{V} - \mathcal{V}(r)$ are separated into two sub-sets, sharing no link between each other. Consequently, the rate equality constraint (15) can not be aware by all nodes in $\mathcal{V} - \mathcal{V}(r)$ in a distributed fashion. In this case, since there is no centralized controller, DMCR parallel calculates two maximum common rates respectively for the two sub-sets, which is different from original centralized LP approach. In fact, this is a good property of our DMCR, because multiple maximum common rates can be parallel found in one DMCR

calculation, which may result in less DMCR-LMD cycles than that of centralized MCR-MSR. Our simulation results show that the number of DMCR-LDM cycles can be 150% less than that of MCR-MSR cycles, which will significantly reduce the system convergence time and overheads.
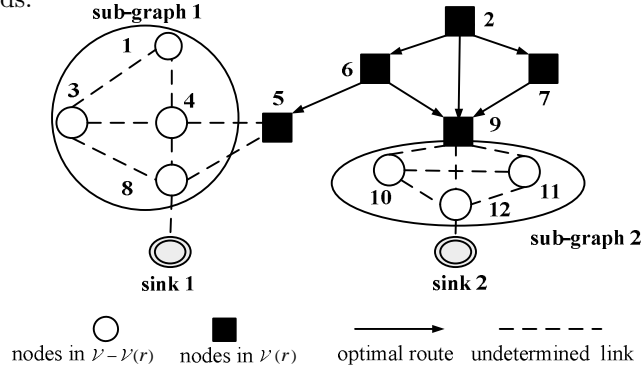

Fig. 5. An example for DMCR.

We construct the dual problem of the primary problem (14) by introducing the Lagrange multipliers λ_x, ν_x for energy constraint (20) and flow conservation constraint (19) respectively at each node $z \in N_x, x \in \mathcal{V} - \mathcal{V}(r)$, and the Lagrange multiplier $\rho_{x,y}$ for rate equality constraint (15) for node pairs $x \in \mathcal{V} - \mathcal{V}(r)$ and $y \in (\mathcal{V} - \mathcal{V}(r)) \cap N_x$. The corresponding Lagrangian is

$$
\begin{aligned}
&L(r, f, \lambda, \upsilon, \rho) \\
&= \sum_{x \in \mathcal{V} - \mathcal{V}(r)} r_x \Big(1 - \lambda_x \mathrm{E_s} - \upsilon_x + \sum_{y \in N_x \cap (\mathcal{V} - \mathcal{V}(r))} (\rho_{x,y} - \rho_{y,x})\Big) \\
&+ \sum_{x \in \mathcal{V} - \mathcal{V}(r)} \sum_{z \in N_x} (f_{x,z}(\upsilon_z - \upsilon_x - \mathrm{E_t}\lambda_x - \mathrm{E_r}\lambda_z)) \\
&+ \sum_{x \in \mathcal{V} - \mathcal{V}(r)} \lambda_x EC_x^{max}
\end{aligned}
\tag{21}
$$

The dual function is:

$$
D(\lambda, \nu, \rho) = \sup_{r \le r_x, 0 \le f_{x,z} \le c_{x,z}} L(r, f, \lambda, \upsilon, \rho)
\tag{22}
$$

Then we have the dual problem

Minimize $\qquad\qquad\qquad\qquad D(\lambda, \nu, \rho)$ $\qquad\qquad\qquad$ (23)

Subject to $\qquad\qquad\qquad\qquad \lambda \ge 0$ $\qquad\qquad\qquad\qquad$ (24)

Since the objective function of problem (14) is not strictly concave (i.e. linear) in both primary variables r and f, the solution should be recovered [Xiao et al. 2004]. We use the strictly concave term $\sum_{x \in \mathcal{V} - \mathcal{V}(r)} \log r_x$ to replace the objective function(14), because maximizing r_x is equal to maximizing $\log(r_x)$ with the equal constraint (15). We also add a small strictly concave regularization term $-\varepsilon \sum_{x \in \mathcal{V} - \mathcal{V}(r)} \sum_{z \in N_x} f_{x,z}^2$ for variable f to the objective function (4). The regularized objective of the primary problem (14) is

$$
\sum_{x \in \mathcal{V} - \mathcal{V}(r)} \log r_x - \varepsilon \sum_{x \in \mathcal{V} - \mathcal{V}(r)} \sum_{z \in N_x} f_{x,z}^2
\tag{25}
$$

By choosing ε small enough, the solution of the regulated problem can be arbitrary close to that of the original problem (14). The corresponding regularized dual problem is

$$
\textbf{Minimize} \sup_{r_x \geq r, 0 \leq f_{x,z} \leq c_{x,z}} \left(\sum_{x \in \mathcal{V} - \mathcal{V}(r)} \log r_x - r_x(\lambda_x E_s + \upsilon_x - \sum_{y \in N_x \cap (\mathcal{V} - \mathcal{V}(r))} (\rho_{x,y} - \rho_{y,x})) \right.
$$
$$
\left. + \sum_{x \in \mathcal{V} - \mathcal{V}(r)} \sum_{z \in N_x} (f_{x,z}(\upsilon_z - \upsilon_x - E_t \lambda_x - E_t \lambda_z) - \varepsilon f_{x,z}^2) + \sum_{x \in \mathcal{V} - \mathcal{V}(r)} \lambda_x EC_x^{max} \right)
$$

$\textbf{Subject to}$ $\qquad\qquad\qquad\qquad \lambda \geq 0$

(26)

We use a sub-gradient algorithm to solve the regulated dual problem (26). The algorithm starts from the initial values $\lambda_x^{(0)}$, $\nu_x^{(0)}$, $\rho_{x,y}^{(0)}$: For the kth iteration step of the sub-gradient algorithm, each node x in $\mathcal{V} - \mathcal{V}(r)$ solves the following two simple maximizing problems:

$$
r_x^{(k)} = \arg \max_{r \leq r_x} (\log r_x - r_x(\upsilon_x^{(k)} + \sum_{y \in (N_x \cap \mathcal{V} - \mathcal{V}(r))} (\rho_{xy}^{(k)} - \rho_{yx}^{(k)}) - \lambda_x^{(k)} E_s)) \tag{27}
$$

$$
f_{x,z}^{(k)} = \arg \max_{z \in N_x, c_{x,z} \geq f_{x,z} \geq 0} (f_{x,z}(\upsilon_z^{(k)} - \upsilon_x^{(k)} - E_t \lambda_x^{(k)} - E_r \lambda_z^{(k)}) - \varepsilon f_{x,z}^2) \tag{28}
$$

For each node $x \in \mathcal{V} - \mathcal{V}(r)$, the next step $\lambda_x^{(k+1)}$, $\upsilon_x^{(k+1)}$, $\rho_{x,y}^{(k+1)}, y \in N_x \cap (\mathcal{V} - \mathcal{V}(r))$ are computed as

$$
\lambda_x^{(k+1)} = |\lambda_x^{(k)} - l_{(k)}(EC_x^{max} - E_s r_x^{(k)} - E_r \sum_{z \in N_x} f_{z,x}^{(k)} - E_t \sum_{z \in N_x} f_{x,z}^{(k)})|_+ \tag{29}
$$

$$
\upsilon_x^{(k+1)} = \upsilon_x^{(k)} - l^{(k)} \left(\sum_{z \in N_x} (f_{x,z}^{(k)} - f_{z,x}^{(k)}) - r_x^{(k)} \right) \tag{30}
$$

$$
\rho_{x,y}^{(k+1)} = \rho_{x,y}^{(k)} - l^{(k)} \left(r_y^{(k)} - r_x^{(k)} \right) \tag{31}
$$

Where $l^{(k)}$ is the step length of the kth iteration, one condition for the convergence of the subgradient algorithm is (see Lemma 6.1 in Section 6.1):

$$
\sum_{k=1}^{\infty} (l^{(k)})^2 \to 0, \quad \sum_{k=1}^{\infty} l^{(k)} \to \infty \tag{32}
$$

For instance we can set $L^{(k)} = 1/k$. From (27)-(31), each node $x \in \mathcal{V} - \mathcal{V}(r)$ exchanges all updated flow and dual variables with its neighbors in $\mathcal{V} - \mathcal{V}(r)$, but only obtain $f_{z,x}$, λ_z, and ν_z from its neighbors $z \in \mathcal{V}(r) \cap N_x$ in every iteration step.

As we have mentioned, besides every node $x \in \mathcal{V} - \mathcal{V}(r)$, every node $z \in \mathcal{V}(r) \cap N_x$ is part of the current DMCR calculation to compute the the amount of data forwarded over links (z, x) should also be determined. The calculation is based on the following Theorem:

THEOREM 4.1. *For each node $z \in \mathcal{V}(r) \cap N_x, x \in \mathcal{V} - \mathcal{V}(r)$, its optimal incoming flows are sent by nodes only $p \in N_z \cap \mathcal{V}(r)$ and have been determined before the current DMCR calculation.*

PROOF. According to Lemma 6.3 in Section 6.2, since $z \in \mathcal{V}(r)$, for any forwarding path $P(p, z), p \in \mathcal{V}(r)$. Since p is z's neighbor, $p \in N_z \cap \mathcal{V}(r)$. □

According to Theorem 4.1, the forwarding rate $f_{z,p}$, $p \in N_z \cap \mathcal{V}(r)$ have been predetermined in previous DMCR-LMD cycles. Since r_z has also be determined (i.e. LM_z), node z only updates $f_{z,x}$, $x \in \mathcal{V} - \mathcal{V}(r)$ in its kth step as follows

$$f_{z,x}^{(k)} = \arg \max_{c_{z,x} \geq f_{z,x} \geq 0, z \in N_x \cap \mathcal{V}(r), x \in \mathcal{V} - \mathcal{V}(r)} (f_{z,x}(v_x^{(k)} v_z^{(k)} - \mathrm{E_t} \lambda_z^{(k)} - \mathrm{E_r} \lambda_x^{(k)}) - \varepsilon f_{z,x}^2) \quad (33)$$

Then, the $\lambda_z^{(k+1)}$ and $v_z^{(k+1)}$ are updated as

$$\lambda_z^{(k+1)} = |\lambda_z^{(k)} - l^{(k)}(EC_z^{max} - \mathrm{E_s} LM_z$$
$$- \mathrm{E_r} \sum_{p \in N_z \cap \mathcal{V}(r)} f_{p,z} - \mathrm{E_t} \sum_{x \in N_z \cap (\mathcal{V} - \mathcal{V}(r))} f_{z,x}^{(k)})|_+ \quad (34)$$

$$v_z^{(k+1)} = v_z^{(k)} - l^{(k)} \left(\sum_{x \in N_z} f_{z,x}^{(k)} - \sum_{p \in N_z \cap \mathcal{V}(r)} f_{p,z} - LM_z \right) \quad (35)$$

In summary, every node $x \in \mathcal{V} - \mathcal{V}(r)$ updates its primary and dual variables using (27)-(31), and every node $z \in \mathcal{V}(r) \cap N_x, x \in \mathcal{V} - \mathcal{V}(r)$ in updates its primary and dual variables (33)-(35). It is obvious that all information for (27)-(35) is either local or can be obtained by neighbor nodes, therefore DMCR is fully distributed.

DMCR calculates $C(r)$ (i.e. maximized r_x) and corresponding optimal routes (i.e. $f_{x,y}$ and $f_{y,x}, x \in \mathcal{V} - \mathcal{V}(r), y \in N_x$). One nice property of our approach is that the optimal routes calculated by DMCR is loop-free(Lemma 6.2 in Section 6).

In order to focus on the global multi-objective LM rate allocation problem, DMCR only adopts the basic dual-decomposition techniques: augmented Lagrangian and sub-gradient algorithms, which may result in relatively large convergence time. However, the convergence speed of DMCR could be significantly improved (e.g. hundreds of times faster) by using recent-proposed distributed convex optimization techniques such as [Necoara and Suykens 2008; Wan and Lemmon 2009].

When current DMCR computation completes, every node $x \in \mathcal{V} - \mathcal{V}(r)$ records $C(r)$ and data forwarding rates over its optimal incoming and outgoing links, for the forthcoming LMD to determine whether $C(r)$ is its LM rate or not (please see Figure 3 for the logical flow of our DMCR-LMD approach).

5. LM RATE DETERMINATION

Let $C(r)$ be the maximum common rate computed by the last DMCR. After computing $C(r)$, each node $x \in \mathcal{V} - \mathcal{V}(r)$ should determine whether $C(r)$ is its LM rate (i.e. LM_x) or not (Please refer to Figure 3 for the logical flow of our DMCR-LMD approach.). This can be achieved by solving the following LP problem (i.e. the MSR problem)

Maximize $\qquad\qquad\qquad\qquad r_x \qquad\qquad\qquad\qquad\qquad$ (36)

Subject to $\qquad\qquad r_y = C(r), \forall y \in \mathcal{V} - \mathcal{V}(r) - \{x\} \qquad\qquad$ (37)

$$\text{constraints } (16) - (20)$$

If the maximized r_x is equal to $C(r)$, then the LM rate of node x, LM_x is equal to $C(r)$; otherwise, LM_x is larger than $C(r)$. However, this approach is not only centralized, but also requires solving $|\mathcal{V} - \mathcal{V}(r)|$ LP problems to determine the LM rates of all nodes in $\mathcal{V} - \mathcal{V}(r)$. Furthermore, each of these LP problems has the similar complexity of MCR problem (14). Therefore, this approach is obvious too heavy for resource-limited SP-WSNs.

For readability, we call sensor nodes in $\mathcal{V} - \mathcal{V}(r)$ whose LM rates are equal to and larger than $C(r)$ as New LM Rate (NLMR) nodes and Larger LM Rate (LLMR) nodes respectively. Obviously, the goal of solving $|\mathcal{V} - \mathcal{V}(r)|$ LP problems (36) is to determine all NLMR nodes and LLMR nodes. In this section, we develop LMD, a fully distributed graph-theoretic scheme to achieve this goal. In contrast to solve $|\mathcal{V} - \mathcal{V}(r)|$ LP problems one by one, LMD manages to determine the state of each node $x \in \mathcal{V} - \mathcal{V}(r)$ simultaneously (i.e. x is a NLMR node or LLMR node), with extremely low overhead.

5.1. Graph-theoretic Understanding of the LM rate Determination Problem

Before presenting the LMD scheme, we first analyze the LM rate determination problem from graph theory perspective. We define a temporary graph formed after the last DMCR:

Definition 5.1 (*Temporary Graph*). A temporary graph $\mathcal{G}(\mathcal{S} \cup \mathcal{V}, \mathcal{F}, r)$ forms after the calculation of the last $C(r)$, where $\mathcal{F}$ is the set of allocated forwarding rate over every link (i.e. end-to-end routes from network-wide perspective), which is calculated by the last DMCR and previous DMCR-LMD cycles.
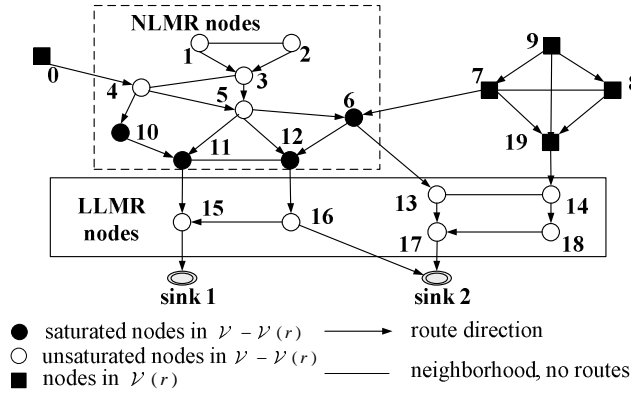

Fig. 6. An example of a temporary graph.

Figure 6 illustrates an example of a temporary graph[2]. Actually, the process of the LM^* calculation can be seen as determining the new LM rates and corresponding temporary graphs step by step. The first temporary graph forms after the calculation of $C(0)$, and the last temporary graph forms when all LM rates have been found and represents the optimal routing corresponding to LM^*.

Definition 5.2 (*Saturated Node and Unsaturated Node*). We call a sensor node x in a temporary graph $\mathcal{G}(\mathcal{S} \cup \mathcal{V}, \mathcal{F}, r)$ a saturated node, if

$$\mathrm{E_s} r_x + \mathrm{E_r} \sum_{y \in N_x} f_{y,x} + \mathrm{E_t} \sum_{y \in N_x} f_{x,y} - EC_x^{max} = 0$$

or

$$f_{x,y} = c_{x,y}, \forall y \in N_x$$

; otherwise, x is called an unsaturated node.

───────

[2]It worth noting that Figure 6 is a temporary graph, but Figure 5 is not a temporary graph, because Figure 5 describes the network states before a DMCR, rather than after a DMCR.

Definition 5.3 (*Path and Forwarding Path*). Given a temporary graph $\mathcal{G}(\mathcal{S} \cup \mathcal{V}, \mathcal{F}, r)$, a path $P(s, d)$ with source s and destination d, $s \in \mathcal{V}$, $d \in \mathcal{S} \cup \mathcal{V}$, is a sequence of links. If $f_{x,y} > 0$, $\forall (x, y) \in P(s, d)$, then $P(s, d)$ is a forwarding path. If $P(s, d)$ is a forwarding path, it is called the source node $s's$ downstream path, and the destination node $d's$ upstream path. If all sensor nodes in a path is unsaturated, then we call this path an unsaturated path, otherwise we call it a saturated path.

An arbitrary path may not be a forwarding path, but each sensor node $x \in \mathcal{V}$ must have a downstream forwarding path to the sink in every temporary graph $\mathcal{G}(\mathcal{S} \cup \mathcal{V}, \mathcal{F}, r)$, $r \geq C(0)$, because x must have a non-zero sensing rate (i.e. $r_x \geq C(0) > 0$) for any temporary graph, and the flow injected into the network (i.e. r_x) must be transmitted to the sink, according to the flow conservation law.

Theorem 6.8 in Section 6.2 provides a condition to determine the state (NLMR or LLMR) of a node in $\mathcal{V} - \mathcal{V}(r)$: Let $P(x, s) \in \mathcal{G}(\mathcal{S} \cup \mathcal{V}, \mathcal{F}, r)$ be an arbitrary path from a node $x \in \mathcal{V} - \mathcal{V}(r)$ to an arbitrary sink $s \in \mathcal{S}$. Except for the destination s, this path consists of nodes only in $\mathcal{V} - \mathcal{V}(r)$. If all such paths are unsaturated, then x is a LLMR node. Otherwise, x is a NLMR node.

From theorem 6.8, saturated nodes in $\mathcal{V} - \mathcal{V}(r)$ form a saturated cut or multiple saturated cuts between a set of unsaturated nodes in $\mathcal{V} - \mathcal{V}(r)$ and all sinks in a given temporary graph. Take Figure 6 for instance, saturated nodes 10, 11, 12, and 6 form a cut which separates the set of unsaturated nodes 1, 2, 3, 4, and 5 from the two sinks. All possible forwarding paths from these nodes to the two sinks, which only consists of nodes in $\mathcal{V} - \mathcal{V}(r)$, must via the saturated nodes 6, 10, 11, and 12. According to Theorem 6.8, nodes 1 2, 3, 5, 6, 10, 11 and 12 are NLMR nodes and other unsaturated nodes in $\mathcal{V} - \mathcal{V}(r)$ are LLMR nodes.
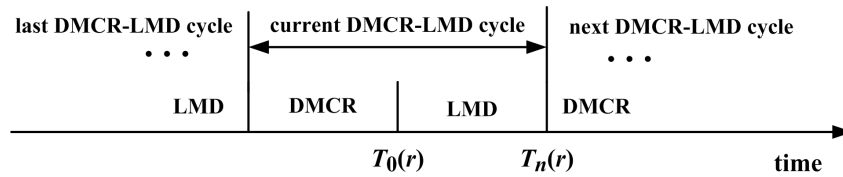
5.2. The LMD scheme

Fig. 7. Illustration of LMD operating durations.

The Pseudo-code of the LMD scheme is described in Algorithm 2. The LMD scheme multicasts a one-hop control packet, Max-min Notice (MN) packet. As shown in Figure 7, LMD starts at time $T_0(r)$ when the last DMCR is finished, and complete at time $T_n(r)$ when next DMCR starts. The upper bound of the duration$[T_0(r), T_n(r)]$ is determined by the per-hop transmission delay and network diameter (see the proof of Lemma 6.14 in Section 6.2), which can be easily estimated in practice.

At $T_0(r)$, every node x in $\mathcal{V} - \mathcal{V}(r)$ records $C(r)$ and it local flow information calculated by the last DMCR. Consequently, x can locally determine its state (i.e. saturated or unsaturated). At $T_0(r)$, every node initiates as a LLMR node (line 1, in self check at $T_0(r)$), it then checks whether it is a saturated node or not (line 3) by simply calculating its local constraint (20). After self-checking, every saturated sensor node x in $\mathcal{V} - \mathcal{V}(r)$ sets its state as NLMR and multicasts a one-hop MN packet to all its upstream neighbors in $\mathcal{V} - \mathcal{V}(r)$ (lines 3-7). During the MN packet transmission phase, the MN packets generated by the saturated nodes in $\mathcal{V} - \mathcal{V}(r)$ are relayed through their upstream neighbors in $\mathcal{V} - \mathcal{V}(r)$.

In the MN packet transmission phases $(T_0(r), T_n(r))$, when an unsaturated node x receives a MN packet from its downstream neighbor y (line 1), x knows it is a NLMR node and sets

its state as NLMR (line 2), then it records the ID of the sender of this packet (line 3). Then x checks whether its every downstream neighbor in $D_x(r)$ has sent it a MN packet and its upstream neighbor set $U_x(r)$ is non-empty (line 4). If yes, x multicasts a MN packet to all node(s) in $U_x(r)$ (line 5).

ALGORITHM 2: Pseudo-code of the LMD for a sensor node $x \in \mathcal{V} - \mathcal{V}(r)$.

Control Packets:
Maxmin Notice(MN)

Variables:
$x.state$: this value could be NLMR or LLMR, determine this value is the objective of LMD.
$N_x(r) = N_x \cap (\mathcal{V} - \mathcal{V}(r))$.
$D_x(r) = \{y | y \in N_x(r), f_{x,y} > 0\}$: the set of x's all downstream neighbors in set $N_x(r)$.
$U_x(r) = \{y | y \in N_x(r), f_{y,x} > 0\}$: the set of x's all upstream neighbors in set $N_x(r)$.
SN_x : the set of neighbors who have sent MN packets to x.

Function:
multicast$(U_x(r))$: multicast a MN packet to all nodes in the set $U_x(r)$.

Self Check at $T_0(r)$
1: $x.state \leftarrow$ LLMR;
2: $SN_x \leftarrow \varnothing$;
3: **if** x is saturated **then**
4:    $x.state \leftarrow$ NLMR;
5:    **if** $U_x(r) \neq \varnothing$ **then**
6:      **multicast** $(U_x(r))$;
7:    **end if**
8: **end if**

Receive a MN packet from $y \in D_x(r)$ ***before*** $T_n(r)$
1: **if** x is unsaturated **then**
2:    $x.state \leftarrow$ NLMR;
3:    $SN_x \leftarrow SN_x \cup \{y\}$;
4:    **if**$(D_x(r) = SN_x) \wedge (U_x(r) \neq \varnothing)$ **then**
5:      **multicast**$(U_x(r))$;
6:    **end if**
7: **end if**

Take Figure 6 for example, nodes 6, 10, 11, and 12 know that they are saturated after self-checking and send MN packets to their upstream neighbors 4 and 5 respectively. At $T_n(r)$, the unsaturated nodes 1, 2, 3, 4, and 5 have received MN packets but only nodes 3 and 5 have been sent a MN packet respectively. The MN packet transmission procedure is: firstly $10 \rightarrow 4$ and $\{11, 12, 6\} \rightarrow 5$, then $5 \rightarrow \{3, 4\}$, finally $3 \rightarrow \{1, 2\}$.

Consequently, every node $x \in \mathcal{V} - \mathcal{V}(r)$ manages to determine its state (LLMR or NLMR) at $T_n(r)$ as follows: x is a NLMR node, if it is a saturated node, or an unsaturated node and has received a MN packet; x is a LLMR node, if x is an unsaturated node and has not receive any MN packet.

At $T_n(r)$, all LLMR nodes and their neighbors will be part of the calculation of the next DMCR-LMD cycle. It is obvious that each NLMR node transmits at most one MN packet for entire procedure of the LM^* calculation. Theorem 6.16 in Section 6 demonstrates the accuracy of LMD scheme.

6. PERFORMANCE ANALYSIS

In this section, we provide rigorous analysis of the system performance. The main results are as follows.

Firstly, DMCR converges arbitrarily close to the optimal solution of problem (14) (Lemma 6.1). The routes computed by any DMCR is loop-free, including the optimal routes with regards to LM^* (Lemma 6.2).

Secondly, LMD let each node in $\mathcal{V} - \mathcal{V}(r)$ know it is a NLMR node or LLMR node before $T_n(r)$ (Theorem 6.16). LMD determines the LM rate for every node with at most 1 control packet for the whole procedure of LM^* calculation (Theorem 6.17).

Thirdly, the whole system (LPM, DMCR, and LMD) converges to optimal LM rate allocation LM^* and corresponding optimal routes (Theorem 6.18).

6.1. Analysis of DMCR

LEMMA 6.1. *Each DMCR can converges arbitrarily close to the optimal solution of problem (14) as $\varepsilon \to 0$.*

PROOF. Let the optimal dual variables be $(\lambda, \nu, \rho)^*$ that minimize the regulated dual function $D_\varepsilon(\lambda, \nu, \rho)$ (26). At the kth step of the subgradient algorithm, we have

$$\|(\lambda, \nu, \rho)^{(k+1)} - (\lambda, \nu, \rho)^*\|_2^2$$
$$= \|(\lambda, \nu, \rho)^{(k)} - l^{(k)}g^{(k)} - (\lambda, \nu, \rho)^*\|_2^2$$
$$= \|(\lambda, \nu, \rho)^{(k)} - (\lambda, \nu, \rho)^*\|_2^2 - 2l^{(k)}g^{(k)T}((\lambda, \nu, \rho)^{(k)} - (\lambda, \nu, \rho)^*) + (l^{(k)})^2\|g^{(k)}\|_2^2$$
$$\leq \|(\lambda, \nu, \rho)^{(k)} - (\lambda, \nu, \rho)^*\|_2^2 - 2l^{(k)}(D_\varepsilon((\lambda, \nu, \rho)^{(k)}) - D_\varepsilon^*((\lambda, \nu, \rho)^*)) + (l^{(k)})^2\|g^{(k)}\|_2^2$$

where $\|\cdot\|$ is the Euclidean norm operator and the last inequality is based on the definition of subgradient. Applying the inequality above recursively, we have

$$\|(\lambda, \nu, \rho)^{(1)} - (\lambda, \nu, \rho)^*\|_2^2 - 2\sum_{i=1}^{k} l^{(i)}(D_\varepsilon((\lambda, \nu, \rho)^{(i)}) - D_\varepsilon^*((\lambda, \nu, \rho)^*)) + \sum_{i=1}^{k}(l^{(i)})^2\|g^{(i)}\|_2^2$$
$$\geq \|(\lambda, \nu, \rho)^{(k+1)} - (\lambda, \nu, \rho)^*\|_2^2 \geq 0$$

Assume the subgradient is bounded $\|g^{(k)}\|_2^2 \leq G$ (this can be easily achieved by adding sufficient large upper bounds for both r and f), then we have

$$D_\varepsilon((\lambda, \nu, \rho)^{(k)}) - D_\varepsilon^*((\lambda, \nu, \rho)^*) \leq \frac{\|(\lambda, \nu, \rho)^{(1)} - (\lambda, \nu, \rho)^*\|_2^2 + G^2 \sum_{i=1}^{k}(l^{(i)})^2}{2\sum_{i=1}^{k} l^{(i)}}$$

According to step size updating rule (32), i.e. $\sum_{i=1}^{\infty}(l^{(k)})^2 \to 0, \quad \sum_{i=1}^{\infty} l^{(i)} \to \infty$, we have

$$\lim_{k\to\infty} D_\varepsilon((\lambda, \nu, \rho)^{(k)}) = D_\varepsilon^*((\lambda, \nu, \rho)^*)$$

Consider the strong duality of the system and the expression of $D_\varepsilon(\cdot)$, we can conclude that as $\varepsilon \to 0$, DMCR converges to the optimal solution of problem (14). □

LEMMA 6.2. *The routes calculated by any DMCR are loop-free.*

PROOF. Let $\lambda_x^*, \nu_x^*, f_{x,y}^*$ be the optimal value of $\lambda_x, \nu_x, f_{x,y}$ respectively. From equations (18) and (23), we have:

$$f_{x,z}^* = \begin{cases} \frac{-(\Delta\upsilon^* + E_t\lambda_x^* + E_r\lambda_z^*)}{2\varepsilon} & \frac{-(\Delta\upsilon^* + E_t\lambda_x^* + E_r\lambda_z^*)}{2\varepsilon} > 0 \\ 0 & \frac{-(\Delta\upsilon^* + E_t\lambda_x^* + E_r\lambda_z^*)}{2\varepsilon} \leq 0 \end{cases}$$

Where $\Delta\nu_x^* = \nu_x^* - \nu_z^*$. Suppose there is a loop such that $f_{x1,x2}^* > 0, f_{x2,x3}^* > 0, \cdots, f_{xn,x1}^* > 0$, which implies:

$$\upsilon_{x1}^* - \nu_{x2}^* < -\mathrm{E_t}\lambda_{x1}^* - \mathrm{E_r}\lambda_{x2}^*$$

$$\cdots$$

$$\upsilon_{xn}^* - \nu_{x1}^* < -\mathrm{E_t}\lambda_{xn}^* - \mathrm{E_r}\lambda_{x1}^*$$

Taking a telescopic sum of above inequations from 1 to n, we get:

$$0 < -(\mathrm{E_t} + \mathrm{E_r}) \sum_{i=1}^{n} \lambda_{x_i}^*$$

Which is impossible, because $(\mathrm{E_t} + \mathrm{E_r}) \sum_{i=1}^{n} \lambda_{x_i}^*$ is always non-negative. □

6.2. Analysis of LMD

LEMMA 6.3. *Consider an arbitrary forwarding path $P(x,y)$ in a temporary graph $\mathcal{G}(\mathcal{S} \cup \mathcal{V}, \mathcal{F}, r)$, if $y \in \mathcal{V}(r)$, then $x \in \mathcal{V}(r)$.*

PROOF. We prove Lemma 6.3 by contradiction. Suppose there is a forwarding path $P(x,y), x \in \mathcal{V} - \mathcal{V}(r)$ and $y \in \mathcal{V}(r)$, then we have $LM_x \geq C(r) > LM_y$. Therefore, we can set $r_x = C(r) - \Delta r$ and $r_y = LM_y + \Delta r$, where $\Delta r < (LM_y + LM_x)/2$, by reducing the Δr amount of flow over $P(x,y)$. Since the amount of flow over $P(x,y)$ is reduced, the unchanged sensing rates of other nodes on $P(x,y)$ are still feasible. As a result, we have a new feasible rate allocation, in which $r_x = C(r) - \Delta r, r_y = LM_y + \Delta r, r_m = C(r), \forall m \in \mathcal{V} - \mathcal{V}(r) - \{x\}$, and $r_n = LM_n, \forall n \in \mathcal{V}(r) - \{y\}$. This new rate assignment is lexicographically greater than LM^*, which conflicts with the fact that LM^* is the unique lexicographically optimal feasible rate allocation (Theorem 2.3). □

Lemma 6.3 means that data traffic generated by any node in $\mathcal{V} - \mathcal{V}(r)$ will not pass any node in $\mathcal{V}(r)$.

Definition 6.4. Real and Fake saturated nodes: Consider a saturated node x, and an unsaturated node $y, x, y \in \mathcal{V} - \mathcal{V}(r)$. If there exist two forwarding paths $P(y,x)$ and $P(y,s)$ $s \in \mathcal{S}$ in $\mathcal{G}(\mathcal{S} \cup \mathcal{V}, \mathcal{F}, r)$, where $P(y,s)$ consists of pure unsaturated nodes, then x is called a fake saturated node. Otherwise, it is a real saturated node.

LEMMA 6.5. *For a temporary graph $\mathcal{G}(\mathcal{S} \cup \mathcal{V}, \mathcal{F}, r)$, the probability that a fake saturated node exists is zero.*

PROOF. Suppose there is a fake saturated node x in a temporary graph in $\mathcal{G}(\mathcal{S}+\mathcal{V}, \mathcal{F}, r)$, such that an unsaturated node y has a forwarding path to x and a forwarding path to a sink. In this case, x can also be unsaturated by reducing the Δr amount of flow on $P(y,x)$ and increasing the same amount of flow on $P(y,s)$. All nodes in $\mathcal{V}-\mathcal{V}(r)$ can still keep the sensing rate of $C(r)$. As long as Δr is smaller than the residual capacity of path $P(y,s)$, this is still a feasible solution to DMCR. Hence, node x is not the bottleneck of the network. Let λ_x^* be the optimal dual variable λ_x computed by any DMCR. According to complementary slackness condition [Boyd and Vandenberghe 2004], $\lambda_x > 0$, if x is saturated; $\lambda_x = 0$, otherwise. Since λ_x is a continuous variable, there exists infinite $\lambda_x > 0$ but only one $\lambda_x = 0$. Therefore, the probability that x is a fake saturated node is zero. □

For instance, the probability that node 15 in Figure 6 is a saturated node is zero.

LEMMA 6.6. *For a temporary graph $\mathcal{G}(\mathcal{S} \cup \mathcal{V}, \mathcal{F}, r)$, all real saturated nodes in $\mathcal{V} - \mathcal{V}(r)$ are NLMR nodes.*

PROOF. Supposed a real saturated node is x is a LLMR node, then we can write $r_x = \Delta r + C(r)$, where $\Delta r > 0$ and all other nodes in $\mathcal{V} - \mathcal{V}(r)$ keep the sensing rate of $C(r)$. To guarantee feasibility, x has to reduce its Δr relay flow of its upstream nodes. Let y be an arbitrary upstream node of x, then y has to reduce the flow on $P(y, x)$ and increase the same amount of flow on a forwarding path $P(y, s)$, according to the flow conservation law. To relay the additional flow, $P(y, s)$ must consist of all unsaturated nodes, which means x is a fake saturated node. This conflicts with the supposition. □

LEMMA 6.7. *For a forwarding path $P(x, y)$ in $\mathcal{G}(\mathcal{S} \cup \mathcal{V}, \mathcal{F}, r), x, y \in \mathcal{V} - \mathcal{V}(r)$. If x is unsaturated and y is saturated, then there must exist at least one saturated node on every path $P(x, s), \forall s \in \mathcal{S}$.*

PROOF. According to Lemmas 6.5 and 6.6, lemma 6.7 obviously holds. □

Since there is no fake saturated node, in later discussions we will directly use the term "saturated node" as a shorthand notion of "real saturated nodes".

THEOREM 6.8. *Let $P(x, s) \in \mathcal{G}(\mathcal{S} \cup \mathcal{V}, \mathcal{F}, r)$ be an arbitrary path from a node $x \in \mathcal{V} - \mathcal{V}(r)$ to an arbitrary sink $s \in \mathcal{S}$. Except for the destination s, this path consists of nodes only in $\mathcal{V} - \mathcal{V}(r)$. If all such paths are unsaturated, then x is a LLMR node. Otherwise, x is a NLMR node.*

PROOF. If x is a saturated node, then it is a NLMR node according to lemma 6.6.

If x is an unsaturated node, suppose x is a LLMR node, then we can write $r_x = \Delta r + C(r)$, and $r_y = C(r)$ and $r_z = LM_z$, where $\Delta r > 0, y \in \mathcal{V} - \mathcal{V}(r)$, and $z \in \mathcal{V}(r)$. According to flow conservation law, there must exist a forwarding path $P(x, s), s \in \mathcal{S}$, which is able to relay the incremental flow $\Delta r, i.e. P(x, s)$ must consist of pure unsaturated sensor nodes and the minimal residual capacity of node on $P(x, s)$ is larger than Δr, which conflict with the supposition. □

Definition 6.9 (*Temporary Upstream Path Enclosure (TUPE)*). For a given $\mathcal{G}(\mathcal{S} \cup \mathcal{V}, \mathcal{F}, r)$. Let $\mathcal{P}(x)$ be the union of unsaturated nodes in $\mathcal{V} - \mathcal{V}(r)$ on all upstream paths of an arbitrary saturated node $x \in \mathcal{V} - \mathcal{V}(r)$. Denote the saturated cut X be a set of saturated node in $\mathcal{V} - \mathcal{V}(r)$ such that $\forall x, y \in X, \mathcal{P}(x) \cap \mathcal{P}(y) \neq \varnothing$. We define a TUPE as $\mathbb{P}(X) = \cup_{x \in X} \mathcal{P}(x)$.

It can be seen all nodes in $\mathbb{P}(X)$ are unsaturated in $\mathcal{V} - \mathcal{V}(r)$ and X separates these unsaturated nodes from all sinks. For example, in Figure 6, $X = \{6, 10, 11, 12\}$ and $\mathbb{P}(X) = \{1, 2, 3, 4, 5\}$.

LEMMA 6.10. *All unsaturated nodes in a TUPE $\mathbb{P}(X)$ are NLMR nodes.*

PROOF. Since every unsaturated sensor node y in a TUPE must has one downstream saturated node, there must be a saturated node on y·s all downstream paths, according to Lemma 6.7. Furthermore, according to theorem 6.8, y is a NLMR node. □

Definition 6.11. For a node $y \in \mathbb{P}(X)$, let $\mathcal{P}(y, X)$ be set of all forwarding paths $P(y, x), x \in X$. Denote $D(y, X)$ as the maximal hop count of all forwarding paths in $\mathcal{P}(y, X)$. In addition, denote $Dis(y, X) = 0$, if $y \in X$.

For example, in Figure 6, $X = \{6, 10, 11, 12\}, D(5, X) = 1, D(4, X) = 2$, and $Dis(10, X) = 0$.

LEMMA 6.12. *For any node $y \in \mathbb{P}(X), Dis(y, X) \leq |\mathbb{P}(X)|$.*

PROOF. According to Lemma 6.2, there is no loop in any temporary graph. Consequently, the maximal possible $Dis(y, X) = |\mathbb{P}(X)|$, when all nodes are distributed as a linear topology. □

LEMMA 6.13. *For any node $y, z \in \mathbb{P}(X)$, if z is a downstream neighbor of y, then $Dis(z, X) < Dis(y, X)$.*

PROOF. According to Lemma 6.2, z will not be an upstream node of y. Hence, it is obvious that $Dis(z, X) < Dis(y, X)$ by definition. □

LEMMA 6.14. *Every node $y \in \mathbb{P}(X)$ can know it is a NLMR node before $T_n(r)$.*

PROOF. Let the duration of transmitting a MN packet be ΔT. At $T_0(r)$, all saturated nodes in X multicasts a MN packet to all their upstream neighbors. At $T_0(r) + \Delta T$, if $Dis(y, X) = 1$, y can receive MN packets from all its downstream neighbors belong to X. Consequently, it can know that it is a NLMR node and multicasts a MN packet to its all upstream neighbors. Otherwise, y just stores the packet. At $T_0(r) + 2\Delta T$, if $D(y, X) = 2$, then it have two kinds of downstream neighbors $z : Dis(z, X) = 1$ or $Dis(z, X) = 0$, according to Lemma 6.13. It is obvious that in both cases, z has transmitted a MN packet to y. Consequently, y can know it is a NLMR node and multicasts a MN packet to all its upstream neighbors. By repeating this process, we can see that every y can receive MN packets from its all downstream neighbors at $T_0 + Dis(y, X)\Delta T$. According to Lemma 6.12, we can set $|\mathbb{P}(X)|\Delta T \geq T_n(r) \geq \max_{y \in \mathbb{P}(X)}\{Dis(y, X)\Delta T\}$ such that $y \in \mathbb{P}(X)$ can successfully know it is a NLMR node before $T_n(r)$. □

LEMMA 6.15. *If a node x does not belong to any TUPE, it will know it is a LLMR node at $T_n(r)$.*

PROOF. First, it is obvious that x is LLMR node. Second, since every $P(x, s)$ consists of pure unsaturated node, x can not receive any MN packet, i.e. x will know it is a LLMR node at $T_n(r)$. □

THEOREM 6.16. *LMD lets each node in $\mathcal{V} - \mathcal{V}(r)$ know it is a NLMR node or LLMR node before $T_n(r)$ in a fully distributed way.*

PROOF. Saturated nodes can know they are NLMR nodes by self-checking, according to Lemma 6.6. There are two categories of unsaturated nodes: in a TUPE and not in any TUPE. Lemmas 6.14 and 6.15 respectively state that all unsaturated nodes can know they are NLMR nodes or LLMR nodes before $T_n(r)$. □

6.3. Analysis of The Whole System

THEOREM 6.17. *Each node sends at most a MN packet for the whole procedure of the LM^* calculation.*

PROOF. We prove this theorem by contradiction. In a DMCR-LMD cycle, a sensor node $x \in V$ can send at most one MN packet to its upstream neighbor(s) only if it belongs to a TUPE of the temporary graph in this DMCR-LMD cycle. According to Lemma 6.10 and Theorem 6.16, x will determine its LM rate at the end of this DMCR-LMD cycle. Suppose x sends more than one MN packet, which means that x determines multiple LM rates in several DMCR-LMD cycles. This conflicts with the fact that LM_x is unique. □

THEOREM 6.18. *Our approach LPM+DMCR+LMD converges to the optimal LM rate allocation and corresponding optimal routes.*

PROOF. From Theorem 3.6, LPM computes EC_x^{max} for each node $x \in \mathcal{V}$. With the energy consumption constraints provided by LPM, The system begin to run multiple DMCR-LMD cycles. At each DMCR-LMD cycle, DMCR first manages to converge to the maximum common rate and corresponding temporary graph (Lemma 6.1) and then LMD can determine LM rate (Theorem 6.16). As each DMCR-LMD cycle can determine at least one

LM rate, the whole system will converge to the optimal within at most $|\mathcal{V}|$ DMCR-LMD cycles. $\square$

7. DISCUSSION OF PRACTICAL ISSUES

In practical SP-WSN, the wireless channel quality is time-varying, and data transmissions are unreliable, due to the stochastic nature of wireless channels. This has significant impact on the practical implementation of our SP-WSNs. In order to guarantee the optimality of the DMCR-LMD approach in practical SP-WSNs, the following practical issues should be considered:

Establishment of the stable SP-WSN Graph $\mathcal{G}(\mathcal{V} \cup \mathcal{S}, \mathcal{L})$. Our DMCR-LMD approach assume that the topology of the SP-WSN $\mathcal{G}(\mathcal{V} \cup \mathcal{S}, \mathcal{L})$ remains constant over a prediction interval. However, since the wireless link capacity is time-varying, the topology of $\mathcal{G}(\mathcal{V} \cup \mathcal{S}, \mathcal{L})$ may also change. In order to avoid this, a wireless link (x, y) is consider in $\mathcal{L}$, only if both PRR values $PRR_{x,y}$ and $PRR_{y,x}$ are higher than a threshold $\text{PRR}_{\min}$ (e.g. 90%). All weak links are not considered. By doing this, although link capacity is time-varying, the topology of $\mathcal{G}(\mathcal{V} \cup \mathcal{S}, \mathcal{L})$ remains static over time with an extremely high probability. This is because the probability of a slow-fading link $(x, y) \in \mathcal{L}$ fails (i.e. $c_{x,y} = 0$) for a given duration (e.g. a prediction interval) is close to zero, according to both practical observations [Baccour et al. 2012] and wireless communication theory (e.g. Rayleigh fading channel).

Reliable control packet transmissions. To ensure the successful transmission of control information, all control packets of DMCR and LMD should be set as link layer unicast messages (i.e. with link layer acknowledgements). A control message will be transmitted again, until it is successfully received. This will not increase the communication overhead significantly (less than twice), because the expected transmission times or the ETX [De Couto et al. 2003; Gnawali et al. 2009] value over each link, is small (smaller than $1/\text{PRR}_{\min}^2$).

8. EVALUATION

8.1. Evaluation of LPM

In this subsection, we evaluate LPM using our own developed solar powered sensor node shown in Figure 8 (a), which consists of following components: a MicaZ mote with $E_{\min} = 30.4$ J and $E_{\max} = 167.9$ J, a 9×3.8 cm^2 solar panel, a battery with capacity $B_{\max} = 10.7$ KJ, efficiency $\eta = 74.8\%$, and a circuit board to power the MicaZ mote and control the battery recharging and discharging process. We set the duration of a slot $\delta = 30$ minutes and the length of prediction interval as $L = 6$. The experiment last for two days which consists 96 slots and 16 prediction intervals.

Figure 8 (b) and (c) show the results of LPM algorithm for two day, which consists of 12 prediction intervals. We developed a simple solar power prediction algorithm based on weather-condition awareness and the Exponentially Weighted Moving Average(EWMA) Scheme[Cox 1961] to predicted $h_x^i, i \in I$ in the beginning of every prediction interval. The initial battery level of the first prediction interval was 1 KJ and the final state constraint φ was set as 1 KJ for all prediction intervals.

It is obvious that both the node's battery level and energy consumption varies according to the harvested solar power dynamics. It can be seen that LPM problem (9) has a feasible solution during most prediction intervals of daytime. However, problem (9) does not have a feasible solution during every night. As a result, LPM algorithm returns $E_{\min}$ and the battery level linearly reduces caused by both $E_{\min}$ and E_{leak}. From Figure 8 (b) and (c), we can see that a properly selected φ is robust to solar prediction errors and poor harvesting opportunities, which manages to avoid battery exhausting in real world implementation.

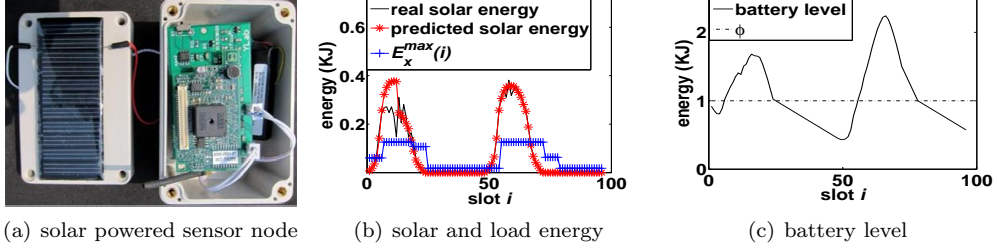(a) solar powered sensor node (b) solar and load energy (c) battery level

Fig. 8.   Evaluations of LPM on a real solar-powered sensor node.
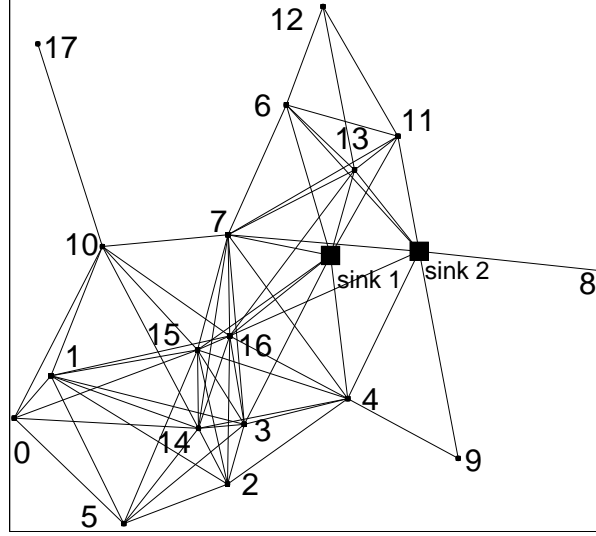

Fig. 9.   The topology of a 20-node SP-WSN.

8.2. Performance of the Whole System: A Case Study

In this subsection, we simulate the performance of our distributed approach in one prediction interval by considering a randomly deployed SP-WSN with 18 solar powered sensor nodes and two sinks, illustrated in Figure 9 (a). Every sensor node sends data to any of the two sinks. The real outdoor solar energy data collected in [sdf] was used in our simulations. The data was appropriately scaled to create a solar profile for a solar panel (9×3.8 cm^2) and assign to each sensor node with a random vibration of $\pm 5\%$. We set δ=5 minutes[3] and L=96 (i.e. 8 hours). The parameters η , η, B_{max} were set as the same as in the LPM experiment and E_{leak}, E_{min}, E_{max} are 6 times less than that in LPM experiment (since the value of δ in this experiment is 6 times less than that in LPM experiment). We set initial battery level and final state constraint parameter as $B_x^1 = 500 \pm 50$J and $\varphi = 500 \pm 50$J respectively.

We set E_t, E_r, and E_s as 3×10^{-7}, 3×10^{-7}, and 1.5×10^{-7} J/bit respectively[4]. In addition, we also set the the duration $[T_0(r), T_n(r)]$=10 seconds for each LMD and the regularization parameter ε=0.02 and step size l_k=0.6/k for the kth iteration step of the DMCR.

---

[3]we set δ=5 minutes according to the solar data granularity.

[4]E_t, E_r were approximated by considering power consumption of MicaZ mote (MCU and CC2420 radio), and maximum data transmission of CC2420. E_s are approximated by power consumption of typical sensors, TinyOS timer (e.g. several readings per millisecond), and ADC (10 bits).

(a) the calculated LM^* (b) the first DMCR convergence

(c) the second DMCR convergence

(d) the first temporary graph   (e) the second temporary graph   (f) battery exhausting slot number
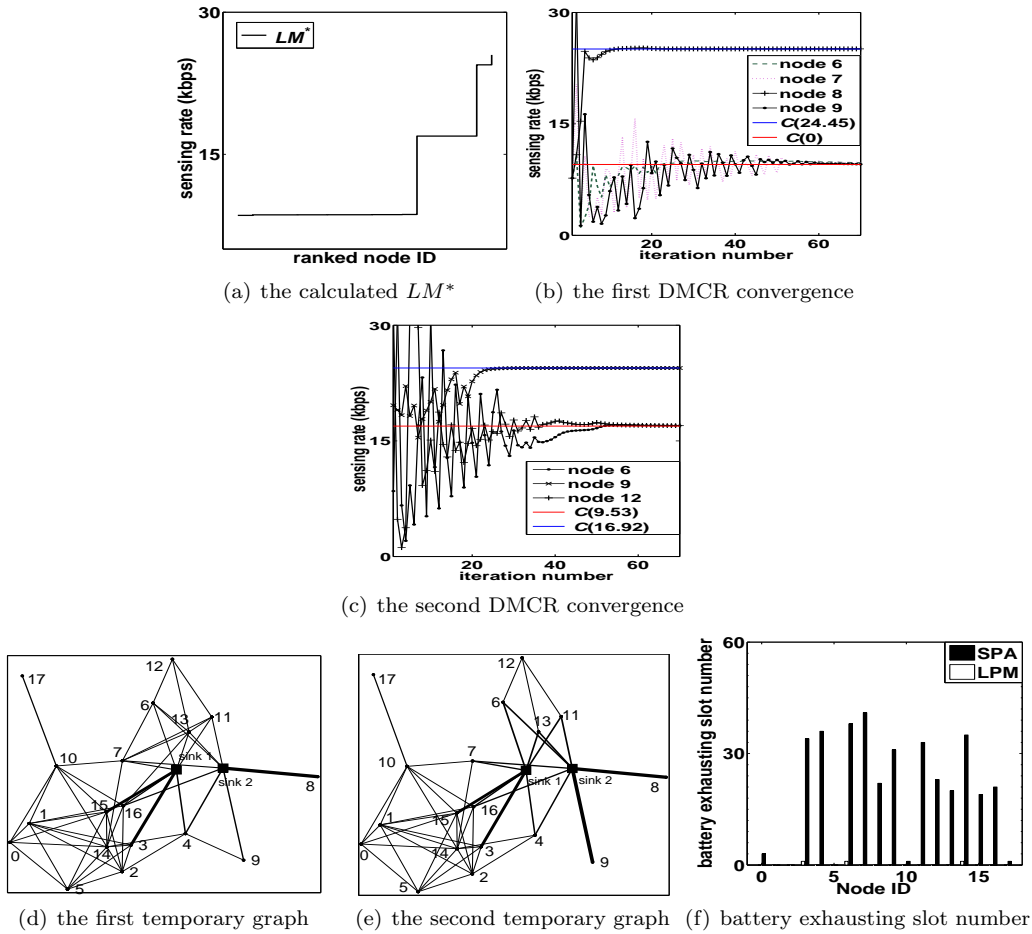
Fig. 10.   Simulation results of a 20-node SP-WSN.

After each node calculates its maximum feasible energy consumption by LPM, LM^* was computed by two DMCR-LMD iterations. Figure 10 (a) shows the LM rate allocation LM^* calculated by a centralized LP solver [Grant and Boyd]. Figure 10 (b) and (c) show the convergence of the two DMCR computations. It is clear that after several iterations, sensor nodes' rates tend to be equal and converge to the optimal maximal common rate (in kbps): $C(0)=9.53$, $C(9.53)=16.92$, $C(16.92)=24.45$, and $C(24.25)=25.51$. Figure 10 (d) and (e) show the corresponding optimal routes calculated by the two DMCRs, i.e. the two temporary graphs. The width of each edge in the graph is proportional to the amount of allocated data rate. Specifically, Figure 10 demonstrates the optimal routes with regards to LM^*. The saturated cuts of the first temporary graph are $\{3, 4, 7, 15, 16\}$ and $\{8\}$; and the saturated cuts of the second temporary graph are $\{6, 13, 11\}$ and $\{9\}$.

In the whole two DMCR-LMR iterations, only 13 MN packets are transmitted (i.e. the communication overhead of LMD is only about 0.72 packet per node.) and the average communication overhead of the two DMCRs is about 68.6 packets per node. Hence, the total overhead is about 69 control packets per node plus some very simple calculations on each node, which is acceptable in current real WSN deployments. On the other hand, the centralized approach requires every node sends highly reliable local information to a

powerful device and then solve $1 + 18 + 1 + 6 + 1 + 2 + 1 + 1 = 31$ LP problems for 4 MCR-MSR cycles (totally millions of matrix calculations) and then the powerful device transmits the calculated LM rate to each sensor node with high reliability. Obviously, our distributed approach has much more promising potential for practical SP-WSN implementation.

The whole system converges within two DMCR-LMD cycles which is only half of the number of centralized MCR-MSR cycles. The total convergence time is about 120 DMCR steps plus $2 \times [T_0(r), T_n(r)] = 20$ seconds for LMD. If we set the length of a iteration step of DMCR is one second (i.e each node does simple arithmetic calculations and sends a beacon per second), then LM^* can be calculated within about 140 seconds which is less than the duration of a slot (5 minutes). Consequently, we can compute LM^* either at the beginning of a prediction interval or in an additionally allocate slot before a prediction interval in practice, depending on the duration of a slot.

The major complexity of centralized approach is the $O(|\mathcal{V}|^2)$ MSR problems. Since we develop the extremely lightweight LMD for LM rate determination problems, the key complexity source of our distributed approach is DMCR (especially the first one). Consequently, current developed techniques for distributed optimization such as the smooth Lagrangian[Necoara and Suykens 2008] and the event-triggered approach[Wan and Lemmon 2009] can be adopted to significantly reduce (an orders of magnitude) the communication overhead and convergence time of DMCR and therefore the whole system.

Theoretically, our whole system should achieve energy neutral operation. However, since there is no prefect solar power predictor in practice, we run simulations to study the impact of prediction errors on the system performance. We set the predicted solar profile of each node be $\pm 20\%$ different from its real solar power in every slot. Figure 10 (f) shows the number of battery exhausting slot for our distributed LM rate allocation with LPM and Solar Power Average (SPA). In the SPA simulation, each node x uses the average solar power over the whole prediction interval as EC_x^{max}. It is obvious that without considering hardware details, many nodes (especially the saturated nodes) run out of energy for many slots. With the presentence of the prediction error, there are still 3 saturated nodes run out of battery in a slot even when LPM is adopted, which shows that prediction error is an important practical issues to affect the ENO performance. To check whether we can avoid battery exhausting by adjusting parameter settings, we also run two additional simulations with larger initial battery levels and corresponding φ and shorter the prediction interval length respectively. Both simulations achieve ENO for multiple prediction intervals. In addition, dynamically adjust nodes' sensing rates without losing optimality would also an approach to prevent node run out of energy, which will be our future work.

In addition, we also constructed an simulation to evaluate the impact of the reliable wireless channels on the performance of our DMCR-LMD approaches. For each link (x, y), its packet reception ratio $PRR_{x,y}$ was set as a time-varying variable with a random expected value of $90\% \pm 10\%$. Beside unreliable wireless link, all other parameters remain the same. The simulation results show that DMCR-LMD manages to compute the optimal LM rate allocation and corresponding optimal routes, shown in Figure 10 (a) and (e) respectively. However, the communication overhead of our approach in this SP-WSN with unreliable links is 19.2% more than that in SP-WSN with prefect links.

8.3. Sub-optimality Study

This subsection compares our optimal distributed approaches with recent proposed sub-optimal approach DLEX (compute LM^* with a predefined routing tree) and a heuristic approach DLEX-DAG (compute LM^* with predefined multi-path routes) based on simulations.

Since both DLEX and DLEX-DAG are both designed only for the WSN with single sink, we changed sink 1 in Figure 9 into an additional sensor node 18, while the topology and all other parameters remain the same. We construct an arbitrary shortest path tree (SPT).
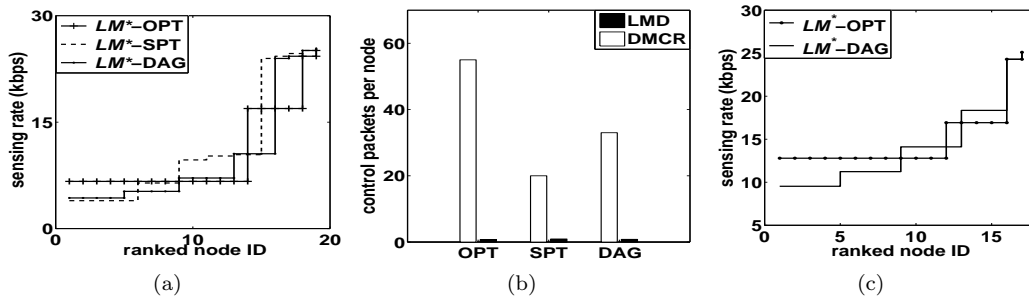
Fig. 11. Simulations of LM rate allocation for original topology, predetermined Shortest Path Tree (SPT) and Directed Acyclic Graph (DAG). (a) LM^* for a SP-WSN with a single sink. (b) Communication overheads of our approach over three topologies. (c) LM^* for a SP-WSN with 3 sinks.

Then a DAG is generated as follows: for every sensor node x, if it has a neighbor y which is closer to the sink than y, then x adds an outgoing link (x, y) with probability 0.5.

Figure 11 (a) shows that the optimal LM^* achieve better fairness (i.e. lexicographically greater) than LM^* of both predetermined SPT and DAG. It is worth noting that LM^*-DAG is the optimal LM rate allocation, which is not lexicographically less than the approach DLEX-DAG, since DLEX-DAG is heuristic for a given DAG. Hence, even we provide the optimal routes (i.e. last temporary graph) with respect to LM^* as the DAG, DLEX-DAG can not guarantee to computer LM^*.

Figure 11 (b) illustrate the communication overhead. It is clear that DMCR overhead for the SPT and DAG is significantly smaller the original topology. The main reasons are two-folds: (1) the number of primary variables ($f_{x,y}, (x, y) \in \mathcal{L}$) of SPT and DAG is much smaller than that of original topology; (2) shortest path tree and DAG provide a good initialization to direct DMCR convergent towards the sub-optimal. For complexity, the overheads of our approach are around 30 one-hop control packets per node and several simple calculations, but DLEX-DAG requires a large number of (could be exponential in $|\mathcal{V}|$) of Depth-First Searches (DFS) and multi-hop control packets.

In fact, for any given SP-WSN topology, the optimal routes with regards to the optimal LM^* is the optimal DAG among all possible DAGs (the number of all possible DAGs is exponential in the number of all links). However, there is no existing scheme that can determine this optimal DAG in arbitrary SP-WSN topology and heterogeneous nodes' energy resource (e.g. solar power and battery) without calculating LM^*. Consequently, any given DAG without calculating LM^* could almost never be the optimal. As an example, we also run a simple simulation over a SP-WSN with 3 sinks, by changing node 17 in Figure 9 into sink 3, while the topology and all other parameters remain the same. Figure 11 (c) shows the optimal LM^* is still much fairer than LM^* with an arbitrarily constructed shortest-path DAG (i.e. every node sends data to the nearest sink with a arbitrary shortest multi-path routing).

8.4. Scalability Study

In this subsection, we study the scalability of our approach based on simulations. We created three randomly deployed SP-WSNs with 50, 75, and 100 nodes, illustrated in Figure 12(a), (c), and (e) respectively. To investigate the adaptiveness of our approach for different number of sinks, we randomly assign 3, 4, and 5 sinks to the three SP-WSNs respectively. Except for the network topology, all parameter settings are the same as that in subsection 7.2. The calculated optimal LM rate allocation is shown in Figure 13 (a), and corresponding optimal routes are shown in Figure 12((b), (d), and (f) respectively. There exist several routing
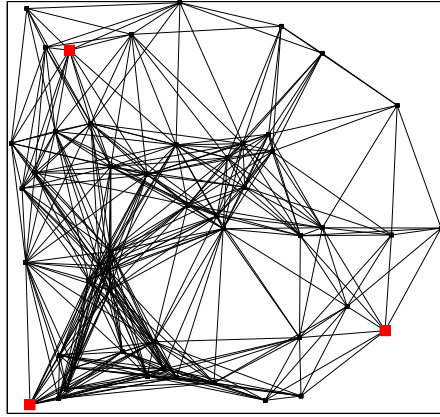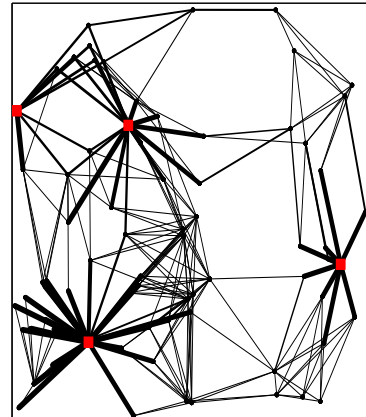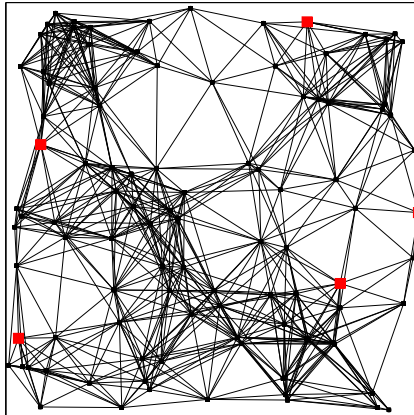
(a) 50-node topology
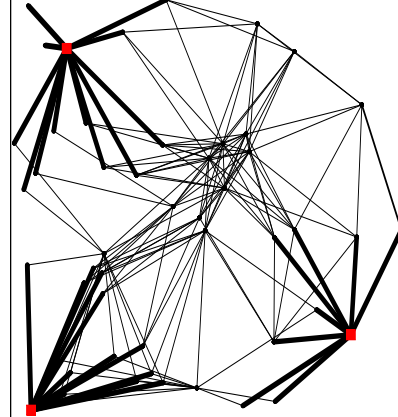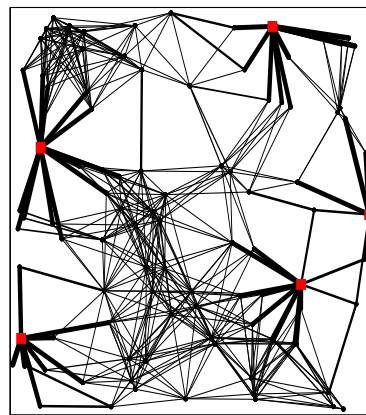
(b) 50-node optimal routing
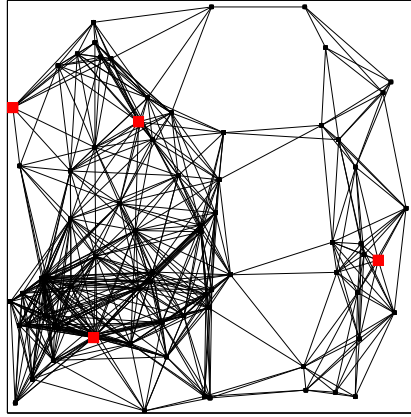
(c) 75-node topology

(d) 75-node optimal routing

(e) 100-node topology

(f) 100-node optimal routing

Fig. 12. Topology and optimal routing for a 50-node SP-WSN with 3 sinks, a 75-node SP-WSN with 4 sinks, and a 100-node SP-WSN with 5 sinks. All sinks are marked as red squares.

loops in the centralized approach for the 75-node and 100-node simulations, while all routes computed by our distributed approach are loop-free.

(a) LM^* for three topologies   (b) total global loop numbers   (c) total control overhead
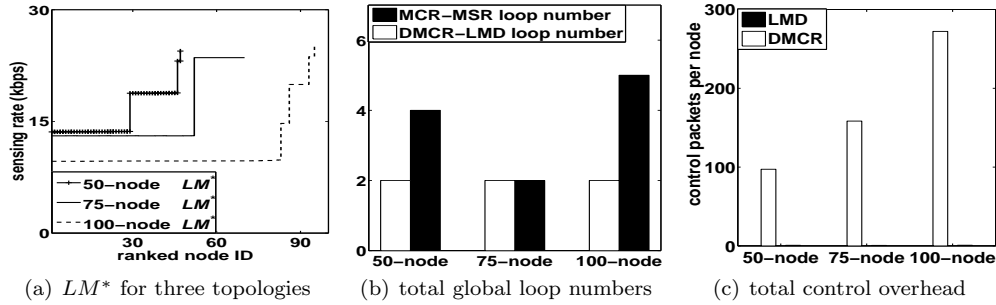
Fig. 13. LM rate, convergence, and control overhead of the three SP-WSNs

Finally, 13 (b) shows that our distributed approach can reduce global loop number by up to 150% , compared with the centralized approach. As a result, the system convergence time and overhead are significantly reduced. From Figure 13 (c) we can see the communication overheads for the LMD is always less than one MN parcket per node, which is independent with the SP-WSN scale and can be ignore. The main source of overhead is still from DMCR, which can be significantly improved by using advanced distributed optimization techniques. Compared with the dramatic complexity of centralized approach (e.g. The 100-node SP-WSN need to solve 124 LP problems to compute LM^*, and collect reliable multi-hop information collection and dissemination), our distributed approach achieves a much more light overhead.

9. RELATED WORK

9.1. Lexicographic Max-min Fairness

Max-min fairness has been considered in various wireless networking schemes [Nace and M.Pioro 2008; Shah-Mansouri et al. 2009; Rangwala et al. 2006; Sridharan and Krishnamachari 2009; Liu et al. 2011]. Approximated max-min fairness has also been considered in many NUM-based schemes [Chiang et al. 2008] by using the aggregated α-fair utility function[Mo and Walrand 2000; Lan et al. 2010]. Many of these approaches are distributed, however, none of them considers the LM rate allocation.

The concepts of Lexicographic ordering and max-min fairness are closely related. [Radunovic and Boudec 2007] presents detailed discuss of max-min fairness, lexicographic ordering, and Pareto optimal. As a specific application of LM fairness, LM rate allocation are studied in capacity-constraint WSNs[Chen et al. 2007], lifetime-restricted WSNs [Hou et al. 2008], and ENO-restricted SP-WSNs[Liu et al. 2011]. All these approaches are based on iterating solving the following two problems: (1) computing the value of each level of LM rate vector, and (2) determine the set of sensor nodes in each level. All of [Chen et al. 2007; Hou et al. 2008; Liu et al. 2011] use a LP (i.e. MCR problem) for the first problem at each LM rate level, which results in a total complexity of $O(N)$ LPs for a N-node WSN. For the second problem at each LM rate level, [Chen et al. 2007; Liu et al. 2011] require to solve $O(N)$ LPs (i.e. MSR problem), while [Hou et al. 2008] adopts a so-called Parametric Analysis (PA). The total complexity of both MSR and PA at all LM rate levels are $O(N^2)$ LPs. Hence, the complexity of all the three approaches are $O(N)+ O(N^2) = O(N^2)$ LPs[5].

---

[5][Radunovic and Boudec 2007] claims that the complexity of max-min programming is $O(N)$ LPs, since the complexity of the second problem are not explicitly considered.

Since our LMD scheme achieves totally $O(1)$ complexity for the second problem at all LM rate levels, the complexity of our DMCR-LMD approach is at most $O(N)$ LPs.

9.2. Energy Harvesting WSNs

There are several power management schemes for individual solar-powered sensor nodes. Stochastic approaches such as [Sharma et al. 2010] may perform poorly in practice, since the complex time-varying dynamics of solar power is quite difficult to model as a piecewise or periodically stationary process. Finite-horizon deterministic schemes [Kansal et al. 2007; Moser et al. 2010; Chen et al. 2011; Gorlatova et al. 2011] is based on the predictability of solar power prediction. None of them considers the impact of time-varying solar power on the network-wide performance.

On the other hand, there are also several network-wide schemes for energy harvesting WSNs, including routing [Eu et al. 2010b], activation [Jaggi et al. 2008], cooperative communication[Li et al. 2011], medium access control [Eu et al. 2010a], rate allocation [Liu et al. 2011] and cross-layer optimization [Liu et al. 2010]. Among these network-wide schemes, only [Liu et al. 2011] and [Liu et al. 2010] integrates power management component to ensure ENO. However, [Liu et al. 2011] does not consider solar prediction errors and assumes an ideal battery model with 100% recharging efficiency and no leakage. SnapIt[Liu et al. 2010] relies on frequent and precise battery level measurement. However it is extremely difficult to meter a slightly changed battery level at runtime in practice [Kansal et al. 2007; Taneja et al. 2008]. In addition, it does not consider recharging inefficiency and battery leakage neither. In contrast, our system adopts a more realistic power model that captures the key properties of solar-battery hardware system, and does not require frequent online battery monitoring. Further more, considering there is no prefect solar power predictor, we adopt multiple prediction intervals and final state constraints to adapt to solar prediction errors, which results in much more reliable performance in real SP-WSNs. For a recent comprehensive survey of energy harvesting WSNs, we refer the reader to [Sudevalayam and Kulkarni 2011].

10. CONCLUSIONS

This paper investigates a multi-objective optimization problem, the Lexicographic Max-min (LM) rate allocation, for solar-powered WSNs with arbitrary topologies. We first develop an efficient power management algorithm that manage to achieve long-term sustainable operations while providing maximum feasible energy budget constraint for each solar-powered sensor node. Then, we present a fully distributed solution to the global LM rate allocation problem, based on iterating a dual-decomposition based algorithm DMCR and a graph-theoretic scheme LMD. To our knowledge, the proposed DMCR-LMD approach is the *first* fully distributed optimal solution to the LM rate allocation problem. We formally prove the optimality, convergence, and efficiency of the whole system. The proposed algorithms are evaluated via experiments on solar-powered MicaZ motes and extensive simulations using real solar energy data and practical power parameter settings. The experiment results verify our theoretical analysis and demonstrates that our approach outperforms the state-of-the-art centralized optimal and distributed heuristic solutions, in terms of complexity, optimality, scalability. Our future work is to extend the our approach to the general LM fair application-defined utilities and design real-time schemes that better adapt to the solar prediction errors.

APPENDIX

PROOF OF PROPOSITION 3.1. According to constraints (3)-(5), it that B_x^{i+1} is a non-increasing in both EC_x and B_x^i. For all $i \in I$, We write

$$B_x^{i+1} = f(EC_x, B_x^i)$$

$$...$$

$$= f(EC_x, f(EC_x, ..., f((EC_x, B_x^1))))$$

Since B_x^1 is fixed, B_x^{i+1} is a monotonic non-increasing function of pure EC_x for all $i \in I$. □

PROOF OF PROPOSITION 3.2. The first condition (12) is to grantee that $\forall i \in I, B_x^{i+1} > 0$. To prove this, we choose a $EC_x \in (E_{min}, h_{min})$ where h_{min} is the minimal h_x^i, $i \in I$, then the battery recharges in every slot $i \in I$. Let $\beta = h_{min} - E_{leak}/\eta \in (E_{min}, h_{min})$, we have $B_x^{i+1} = B_x^i + \eta(h_x^i - EC_x) - E_{leak} \geq B_x^i + \eta(\beta + E_{leak}/\eta - EC_x) - E_{leak} \geq 0 \Leftrightarrow EC_x \leq B_x^i/\eta + \beta$, $\forall i \in I$. Since $B_x^1 \geq 0$, there must exists an optimal solution $EC_x^{max} \in (E_{min}, \beta)$ such that $\forall i \in I, B_x^{i+1} \geq 0$.

The combination of both two conditions (12) and (13) can guarantee $B_x^{L+1} \geq \varphi$. To prove this, we choose a $EC_x \in (E_{min}, h_{min})$ and define

$$\gamma = B_x^1 + \eta \sum_{i \in I}(h_x^i - E_{min}) - LE_{leak} - \varphi$$

$$= B_x^{L+1} - \varphi + \eta LEC_x - \eta LE_{min}$$

$$\geq 0 \tag{38}$$

We set $EC_x \in (E_{min}, h_{min})$ then there are two cases:

Case 1. $\forall i \in I, B_x^i < B_{max}$. In this case, we have $B_x^{L+1} \geq \varphi \Leftrightarrow EC_x \leq \gamma/\eta L + E_{min}$, i.e. there must exists a $EC_x^{max} \in (E_{min}, \min(h_{min}, \gamma/\eta L + E_{min}))$ such that $B_x^{L+1} \geq \varphi$ is satisfied.

Case 2. $\exists i \in I, B_x^i = B_{max}$. Let j be the first slot the battery is fully recharged then for any $i \geq j$, $B_x^{i+1} = B_{max} \geq \varphi$. Therefore, the optimal solution in this case is $EC_x^{max} \in (E_{min}, \beta)$.

□

PROOF OF LEMMA 3.5. The calculation of EC_x' (lines 21 and 25 of the LMP algorithm) assumes that $L_1' = L_1$ and $\forall t \in [j, k], B_x^t(EC_x') \leq B_{max}$. When the energy consumption level drops from EC_x to EC_x', the following three cases would happen.

Case 1. $L_1' = L_1$ and $\forall t \in [j, k], B_x^t(EC_x') \leq B_{max}$. In this case, $B_x^{k+1}(EC_x') = 0$ if $k < L$; or $B_x^{k+1}(EC_x') = \varphi$, otherwise. The reason is that this case is the assumption of the EC_x' calculation.

Case 2. $\exists t \in [j, k], B_x^t(EC_x') > B_{max}$. In this case, $B_x^{k+1}(EC_x') < 0$ if $k < L$; or $B_x^{k+1}(EC_x') < \varphi$, otherwise. The reason is that with EC_x', some energy will be wasted in some slots between j and k due to battery overcharging.

Case 3. $L_1' > L_1$. In this case, $B_x^{k+1}(EC_x') < 0$ if $k < L$; or $B_x^{k+1}(EC_x') < \varphi$, otherwise. From the calculation of EC_x', we have

$$\left. \begin{array}{l} B_x^{k+1}(EC_x') - 0, k < L \\ B_x^{k+1}(EC_x') - \varphi, k = L \end{array} \right\} = (\eta - 1)\left(\sum_{t \in H} h_x^t + EC_x'\Delta L\right)$$

where $H = \{t | (t \in [j, k]) \wedge (EC_x > h_x^t) \wedge (EC_x' < h_x^t)\}$ and $\Delta L = L_1' - L_1$. Consider $\eta < 1, h_x^t > 0, EC_x' > 0$ and $\Delta L > 0$, we have:

$$\left. \begin{array}{l} B_x^{k+1}(EC_x') - 0, k < L \\ B_x^{k+1}(EC_x') - \varphi, k = L \end{array} \right\} < 0$$

If either cases 2 or 3 happens, LPM algorithm will recheck the feasibility of EC_x over slots j to k until case 1 happens, according to Proposition 3.1 and Lemma 3.4. Consider the battery level B_x^{k+1} in case 1 and Propositions 3.1, the last updated EC_x is the maximum feasible energy consumption for all slots before k. □

REFERENCES

micaz mote,. http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf.

Solar data from aberdeen,. Aberdeen, USA, May 2010: http://solardat.uoregon.edu/SolarData.html.

BACCOUR, N., KOUBAA, A., MOTTOLA, L., ZUNIGA, M. A., YOUSSEF, H., BOANO, C. A., AND ALVES, M. 2012. Radio link quality estimation in wireless sensor networks: a survey. *ACM Transactions on Sensor Networks (TOSN) 8,* 4, 34.

BERGONZINI, C., BRUNELLI, D., AND BENINI, L. 2010. Comparison of energy intake prediction algorithms for systems powered by photovoltaic harvesters. *Microelectronics Journal, 41,* 11, 766–777.

BERTSEKAS, D. AND GALAGER, R. 1992. *Data Networks Second ed.* Prentice Hall.

BOYD, S. AND VANDENBERGHE, L. 2004. *Convex Optimization.* Cambridge University Press,, Reading, MA.

CHALLEN, G., WATERMAN, J., AND WELSH, M. 2010. Idea: integrated distributed energy awareness for wireless sensor networks. In *ACM MobiSys 2010,.* 35–48.

CHEN, S., FANG, Y., AND XIA, Y. 2007. Lexicographic maxmin fairness for data collection in wireless sensor networks,. *IEEE Trans. Mobi. Comput. 6,* 7, 62–76.

CHEN, S., SINHA, P., N.B.SHROFF, AND C.JOO. 2011. Finite-horizon energy allocation and routing scheme in rechargeable sensor networks. In *IEEE INFOCOM 2011.* 2273 – 2281.

CHIANG, M., LOW, S., CALDERBANK, A., AND DOYLE, J. 2008. Layering as optimization decomposition: A mathematical theory of network architectures. *Proc. IEEE, 95,* 1, 255–312.

COX, D. R. 1961. Prediction by exponentially weighted moving averages and related methods. *Journal of the Royal Statistical Society 23,* 2, 414–422.

DE COUTO, D. S., AGUAYO, D., CHAMBERS, B. A., AND MORRIS, R. 2003. Performance of multihop wireless networks: Shortest path is not enough. *ACM SIGCOMM Computer Communication Review 33,* 1, 83–88.

EU, Z., TAN, H.-P., AND SEAH, W. 2010a. Design and performance analysis of mac schemes for wireless sensor networks powered by ambient energy harvesting. *Ad Hoc Networks, 9,* 3, 1326–1336.

EU, Z. A., TAN, H.-P., AND SEAH, W. K. G. 2010b. Opportunistic routing in wireless sensor networks powered by ambient energy harvesting. *Computer Networks, 54,* 17, 2943–2966.

GNAWALI, O., FONSECA, R., JAMIESON, K., MOSS, D., AND LEVIS, P. 2009. Collection tree protocol. In *Proc. ACM SenSys.*

GORLATOVA, M., WALLWATER, A., AND ZUSSMAN, G. 2011. Networking low-power energy harvesting devices: measurements and algorithms. In *IEEE INFOCOM 2011.* 1602 – 1610.

GRANT, M. AND BOYD, S. Cvx: Matlab software for disciplined convex programming. http://cvxr.com/cvx/.

HOU, Y., SHI, Y., AND SHERALI, H. 2008. Rata allocation and network lifetime problems for wireless sensor networks. *IEEE/ACM Trans. Netw., 16,* 2, 1–14.

HUANG, P. 2007. Dual approach to multi-path maxmin rate allocation. Ph.D. thesis, University of California, Irvine.

JAGGI, N., KAR, K., AND KRISHNAMURTHY, A. 2008. Near-optimal activation policies in rechargeable sensor networks under spatial correlations. *ACM Trans. Sens. Netw. 4,* 3.

KANSAL, A., HSU, J., ZAHEDI, S., AND SRIVASTAVA, M. 2007. Power management in energy harvesting sensor networks. *ACM Trans. Embedded Comput. Sys., 6,* 4.

LAN, T., KAO, D., CHIANG, M., AND SABHARWAL, A. 2010. An axiomatic theory of fairness in network resource allocation. 1–9.

LI, H., JAGGI, N., AND SIKDAR, B. 2011. Relay scheduling for cooperative communications in sensor networks with energy harvesting. *IEEE Trans. Wireless Commun. 10,* 9, 2918–2928.

LIU, E., SINHA, P., AND KOKSAL, C. 2010. Joint energy management and resource allocation in rechargeable sensor networks. In *IEEE INFOCOM 2010,.* 1–9.

LIU, R., FAN, K., ZHENG, Z., AND SINHA, P. 2011. Perpetual and fair data collection for environmental energy harvesting sensor networks. *IEEE/ACM Trans. Netw., 19,* 4, 947– 960.

MO, J. AND WALRAND, J. 2000. Fair end-to-end window based congestion control,. *IEEE/ACM trans. Netw. 8,* 5, 556–567.

MOELLER, S., SRIDHARAN, A., KRISHNAMACHARI, B., AND GNAWALI, O. 2010. Routing without routes: the backpressure collection protocol. In *Proc. ACM/IEEE IPSN*.

MOSER, C., THIELE, L., BRUNELLI, D., AND BENINI, L. 2010. Adaptive power management for environmentally powered systems. *IEEE Trans. Comput., 59,* 4.

NACE, D. AND M.PIORO. 2008. Max-min fairness and its applications to routing and load-balancing in communication networks: A tutorial. *IEEE Commun. Surveys Tuts., 10,* 4, 5–17.

NECOARA, I. AND SUYKENS, J. A. K. 2008. Application of a smoothing technique to decomposition in convex optimization. *IEEE Trans. Autom. Control, 53,* 11, 2674–2679.

NIYATO, D., HOSSAIN, E., AND FALLAHI, A. 2007. Sleep and wakeup strategies in solar-powered wireless sensor/mesh networks: Performance analysis and optimization. *IEEE Trans. Mobi. Comput., 6,* 2, 221–23.

RADUNOVIC, B. AND BOUDEC, J. 2007. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Trans. Netw., 15,* 5, 1073–1083.

RANGWALA, S., R.GUMMADI, GOVINDAN, R., AND PSOUNIS, K. 2006. Interference-aware fair rate control in wireless sensor networks. In *ACM SIGCOMM 2006*. 63–74.

SALLES, R. AND BARRIA, J. 2008. Lexicographic maximin optimisation for fair bandwidth allocation in computer networks. *European Journal of Operational Research, 182,* 2, 778–794.

SHAH-MANSOURI, V., MOHSENIAN-RAD, A.-H., AND WONG, V. 2009. Lexicographically optimal routing for wireless sensor networks with multiple sinks. *IEEE Trans. Veh. Technol., 58,* 3, 1490–1500.

SHARMA, V., MUKHERJI, U., JOSEPH, V., AND GUPTA, S. 2009. Challenge: Ultra-low-power energy-harvesting active networked tags (enhants). In *ACM MOBICOM 2009,*. 253–260.

SHARMA, V., MUKHERJI, U., JOSEPH, V., AND GUPTA, S. 2010. Optimal energy management policies for energy harvesting sensor nodes. *IEEE Trans. Wireless Commun., 9,* 4, 1326–1336.

SRIDHARAN, A. AND KRISHNAMACHARI, B. 2009. Explicit and precise rate control for wireless sensor networks. In *ACM SenSys 2009,*. 29–42.

SUDEVALAYAM, S. AND KULKARNI, P. 2011. Energy harvesting sensor nodes: Survey and implications. *IEEE Commun. Surveys Tuts., 13,* 3, 443–461.

TANEJA, J., JEONG, J., AND CULLER, D. 2008. Design, modeling, and capacity planning for micro-solar power sensor networks. In *IPSN 2008,*. 407–418.

WAN, P. AND LEMMON, M. D. 2009. Event-triggered distributed optimization in sensor networks. In *IPSN 2009,*. 49–60.

XIAO, L., JOHANSSON, M., AND BOYD, S. 2004. Simultaneous routing and resource allocation via dual decomposition. *IEEE Trans. Commun., 52,* 7, 1146–1144.