Distributed Computation of Fair Packet Rates in Energy Harvesting Wireless Sensor Networks

Fayçal Ait Aoudia, Matthieu Gautier, Olivier Berder

Distributed Computation of Fair Packet Rates in Energy Harvesting Wireless Sensor Networks

Fayçal Ait Aoudia, Matthieu Gautier, Olivier Berder IRISA, University of Rennes 1
Email: {faycal.ait-aoudia, matthieu.gautier, olivier.berder}@irisa.fr

***Abstract*—Energy harvesting is a key technology to enable long-term wireless sensor network applications. In the case of multi-hop networks, each node both performs measurements to produce data to be sent to a sink, and relays data packets from other nodes. In this letter, we propose a distributed algorithm for computation of fair packet rates for multi-hop energy harvesting wireless sensor networks. The packet rate computation problem is formulated as a convex optimization problem, and using the fast alternating direction method of multipliers, the original problem is decomposed into smaller subproblems that can be solved in parallel. Simulations using real indoor light energy traces show that the algorithm computes high accuracy solutions, even with a low median number of iterations (10 or less). By setting the stop criteria parameter, a compromise can be set between the accuracy of the solution and the number of iterations required.**

I. Introduction

Wireless Sensor Networks (WSNs) are made of several autonomous sensor nodes that monitor an environment and gather the data to a main location. A promising technology to extend the lifetime of WSNs is to allow each node to harvest energy directly from its environment. As the energy sources are typically dynamic and uncontrolled, on-line energy management is required to avoid power failures while maximizing quality of service. Several Energy Managers (EMs) were proposed in the recent years [1]. However, most of them were designed for point-to-point communication, and cannot be easily extended to multi-hop networks.

In multi-hop networks, in addition to performing measurements and sending the so-obtained data to the sink, each node is also a relay that forwards packets from other nodes. Therefore, the energy consumed by each node is shared between packet generation, which consists of sensing to produce new data and send the so-obtained data, and packet relaying, which consists of forwarding packets generated by other nodes. An important consideration in multi-hop WSNs is the fairness of the Packet Generation Rates (PGRs) of the nodes given the amount of energy harvested by each node. Several approaches have been proposed to achieve efficient energy management in multi-hop Energy Harvesting WSNs (EH-WSNs). *Diamantoulakis et al.* studied in [2] proportional fairness maximization in the context of energy harvesting from Radio Frequency (RF) signals, which also transfer information. However, the challenges addressed in RF harvesting are different from the ones addressed when nodes are powered by uncontrollable natural sources. In [3], *Yang et al.* proposed a cross-layer energy management scheme for EH-WSNs, but the proposed solution targets only solar powered WSNs.

The main contribution of this letter is the design of a distributed algorithm for setting the PGRs of sensor nodes forming a multi-hop energy EH-WSN. Routing is not considered, and is assumed to be either imposed by the network topology, or established by a routing algorithm. The nodes are supposed to be organized in a routing tree, where each node has a one-hop successor to which it forwards packets. No assumption is made on the energy source type. The problem of network-scale energy management over a finite time window was formulated as a convex optimization problem, with logarithmic utility function both to maximize the PGR and to achieve proportional fairness. The problem was then reformulated such that, using a fast version of the Alternating Direction Method of Multipliers (ADMM) [4], the problem can be decomposed into smaller subproblems that can be solved in parallel. ADMM method was chosen as it is well-suited to distributed convex optimization [5]. To the best of our knowledge, the closest work to ours is [6], in which the authors studied the effect of deterministic delays and quantization errors on the convergence of decentralized optimization in EH-WSNs. However, the authors did not address the problem of energy management. The proposed approach was validated by simulations of a network of 15 nodes powered by real measurements of indoor light.

II. Distributed and Fair Optimization

Time is divided into equal length time slots of duration T, and the EM is executed at the end of every time slot. An energy predictor (e.g. [7]) provides predictions of the harvested energy over a window of K time slots and the number of nodes forming the network is denoted by N. We define $\mathcal{K} \coloneqq \{1, \dots, K\}$ and $\mathcal{N} \coloneqq \{1, \dots, N\}$ for convenience. The current time slot is denoted by t, and the predicted harvested energy for the time slot $t+k$ is denoted by $h[n,k],\ n \in \mathcal{N},\ k \in \mathcal{K}$. Each node embeds an energy buffer of finite capacity denoted by $B[n]$. The energy failure threshold, *i.e.* the minimum energy level required by the node n to operate, is denoted by $b[n]$. At each execution of the EM, each node n measures the residual energy denoted by $e[n]$. The predicted available energy at slot $t+k$ for the node n is defined by:

$$a[n,k] \coloneqq e[n] + \sum_{i=1}^{k} h[n,i],\ n \in \mathcal{N},\ k \in \mathcal{K}. \tag{1}$$

The energy cost for the node n of a packet generation is denoted by $C_L[n]$, while the energy cost of relaying a packet is denoted by $C_R[n]$. For each node n, the sets $A(n)$ and $S(n)$ are respectively defined as the set of all nodes for

which n is a relay, and the set of all nodes that serve as a relay for n. Formally, $A(n) = \{m \in \mathcal{N} \mid \exists m \rightsquigarrow n\}$ and $S(n) = \{m \in \mathcal{N} \mid \exists n \rightsquigarrow m\}$, where $m \rightsquigarrow n$ designates a path from m to n in the routing tree.

Let $f(n, k, \mathbf{x}, \mathbf{w})$ be the prediction of the residual energy:

$$f(n,k,\mathbf{x},\mathbf{w}) \coloneqq\ a[n,k] - C_L[n]\sum_{i=1}^{k} x[n,i] - C_R[n]\sum_{i=1}^{k}\sum_{m\in A(n)} x[m,i] - \sum_{i=1}^{k} w[n,i], \quad (2)$$

where $\mathbf{x}$ is the PGR vector and $\mathbf{w}$ is the energy waste vector. $x[n,k]$ corresponds to the PGR of the node n at the time slot $t+k$, while $w[n,k]$ corresponds to the energy wasted by the node n at the time slot $t+k$, *i.e.* the harvested energy that was not consumed by the node, and could not be stored as the energy buffer was full. The energy management problem is formulated as follows:

$$\begin{aligned} \underset{\mathbf{x},\mathbf{w}}{\text{minimize}}\ & f_0(\mathbf{x}) \coloneqq -\sum_{n=1}^{N}\sum_{k=1}^{K}\log x[n,k] \\ \text{subject to: } & w[n,k] \geqslant 0,\ n\in\mathcal{N},\ k\in\mathcal{K} \\ & f(n,k,\mathbf{x},\mathbf{w}) \leqslant B[n],\ n\in\mathcal{N},\ k\in\mathcal{K} \\ & f(n,k,\mathbf{x},\mathbf{w}) \geqslant b[n],\ n\in\mathcal{N},\ k\in\mathcal{K} \end{aligned} \quad (\mathrm{P}_1)$$

where the first constraint requires positive wasted energy, the second constraint represents the energy buffer capacity and the last constraint requires no power failure. (P_1) is convex, and the logarithmic utility function f_0 leads to proportional fairness [8] with regard to nodes and time, *i.e.* if $\mathbf{x}$ is an optimal solution of (P_1), then for any feasible solution $\mathbf{x}'$ of (P_1) we have:

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\frac{x'[n,k]-x[n,k]}{x[n,k]} \leqslant 0, \quad (3)$$

which states that any change in the solution must have a negative average change.

To decompose (P_1), each node n keeps a local copy of the PGR of its predecessors, denoted by $\mathbf{c_n}$, and (P_1) is reformulated as follows:

$$\begin{aligned} \underset{\mathbf{x},\mathbf{w},\{\mathbf{c_n}\}}{\text{minimize}}\ & f_0(\mathbf{x}) \\ \text{subject to: } & (\mathbf{x},\mathbf{w},\{\mathbf{c_n}\}) \in \mathcal{C} \\ & c_n[m,k] = x[m,k],\ n\in\mathcal{N},\ k\in\mathcal{K},\ m\in A(n) \end{aligned} \quad (\mathrm{P}_2)$$

where $\{\mathbf{c_n}\}$ denotes the set of vectors $\{\mathbf{c_1},\dots,\mathbf{c_N}\}$, $c_n[m,k]$ is the local copy of the PGR of the node m at the time slot $t+k$ stored by the node n for any $m \in A(n)$, g is defined by:

$$g(n,k,\mathbf{x},\mathbf{w},\mathbf{c_n}) \coloneqq\ a[n,k] - C_L[n]\sum_{i=1}^{k} x[n,i] - C_R[n]\sum_{i=1}^{k}\sum_{m\in A(n)} c_n[m,i] - \sum_{i=1}^{k} w[n,i]. \quad (4)$$

and $\mathcal{C}$ is the convex set of feasible solutions of defined by:

$$\mathcal{C} \coloneqq \Big\{(\mathbf{x},\mathbf{w},\{\mathbf{c_n}\}) \mid n\in\mathcal{N},\ k\in\mathcal{K},\ w[n,k]\geqslant 0,\ g(n,k,\mathbf{x},\mathbf{w},\mathbf{c_n}) \leqslant B[n],\ g(n,k,\mathbf{x},\mathbf{w},\mathbf{c_n}) \geqslant b[n]\Big\}. \quad (5)$$

The indicator function of $\mathcal{C}$ denoted by $\mathcal{I}_{\mathcal{C}}$ is defined by:

$$\mathcal{I}_{\mathcal{C}}(\mathbf{x},\mathbf{w},\{\mathbf{c_n}\}) \coloneqq \begin{cases} 0, & \text{if } (\mathbf{x},\mathbf{w},\{\mathbf{c_n}\}) \in \mathcal{C} \\ \infty, & \text{otherwise} \end{cases} \quad (6)$$

In order to make (P_2) suitable for the ADMM, the variable $\mathbf{x}$ is replicated by a variable $\mathbf{r}$, and (P_2) is equivalent to:

$$\begin{aligned} \underset{\mathbf{x},\mathbf{w},\mathbf{r},\{\mathbf{c_n}\}}{\text{minimize}}\ & f_0(\mathbf{x}) + \mathcal{I}_{\mathcal{C}}(\mathbf{r},\mathbf{w},\{\mathbf{c_n}\}) \\ \text{subject to: } & x[m,k] = c_n[m,k],\ n\in\mathcal{N},\ k\in\mathcal{K},\ m\in A(n) \\ & x[n,k] = r[n,k],\ n\in\mathcal{N},\ k\in\mathcal{K} \end{aligned} \quad (\mathrm{P}_3)$$

The augmented Lagrangian of (P_3) in the scaled form is:

$$L_\rho(\mathbf{x},\mathbf{r},\mathbf{w},\{\mathbf{c_n}\},\{\mathbf{u_n}\},\mathbf{v}) = f_0(\mathbf{x}) + \mathcal{I}_{\mathcal{C}}(\mathbf{r},\mathbf{w},\{\mathbf{c_n}\}) + \frac{\rho}{2}\sum_{n=1}^{N}\Big(\|\mathbf{x[n]} - \mathbf{r[n]} + \mathbf{v[n]}\|_2^2 + \sum_{m\in A(n)}\|\mathbf{x[m]} - \mathbf{c_n[m]} + \mathbf{u_n[m]}\|_2^2\Big), \quad (7)$$

where $\rho > 0$ is the penalty parameter, and $\mathbf{v},\ \mathbf{u_1},\dots,\mathbf{u_N}$ are the scaled dual variables. For a vector $\mathbf{y}$, the notation $\mathbf{y[n]}$ denotes the sub-vector $(y[n,1],\dots,y[n,K])^\intercal$.

The proposed iterative algorithm, shown in Algorithm 1, is based on the fast ADMM [4] and is executed at the end of each time slot by each node. The first step of an iteration is to solve:

$$\mathbf{x^i} = \arg\min_{\mathbf{x}} L_\rho(\mathbf{x},\hat{\mathbf{r}}^\mathbf{i},\hat{\mathbf{w}}^\mathbf{i},\{\hat{\mathbf{c}}_\mathbf{n}^\mathbf{i}\},\{\hat{\mathbf{u}}_\mathbf{n}^\mathbf{i}\},\hat{\mathbf{v}}^\mathbf{i}), \quad (8)$$

where $\mathbf{y^i}$ denotes the value of the vector $\mathbf{y}$ at the i^{th} iteration, and $\hat{\mathbf{y}}$ is a variable associated to $\mathbf{y}$ required by the acceleration step of the fast ADMM. If it is assumed that the vector $(\hat{\mathbf{r}},\hat{\mathbf{w}},\{\hat{\mathbf{c}}_\mathbf{n}\}) \in \mathcal{C}$, then (8) can be written as follows:

$$\mathbf{x^i} = \arg\min_{\mathbf{x}}\left\{f_0(\mathbf{x}) + \frac{\rho}{2}\sum_{n=1}^{N}\Big(\big\|\mathbf{x[n]} - \hat{\mathbf{r}}^\mathbf{i}\mathbf{[n]} + \hat{\mathbf{v}}^\mathbf{i}\mathbf{[n]}\big\|_2^2 + \sum_{m\in S(n)}\big\|\mathbf{x[n]} - \hat{\mathbf{c}}_\mathbf{m}^\mathbf{i}\mathbf{[n]} + \hat{\mathbf{u}}_\mathbf{m}^\mathbf{i}\mathbf{[n]}\big\|_2^2\Big)\right\}, \quad (9)$$

by noticing that:

$$\sum_{n=1}^{N}\sum_{m\in A(n)}\|\mathbf{x[m]} - \hat{\mathbf{c}}_\mathbf{n}\mathbf{[m]} + \hat{\mathbf{u}}_\mathbf{n}\mathbf{[m]}\|_2^2 = \sum_{n=1}^{N}\sum_{m\in S(n)}\|\mathbf{x[n]} - \hat{\mathbf{c}}_\mathbf{m}\mathbf{[n]} + \hat{\mathbf{u}}_\mathbf{m}\mathbf{[n]}\|_2^2. \quad (10)$$

To solve this problem, each node computes:

$$\mathbf{x^i[n]} = \underset{\mathbf{x[n]}}{\arg\min} \left\{ \sum_{k=1}^{K} \log x[n,k] + \frac{\rho}{2} \sum_{m \in S(n)} \left\| \mathbf{x[n]} - \mathbf{\hat{c}^i_m[n]} + \mathbf{\hat{u}^i_m[n]} \right\|_2^2 + \frac{\rho}{2} \left\| \mathbf{x[n]} - \mathbf{\hat{r}^i[n]} + \mathbf{\hat{v}^i[n]} \right\|_2^2 \right\}. \quad (11)$$

To solve (11) (line 6 of Algorithm 1), each node n needs the values of $\mathbf{c_m}$ and $\mathbf{u_m}$ of all its successors m (lines 4–5), and therefore each node that is not a leaf of the routing tree sends its values of $\mathbf{c_n}$ and $\mathbf{u_n}$ to its predecessors (lines 2–3). This is done before solving the local problem, to avoid stopping the predecessors from starting solving (11).

The second step of an iteration is to solve:

$$\begin{bmatrix} \mathbf{r^i} \\ \mathbf{w^i} \\ \{\mathbf{c^i_n}\} \end{bmatrix} = \underset{\mathbf{r},\mathbf{w},\{\mathbf{c_n}\}}{\arg\min} L_\rho(\mathbf{x^i}, \mathbf{r}, \mathbf{w}, \{\mathbf{c_n}\}, \{\mathbf{\hat{u}^i_n}\}, \mathbf{\hat{v}^i}), \quad (12)$$

which can be done if each node computes:

$$\underset{\mathbf{r[n]},\mathbf{w[n]},\mathbf{c_n}}{\text{minimize}} \left\{ \left\| \mathbf{x^i[n]} - \mathbf{r[n]} + \mathbf{\hat{v}^i[n]} \right\|_2^2 + \sum_{m \in A(n)} \left\| \mathbf{x^i[m]} - \mathbf{c_n[m]} + \mathbf{\hat{u}^i_n[m]} \right\|_2^2 \right\} \quad (13)$$

$$\text{subject to: } w[n,k] \geqslant 0, \ k \in \mathcal{K}$$
$$g(n,k,\mathbf{r[n]},\mathbf{w[n]},\mathbf{c_n}) \leqslant B[n], \ k \in \mathcal{K}$$
$$g(n,k,\mathbf{r[n]},\mathbf{w[n]},\mathbf{c_n}) \geqslant b[n], \ k \in \mathcal{K}$$

The constraints of (13) guarantee that the global solution of this problem $(\mathbf{r^i}, \mathbf{w^i}, \{\mathbf{c^i_n}\}) \in \mathcal{C}$, and therefore $\mathcal{I}_\mathcal{C}(\mathbf{r^i}, \mathbf{w^i}, \{\mathbf{c^i_n}\}) = 0$. Solving (13) (line 11) requires each node n to have the newly computed PGR values $\mathbf{x[m]}$ of its predecessors (lines 9–10), and therefore each node sends its value of $\mathbf{x[n]}$ (lines 7–8) before starting solving (13). The third step of an iteration is to update the scaled dual variables (lines 12–13).

The residual denoted by C^i measures how far the current iteration is from the optimal solution, and is defined by $C^i = \sum_{n=1}^{N} C_n^i$, where C_n^i is the local residual of the node n defined by:

$$C_n^i = \frac{1}{\rho} \left(\left\| \mathbf{u^i_n} - \mathbf{\hat{u}^i_n} \right\|_2^2 + \left\| \mathbf{v^i[n]} - \mathbf{\hat{v}^i[n]} \right\|_2^2 \right) + \rho \left(\left\| \mathbf{r^i[n]} - \mathbf{\hat{r}^i[n]} \right\|_2^2 + \left\| \mathbf{w^i[n]} - \mathbf{\hat{w}^i[n]} \right\|_2^2 + \left\| \mathbf{c^i_n} - \mathbf{\hat{c}^i_n} \right\|_2^2 \right). \quad (14)$$

It is proved that $\lim_{i \to \infty} C^i = 0$ [4]. At each iteration, each node computes its local residual (line 14) and sends it to the root (line 15). The acceleration steps of fast ADMM are then performed (lines 16–18). The root computes the global residual by gathering all the local residuals (line 20), and performs the stop criteria $C^i < \epsilon$, where $\epsilon > 0$ (line 21–23). Once the stop criteria is passed, each node n sets its PGR to the value of $x[n,0]$ just calculated (line 28). Because (P_3) is weakly convex, a "restart" rule is used to enforce stability (lines 24–27). If the most recent ADMM step has not decreased the residual by a factor of at least $\eta \in (0,1)$, then the most recent iteration is thrown up, and the algorithm is "restarted". The function RESTART is used to perform the "restart" operation. It is called synchronously by the root (line 27), or at the reception of a restart instruction by the other nodes.

Algorithm 1 Executed by the node n at the end of each slot.

Init: $\alpha^1 = 1, \ \mathbf{\hat{r}^1} = \mathbf{r^0}, \ \mathbf{\hat{w}^1} = \mathbf{w^0}, \ \mathbf{\hat{c}^1_n} = \mathbf{c^0_n}, \ \mathbf{\hat{u}^1_n} = \mathbf{u^0_n}, \ \mathbf{\hat{v}^1} = \mathbf{v^0}$

1: **for** $i = 1, \ldots$ **do**
2: **if** $A(n) \neq \varnothing$ **then**
3: Send $\mathbf{\hat{c}^i_n}$ and $\mathbf{\hat{u}^i_n}$ to all predecessors
4: **if** $S(n) \neq \varnothing$ **then**
5: Wait for all $\mathbf{\hat{c}^i_m}$ and $\mathbf{\hat{u}^i_m}$, $m \in S(n)$
6: Compute $\mathbf{x^i[n]}$ by solving (11)
7: **if** $S(n) \neq \varnothing$ **then**
8: Send $\mathbf{x^i[n]}$ computed at previous step to all successors
9: **if** $A(n) \neq \varnothing$ **then**
10: Wait for all $\mathbf{x^i[m]}$, $m \in A(n)$
11: Compute $\mathbf{r^i[n]}$, $\mathbf{w^i[n]}$ and $\mathbf{c^i_n}$ by solving (13)
12: $\mathbf{u^i_n[m]} \leftarrow \mathbf{\hat{u}^i_n[m]} + \mathbf{x^i[m]} - \mathbf{c^i_n[m]}, \ m \in A(n)$
13: $\mathbf{v^i[n]} \leftarrow \mathbf{\hat{v}^i[n]} + \mathbf{x^i[n]} - \mathbf{r^i[n]}$
14: Compute the local residual C_n^i according to (14)
15: Send the local residual C_n^i to the root
16: $\alpha^{i+1} \leftarrow \frac{1+\sqrt{1+(\alpha^i)^2}}{2}$
17: **for** $\mathbf{y} \in \{\mathbf{r[n]}, \mathbf{w[n]}, \mathbf{c_n}, \mathbf{u_n}, \mathbf{v[n]}\}$ **do**
18: $\mathbf{\hat{y}^{i+1}} \leftarrow \mathbf{y^i} + \frac{\alpha^i - 1}{\alpha^{i+1}}(\mathbf{y^i} - \mathbf{y^{i-1}})$
19: **if** Is root **then**
20: $C^i = \sum_{n=1}^{N} C_n^i$
21: **if** $C^i < \epsilon$ **then**
22: Broadcast stop instruction
23: **break**
24: **else if** $C^i \geqslant \eta C^{i-1}$ **then**
25: $C^i = \frac{C^{i-1}}{\rho}$
26: Broadcast restart instruction for slot i
27: RESTART(i)
28: Set PGR to $x[n,0]$
29: ▷ Executed if a restart instruction is received:
30: **function** RESTART(i)
31: $\alpha^{i+1} \leftarrow 1$
32: **for** $\mathbf{y} \in \{\mathbf{r[n]}, \mathbf{w[n]}, \mathbf{c_n}, \mathbf{u_n}, \mathbf{v[n]}\}$ **do**
33: $\mathbf{\hat{y}^{i+1}} \leftarrow \mathbf{y^{i-1}}$
34: Start iteration $i+1$

III. PERFORMANCE EVALUATION

A network made of 15 PowWow [9] nodes organized in a binary tree was simulated, the root node having as a one-hop successor the sink. To simulate the harvested energy,

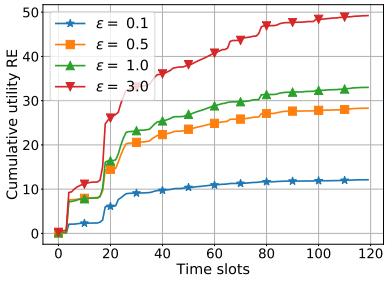

Fig. 1: Utility RE.

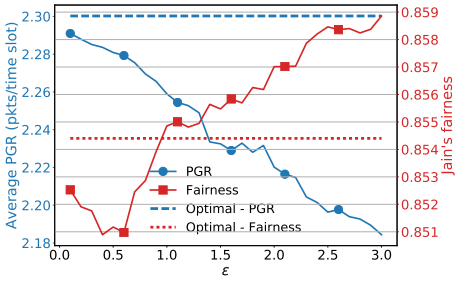

Fig. 2: Average PGR and fairness.

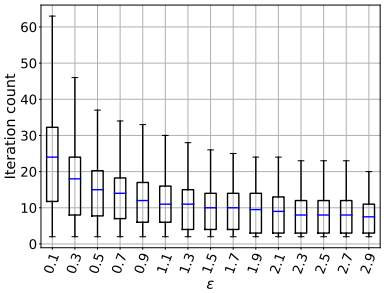

Fig. 3: Iteration count.

indoor light energy traces from [10] were used. These traces correspond to real measurements, and each node was powered with a different trace. All the nodes were equipped with a 0.9F capacitance, with a maximum voltage of 5.2V and a minimum voltage of 2.8 V, and therefore $B[n] = 12.168\,\text{J},\ n \in \mathcal{N}$ and $b[n] = 3.528\,\text{J},\ n \in \mathcal{N}$. Moreover, the energy costs C_R and C_L were measured to be 25 mJ and 15 mJ respectively [9], and these values were used for all the nodes.

The EM was executed every $T = 2$ hours, and the prediction window was $K = 12$, corresponding to 24 hours. The predictor from [7] was used, and the simulated time was 10 days. η was set to the value of 0.999 [4], while ρ was set to the value of 1 as it was found out experimentally to lead to high convergence speed. Two performance metrics were considered: the average PGR, and fairness which was measured using Jain's fairness index defined as follows:

$$J(\mathbf{x}, t) = \frac{\left(\sum_{n=1}^{N} x[n,t]\right)^2}{N \sum_{n=1}^{N} x[n,t]^2}. \tag{15}$$

The Jain's fairness index ranges from $\frac{1}{N}$ to 1, this latter value corresponding to all the nodes having the same PGR. The proposed algorithm was evaluated for values of ϵ in the range $[0.1, 3.0]$. Moreover, (P_1) was also solved by a regular solver and the so-obtained optimal solution serves as a reference for comparison. For all the simulation runs, no power failure was observed. Fig. 1 presents the cumulative Relative Error (RE) between the utility value obtained by the proposed algorithm and the optimal utility for each time step. As expected, the lower is ϵ, the lower is the RE. It can be seen in Fig. 2, that the optimal Jain's fairness is not 1. Indeed, nodes that harvest more energy will have a higher PGR as long as it is fair to the other nodes giving the amount of energy that they harvest. The achieved Jain's fairness is however close to 1, which indicates that the nodes have similar PGRs. In less accurate solutions obtained when choosing high values of ϵ, the average PGR is up to 5% lower than the optimal average PGR, while the Jain's fairness is not strongly impacted, as it stays within a 0.5% range of the optimal Jain's fairness, as shown in Fig. 2. It can be seen that reducing ϵ leads to more accurate solution, as the average average PGR gets closer to the optimal average PGR.

Increasing ϵ leads to less accurate solutions, but significantly decreases the overhead of the EM. Indeed, on some systems, the overhead incurred by the EM is not negligible. As shown in Fig. 3, increasing ϵ significantly reduces the average number of iterations required by the proposed algorithm. When values of ϵ higher than 1.5 are chosen, the median number of iterations required by the algorithm is 10 or less.

IV. Conclusion

In this letter, we proposed a distributed algorithm for computation of fair packet rates for multi-hop energy harvesting WSNs. Simulation results using real indoor light energy traces showed that the proposed scheme achieves high fairness. By adjusting the stop criteria parameters, it is possible to set a compromise between the accuracy of the calculated solution, and the computational overhead incurred by the algorithm.

References

[1] A. A. Babayo, M. H. Anisi, and I. Ali, "A Review on energy management schemes in energy harvesting wireless sensor networks," *Renewable and Sustainable Energy Reviews*, vol. 76, pp. 1176 – 1184, 2017.

[2] P. D. Diamantoulakis and G. K. Karagiannidis, "Maximizing Proportional Fairness in Wireless Powered Communications," *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 202–205, April 2017.

[3] S. Yang, X. Yang, J. A. McCann, T. Zhang, G. Liu, and Z. Liu, "Distributed Networking in Autonomic Solar Powered Wireless Sensor Networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 750–761, December 2013.

[4] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast Alternating Direction Optimization Methods," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1588–1623, 2014.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, 2011.

[6] N. J. Roseveare, S. M. S. Alam, and B. Natarajan, "Bounds on decentralized concave optimization in energy harvesting wireless sensor networks," in *Annual IEEE Systems Conference (SysCon)*, April 2016.

[7] A. Cammarano, C. Petrioli, and D. Spenza, "Online Energy Harvesting Prediction in Environmentally Powered Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 16, no. 17, pp. 6793–6804, September 2016.

[8] W. H. Wang, M. Palaniswami, and S. H. Low, "Application-Oriented Flow Control: Fundamentals, Algorithms and Fairness," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1282–1291, December 2006.

[9] O. Berder and O. Sentieys, "PowWow : Power Optimized Hardware/Software Framework for Wireless Motes," in *International Conference on Architecture of Computing Systems (ARCS)*, February 2010.

[10] M. Gorlatova, A. Wallwater, and G. Zussman, "Networking Low-Power Energy Harvesting Devices: Measurements and Algorithms," in *IEEE INFOCOM*, April 2011.