

Article

Optimal Rate Allocation in Cluster-Tree WSNs

Antoni Morell ^{1,*}, Jose Lopez Vicario ¹, Xavier Vilajosana ², Ignasi Vilajosana ³ and Gonzalo Seco-Granados ¹

¹ Telecommunications and Systems Engineering Department, Universitat Autònoma de Barcelona, UAB Campus, 08193 Cerdanyola del Vallès, Spain; E-Mails: jose.vicario@uab.cat (J.L.V.); gonzalo.seco@uab.cat (G.S.-G.)

² IT, Multimedia and Telecommunications Department, Universitat Oberta de Catalunya, Rambla Poblenou, 156, 08018 Barcelona, Spain; E-Mail: xvilajosana@uoc.edu

³ World Sensing S.L., Baixada Gomis, 1, 08023 Barcelona, Spain; E-Mail: ivilajosana@worldsensing.com

* Author to whom correspondence should be addressed; E-Mail: antoni.morell@uab.cat; Tel.: +34-93-586-82-39.

Received: 26 January 2011; in revised form: 21 March 2011 / Accepted: 23 March 2011 /

Published: 25 March 2011

Abstract: In this paper, we propose a solution to the problem of guaranteed time slot allocation in cluster-tree WSNs. Our design uses the so-called Network Utility Maximization (NUM) approach as far as we aim to provide a fair distribution of the available resources. From the point of view of implementation, we extend here the authors' proposed Coupled-Decompositions Method (CDM) in order to compute the NUM problem inside the cluster tree topology and we prove the optimality of this new extended version of the method. As a result, we obtain a distributed solution that reduces the total amount of signalling information in the network up to a factor of 500 with respect to the classical techniques, that is, primal and dual decomposition. This is possible because the CDM finds the optimal solution with a small number of iterations. Furthermore, when we compare our solution to the standard-proposed First Come First Serve (FCFS) policy, we realize that FCFS becomes pretty unfair as the traffic load in the network increases and thus, a fair allocation of resources can be considered whenever the price to pay in terms of signalling and computational complexity is controlled.

Keywords: wireless sensor networks; contention free access; fair time slot allocation; distributed optimization; reduced signalling

1. Introduction

Wireless Sensor Networks (WSN) have attracted the attention of the scientific community in the last years. This interest is mainly driven by the large amount of industrial applications that have appeared thanks to the cost reduction of the sensors. In this kind of networks, the standard IEEE 802.15.4 is usually adopted for the PHY and MAC layers [1,2]. This is because this standard has been specially designed with the aim of providing low cost devices with long battery life.

The IEEE 802.15.4 standard, in particular, offers two operating modes: the non-beacon mode and the beacon mode. In the first mode, nodes in the network are not synchronized and access the channel via unslotted Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA). In the beacon-enabled mode, on the other hand, the network is synchronized by means of the beacons transmitted by a node acting as network coordinator and transmissions are organized in time slots. Apart from allowing the nodes to access the channel via contention (with slotted CSMA/CA in this case), this mode also provides guaranteed service to the nodes by allocating Guaranteed Time Slot (GTS) by means of a scheduling strategy. The adoption of GTS is included in order to assure QoS requirements in situations where CSMA/CA is less effective (e.g., under high traffic load). In that sense, new versions of the IEEE 802.15.4 standard such as IEEE 802.15.4e [3] or other coming standards such as WirelessHART [4] or ISA 100.11a [5] evolve with a clear definition of resources in order to manage the Quality of Service (QoS) of the ongoing connections. For example, IEEE 802.15.4e will be mostly based on GTS.

Concerning GTS management, the IEEE 802.15.4 standard follows a first-come-first-served (FCFS) strategy. The problem of such strategy is that the bandwidth allocation is not optimized, large delays can occur and fairness between users is not assured. In order to improve this strategy, several studies in the literature address the problem of bandwidth allocation and scheduling. The problem, however, is usually solved for star topologies (e.g., [6]) and further research is needed for the multi-hop case (mesh and tree-based networks). In [7], for instance, a review of scheduling strategies is presented but most of the algorithms are suboptimal and/or heuristic. Besides, the allocation problem for multi-hop networks presented in [7] is not efficiently solved and the studies referred in this work are related to generic wireless networks (instead of wireless sensor networks). Even in the case of the upcoming standards (IEEE802.15.4e, WirelessHART and ISA100.11a), where the use of guaranteed resources is emphasized, their allocation is still an open issue.

The main problem of multi-hop networks in the beacon-enabled mode is that this mode suffers from beacon collisions between different coordinators in the network. In a tree-deployed WSN, in particular, each parent in the network acts as a coordinator of its respective children. Therefore, each parent is in charge of its local synchronization by sending its own beacons and, then, collisions between the different beacon frames can appear if the system is not organized. In order to solve this issue, IEEE 802.15.4b Task Group presents several collision avoidance strategies [8]. Among all of them, it is of special relevance one strategy based on a time division approach as it is adopted by the Zigbee specification [9]. In particular, this approach consists in dividing time in such a way that the beacon frame of each coordinator is sent during the inactive periods of the rest of coordinators. It is worth noting, however, that the proposed techniques are focused on the beacon collision avoidance problem and that the slot allocation problem is not really solved. Indeed, some works in the literature depart from these strategies to introduce

beacon frame scheduling techniques (see [10] for instance) but the scheduling of GTS slots inside each beacon frame is not directly addressed.

In this paper, we propose a resource allocation algorithm for distributing GTS slots in a tree-deployed wireless sensor network based on the IEEE 802.15.4 standard. Nevertheless, our approach is general and can be applied to other standards that define resources and provide means to share them among the sensors. In other words, although our results have considered the current version of the IEEE 802.15.4 standard, the contribution in this paper is not limited to it. More specifically, we derive a distributed algorithm that provides the optimal slot allocation based on a fair strategy that accounts for both the demand and the priorities of the different nodes. To do so, the problem is formulated as a Network Utility Maximization (NUM) problem, where the fairness among the different nodes can be adjusted, and the optimal solution is derived by means of a novel convex decomposition technique obtained by the authors [11,12]. In particular, this novel technique, referred to as the Coupled-Decompositions Method (CDM), has been extended here to the cluster-tree topology with the aim of efficiently addressing the distributed nature of the network. This extension raised new challenges, such as the computation of the primal and dual projections in a distributed manner or the convergence proof of the method. The main characteristics of the proposed approach are the following: (i) it is able to obtain the optimal solution in a distributed way (*i.e.*, the solution is obtained by distributing the optimization problem through the different nodes in the network), (ii) it converges with a significantly lower number of iterations than other state of the art techniques, and (iii) it requires a lower amount of signaling in the network. It is worth noting that the optimization problem is first formulated in a generic way in order not to restrict the solution to a given scenario. Indeed, the proposed algorithm can be easily extrapolated to perform the joint allocation of time and frequency, which is completely aligned with the philosophy of near future systems (e.g., IEEE 802.15.4e). After that, we focus on the beacon enabled mode of IEEE 802.15.4 with a cluster-tree topology and solve the GTS slot allocation problem by adapting the proposed strategy to this scenario. Our method is compared with the first-come-first-served (FCFS) strategy adopted by the standard, showing that our solution provides a significant improvement in terms of fairness performance as the traffic in the network grows.

The rest of the paper is distributed as follows. In Section 2, the resource allocation problem is formulated as a NUM problem and the proposed Coupled-Decompositions Method is described. After that, in Section 3, the proposed strategy is adapted to a realistic scenario based on IEEE 802.15.4 and numerical results are provided. Finally, the conclusions of the paper are presented in Section 5.

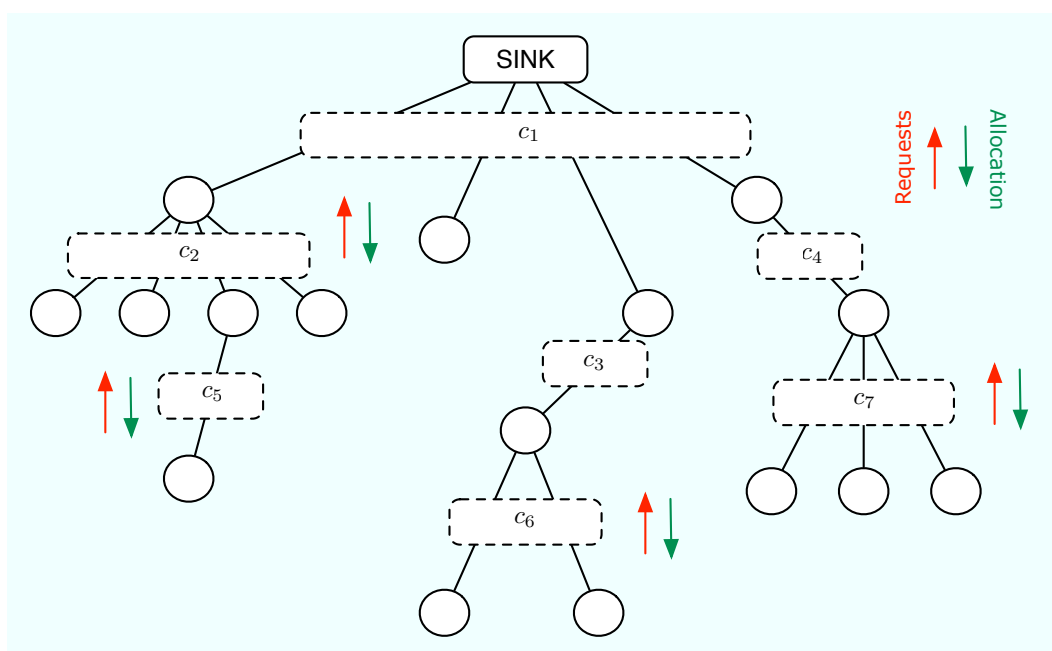
2. Distributed Fair Rate Allocation Algorithm

Let us assume a large-scale Wireless Sensor Network with N nodes that is monitoring a certain area. The mission of the network, as usual, is to gather the node measurements at the sink, where the information is processed. Let us assume that the nodes find their routes to the sink using best-effort solutions such as the Routing Protocol for Low Power and Lossy Networks (RPL) [13] or alternatives [14,15]. As a result, the initial graph of the network turns into a minimum spanning tree. Note that these techniques are very interesting in the context of WSNs since they keep the complexity of the routing protocol reduced, even given that they are suboptimal according to a certain performance metric. As important as routing is the Multiple Access Control (MAC) protocol, that is, how the sensors

gain access to the wireless channel and how the available rate is finally distributed among the nodes. As discussed previously, two different families of approaches are being considered: (i) contention-based (e.g., CSMA/CA) and (ii) demand-based techniques (e.g., GTS allocation). The former is usually implemented due to its simplicity but it becomes inefficient in terms of throughput as the traffic load grows. The latter gives us the optimal solution once a global cost function is determined but it suffers from computational complexity and signalling requirements. In this paper, we propose a distributed algorithm for the second approach that converges fast to the optimal solution, which implies a reduced amount of signalling in the network. Furthermore, it is computationally lightweight.

In the following, we assume that all the nodes in the network first make a rate request according to the amount of information they have to send. After running the distributed allocation algorithm, all of them get a certain transmission rate. The allocation depends on the rate requests, the priorities of the sensors, the minimum guaranteed rates and the maximum transmission rate at each level of the tree topology. All these parameters are known beforehand and the goal is to provide a fair rate distribution so that all the sensors in the network receive their transmission opportunities. On the contrary, note that a solution that maximizes the network throughput may prevent users with bad channel condition from transmitting. In practice, our solution is adequate for Frequency Division Multiple Access (FDMA) or Time Division Multiple Access (TDMA) since in both cases, it is possible to make an arbitrary division of the available total transmission rate in terms of bandwidth and time, respectively. Figure 1 provides a general view of our demand-based solution, where each dashed box represents one multiple-user communication channel between the parent node and its child nodes. Furthermore, a circle represents a node (sensor) and a line between two nodes simply indicates that they can communicate (using the corresponding multi-user channel).

Figure 1. Demand-based multiple access. Rate grants are emitted and depend on the requests and on the rate capacities $\{c_i\}$ inside each cluster.



2.1. Problem Formulation under Fairness Considerations

Let us consider the following optimization problem [16]

$$\begin{aligned} \max_{\{r_j\}} \quad & \sum_{j=1}^N U_j(r_j) \\ \text{s.t.} \quad & m_j \leq r_j \leq M_j, \quad j = 1, \dots, N \\ & \mathbf{A}\mathbf{r} \preceq \mathbf{c} \end{aligned} \quad (1)$$

with variables $r_j \in \mathbb{R}^+$ that represent the rate allocated to the j -th sensor. The functions $U_j : \mathbb{R}^+ \rightarrow \mathbb{R}$ are strictly concave, increasing and differentiable in $\{r_j \mid m_j \leq r_j \leq M_j\}$, where $M_j > m_j$, $m_j \in \mathbb{R}^+$. The vector $\mathbf{c} = [c_1, \dots, c_L]^T$ contains the maximum transmission rate in the L channels of the network and the matrix \mathbf{A} represents the network topology. Note that each wireless channel connects one parent node to its child nodes and that each entry $A_{k,j}$ is either 1 if the flow from the j -th sensor crosses the k -th channel or 0 otherwise. Note also that Problem (1) defines a convex optimization problem [17,18]. Finally, we assume that there exists at least one strictly feasible solution to the problem, *i.e.*, a point that satisfies $\mathbf{A}\mathbf{r} \prec \mathbf{c}$ and $m_j < r_j < M_j \forall j$, so that Slater's constraint qualifications [18] (Section 5.2.3) are fulfilled and thus strong duality holds.

The optimization framework in Problem (1) is known to provide a fair distribution of resources if the utility functions U_j are adequately chosen [19,20]. Moreover, we can specify the desired degree of fairness by fixing the parameter γ in the following family of utility functions [21] (Lemma 2) in Equation (2). Furthermore, $w_j \in \mathbb{R}^+$ is defined as the priority of the j -th flow and it is used to balance the flow allocation towards the j -th sensor once the value of γ is fixed (the higher the value of w_j , the larger the allocation).

$$U_j(r_j; w_j, \gamma) = \begin{cases} w_j \log(r_j), & \gamma = 1 \\ w_j \frac{r_j^{(1-\gamma)}}{1-\gamma}, & \gamma \neq 1 \end{cases} \quad (2)$$

Roughly speaking, with $\gamma \rightarrow \infty$ we get a max-min fair solution, that is, we do not allow to increase the rate of the i -th flow in the network if it implies (due to the constraints) decreasing the j -th flow given that initially $r_j < r_i$. That is the most stringent definition of fairness. The opposite situation is for $\gamma = 0$ and then the goal is to maximize the weighted network throughput. This solution is totally unfair since, depending on the network constraints, some sensors would have no transmission opportunities at all. An intermediate and well-established solution is obtained with $\gamma = 1$, a.k.a., proportional fairness. In this case, a small rate decrease in one flow is accepted if it implies a significant increase in other flows so that the network throughput is less penalized. For further details on fairness formulations, please refer to [21] and references therein. Note, however, that except for $\gamma = 0$, increasing the rate Δr produces a bigger increase in utility when the sensor has a small rate allocation than when it is high.

In our WSN scenario, we assume that the network is built-up so that each sensor knows the route to the sink inside the cluster tree. Furthermore, each parent node has the information about the maximum transmission rate of its cluster, which is given by its own superframe configuration, and each sensor has evaluated its radio link quality, e.g., by means of the Packet Delivery Ratio (PDR). Given all this information, the goal is to compute a fair distribution of the available rate among the sensors. In the following, we focus our attention in the resolution of Problem (1) and we leave the practical implementation issues to Section 3. From the point of view of the optimization mechanism, note that

one possibility is to gather all the requests at the sink node, compute there the optimal rate allocation and send it back to the sensors. Although there exist very efficient methods in the literature to solve generic convex problems, e.g., the so-called interior point methods [18] (Section 11.7), this approach is impractical in large-scale sensor networks as far as it implies an excess of signalling information. Therefore, there is a big interest in solving the problem as efficiently as possible, in a distributed manner. Efficient here means that: (i) the amount of signalling in the network and (ii) the computational load at the sensors must be kept small.

2.2. Existing Work

The problem we have formulated in Problem (1) is known in the literature as the Network Utility Maximization (NUM) problem [22] and several authors have employed classical decompositions, *i.e.*, primal decomposition [17] (Section 6.4.2) and dual decomposition [17] (Section 6.4.1) to solve it. Dual decomposition is usually preferred since it naturally provides a full distributed solution [23,24]. Indeed, the Active Queue Management (AQM) policy employed in the Transport Control Protocol (TCP) can be viewed as a convex optimization problem that is distributedly solved using a dual decomposition approach. Notwithstanding, optimally solving Problem (1) in a distributed manner is now time and signalling consuming due to the fact that the algorithms are based in a projected subgradient approach [17] (Section 2.3). In general, the convergence of the algorithms is slow and furthermore, it depends on a user-adjusted step-size, which is not necessarily optimally fixed.

In order to overcome these drawbacks, we have proposed the Coupled-Decompositions Method (CDM) [11,12]. It simultaneously intertwines primal and dual decomposition techniques in a single approach, converges much faster to the optimal solution and avoids the choice of a user-defined step-size. Note that there exist other primal-dual convex methods in the literature that are different to our approach. Some of them do not take into account the separability of the problem and thus are not true decomposition-based techniques [25,26]. Other works use primal and dual domains as well as the separability of the problem but they concatenate primal and dual domains in their algorithms instead of really mixing them [22,27,28], as we do in our CDM. In the following, we briefly review the classical decomposition methods. Afterwards, we extend the formulation of the CDM to rate allocation problems that exhibit a tree topology thus obtaining a novel distributed version of the method.

2.3. Review of Primal and Dual Decompositions

Let us consider the following equivalent version of Problem (1), which contains the additional variables $\mathbf{y} = [y_1, \dots, y_N]^T$ with $\mathbf{y} \in \mathbb{R}^N$,

$$\begin{aligned} \min_{\{r_j\}} \quad & - \sum_{j=1}^N U_j(r_j) \\ \text{s.t.} \quad & m_j \leq r_j \leq M_j, \quad j = 1, \dots, N \\ & r_j \leq y_j, \quad j = 1, \dots, N \\ & \mathbf{A}\mathbf{y} \preceq \mathbf{c} \end{aligned} \tag{3}$$

Under this transformation, it is clear that given \mathbf{y} , Problem (3) can be separated into N independent problems, which are referred to as the primal subproblems in the specialized literature. In our particular case, these primal subproblems are

$$p_j(y_j) = \begin{cases} \min_{r_j} & -U_j(r_j) \\ \text{s.t.} & m_j \leq r_j \leq M_j \\ & r_j \leq y_j \end{cases} \quad (4)$$

Notwithstanding, we still need to find the optimal values in \mathbf{y} , that is, \mathbf{y}^* . Fortunately, it is known that the primal subproblems define a convex function on the variables y_j and that the Lagrange dual variable λ_j associated with the constraint $r_j \leq y_j$ is a subgradient [29] of p_j at the point y_j [17] (Section 5.4.4).

On the other hand, the primal master problem is defined as

$$\begin{aligned} \min_{\{y_j\}} & \sum_{j=1}^N p_j(y_j) \\ \text{s.t.} & \mathbf{A}\mathbf{y} \preceq \mathbf{c} \end{aligned} \quad (5)$$

and the goal here is to find the optimal values of \mathbf{y} . Given that the subgradients of the subproblems are readily found once they are solved, the primal master problem can be computed using a projected gradient approach [17] (Section 2.3). In other words, $\mathbf{s}_p = -[\lambda_1(y_1), \dots, \lambda_N(y_N)]^T = -\boldsymbol{\lambda}(\mathbf{y})$ is a subgradient of $\sum_{j=1}^N p_j(y_j)$ at \mathbf{y} , where $\lambda_j(y_j)$ is the value of the inner Lagrange multiplier of p_j given the value of y_j , and the following recursion converges to \mathbf{y}^* ,

$$\mathbf{y}^{k+1} = [\mathbf{y}^k - \alpha_p^k \mathbf{s}_p^k]^\dagger = [\mathbf{y}^k + \alpha_p^k \boldsymbol{\lambda}^k(\mathbf{y}^k)]^\dagger \quad (6)$$

where k indexes iterations, α_p^k is the user-defined step-size in primal decomposition and $[\cdot]^\dagger$ is the projection to the set $\{\mathbf{y} \in \mathbb{R}^N \mid \mathbf{A}\mathbf{y} \preceq \mathbf{c}\}$.

Dual decomposition is the dual-domain version of primal decomposition and, in this occasion, we focus on the dual problem [18] (Section 5.2). In particular, we relax only the coupling constraint $\mathbf{A}\mathbf{y} \preceq \mathbf{c}$ in Problem (3) thus defining the following dual master problem,

$$\begin{aligned} \max_{\boldsymbol{\mu}} & \sum_{j=1}^N d_j(\boldsymbol{\mu}) - \boldsymbol{\mu}^T \mathbf{c} \\ \text{s.t.} & \boldsymbol{\mu} \succeq \mathbf{0} \end{aligned} \quad (7)$$

where $d_j(\boldsymbol{\mu})$ are the dual subproblems, defined as

$$d_j(\boldsymbol{\mu}) = \begin{cases} \min_{r_j, y_j^*} & -U_j(r_j) + [\mathbf{A}^T]_j \boldsymbol{\mu} y_j \\ \text{s.t.} & m_j \leq r_j \leq M_j \\ & r_j \leq y_j \end{cases} \quad (8)$$

where $[\mathbf{A}^T]_j$ stands for the j -th row of \mathbf{A}^T . As in primal decomposition, a subgradient of a dual subproblem is found at no cost once $d_j(\boldsymbol{\mu})$ is computed [17] (Section 6.1) and under some non-restrictive conditions, that subgradient is in fact the gradient (see [17] (Section 6.1.1)). In our case, a subgradient is obtained as [30] $\mathbf{A} \boldsymbol{\delta}_j y_j^*(\boldsymbol{\mu})$, where $y_j^*(\boldsymbol{\mu})$ is the optimal value of y_j given $\boldsymbol{\mu}$. Therefore, a subgradient of the dual master problem at $\boldsymbol{\mu}$ is $\mathbf{s}_d = \mathbf{A}\mathbf{y}^*(\boldsymbol{\mu}) - \mathbf{c}$. Finally, the recursion

$$\boldsymbol{\mu}^{k+1} = [\boldsymbol{\mu}^k + \alpha_d^k (\mathbf{A}\mathbf{y}^{*,k}(\boldsymbol{\mu}^k) - \mathbf{c})]^\dagger, \quad (9)$$

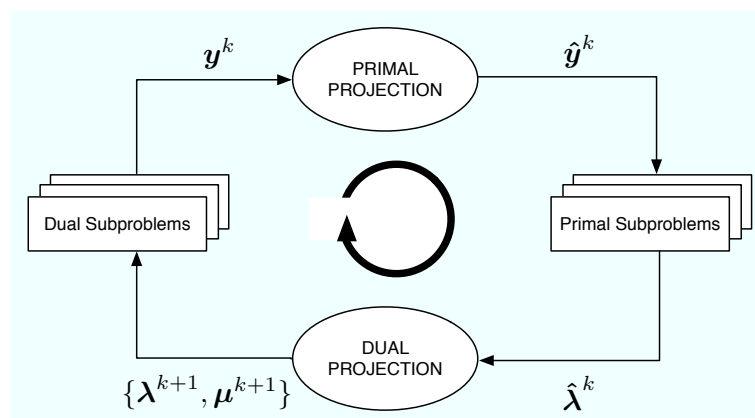
where $[a]^+ = \max\{0, a\}$ and α_d^k is the step-size in dual decomposition, leads to $\boldsymbol{\mu}^*$.

Taking into account that we consider cluster-tree WSNs in this paper, both primal and dual decompositions can be used to deploy distributed flow optimization solutions. However, these techniques suffer from convergence speed and the need of adjusting α_p^k or α_d^k , as discussed before. In the next subsection, we extend the authors' proposed Coupled-Decompositions Method (CDM) [11,12] to provide a fully distributed solution in cluster-tree WSNs. Different to [11], we propose here a single stage algorithm where each iteration of the technique takes into account all the nodes. In [11], the proposed solution had several stages and in each stage, the nodes that belonged to the same network level were optimized thus involving several iterations. Therefore, the total number of iterations grew exponentially with the number of levels in the network. This drawback is overcome in the enhanced version of the method that we describe next.

2.4. Proposed Algorithm

Let us consider the problem formulation in Problem (3) and let us define λ_j as the Lagrange multiplier (equivalently, dual variable) associated with the constraint $r_j \leq y_j$ for all j . As in dual decomposition, we keep $\boldsymbol{\mu} = [\mu_1, \dots, \mu_L]^T$ as the multipliers associated with the constraints in $\mathbf{A}\mathbf{y} \preceq \mathbf{c}$. The technique we propose here intertwines primal and dual decompositions in a single approach and it is composed of four building blocks, namely: (i) dual subproblems; (ii) primal projection; (iii) primal subproblems and (iv) dual projection (see Figure 2). Before going to a more detailed description of the method, let us sketch out the four steps in the Figure 2, that form a complete iteration of the CDM, using a resource-price interpretation. From a practical point of view, this overview together with the contents of Section 2.5 shall suffice to develop our solution in real networks.

Figure 2. Block diagram of the CDM. The primal variables $\mathbf{y}^k, \hat{\mathbf{y}}^k$ and the dual variables $\boldsymbol{\lambda}^k, \hat{\boldsymbol{\lambda}}^k, \boldsymbol{\mu}^k$ are exchanged among the building boxes of the proposed CDM.



It is usual in convex optimization to think of primal variables as resources and dual variables as the prices to be paid for them. Let us assume that $\boldsymbol{\mu}^k$ fixes, at time instant k , the cost of transmitting one unit of rate through each of the available multiple-user channels. Note that the more congested the channel, the higher the cost is and that the cost for a non-congested channel is 0. Given $\boldsymbol{\mu}^k$, each sensor obtains λ_j^{k+1} as the sum of the prices in the multiple-user channels it uses. When the sensor computes the dual subproblem from λ_j^k , it decides how many resources to buy, *i.e.*, y_j^k , which essentially depends on its

own utility function. However, the overall rate acquisition may violate the constraints in the network and it must be corrected. This is the task of the primal projection, that is, it gathers all the values in y_j^k and modifies them to \hat{y}_j^k in order to satisfy the network constraints. Thereafter, the primal subproblems estimate the price they will pay for the new allocation, *i.e.*, $\hat{\lambda}_j^k$. Finally, the dual projection looks at all the individual prices and tries to find a new consensus price μ^{k+1} . The entire process is repeated until all the sensors agree on the prices of the multiple-access channels.

Next, we describe one iteration of the method that starts with μ^k and ends up with μ^{k+1} . Afterwards, we prove in Section 2.8 that $\mu^k \rightarrow \mu^*$. In the following, a star as in y^* or μ^* indicates optimal values and a superindex k as in y^k or μ^k indexes iterations. Furthermore, we make use of some of the Karush-Kuhn-Tucker optimality conditions [18] (Section 5.5.3) of Problem (3). These are

$$\lambda_j = [\mathbf{A}^T]_j \mu \quad (10)$$

$$\mu_j ([\mathbf{A}]_j \mathbf{y} - c_j) = 0, \quad j = 1, \dots, L \quad (11)$$

$$\mu \succeq \mathbf{0} \quad (12)$$

where $[\mathbf{A}^T]_j$ is the j -th row of \mathbf{A}^T . Note that Equation (11) are the slackness conditions [18] (Section 5.5.2) related to the constraints $\mathbf{A}\mathbf{y} \preceq \mathbf{c}$ in Problem (3).

2.4.1. Dual Subproblems

From μ^k , each sensor computes $\lambda_j^k = [\mathbf{A}^T]_j \mu^k$ as in Equation (10). Thereafter, the sensor solves the following optimization problem,

$$d_j(\lambda_j^k) = \begin{cases} \max_{r_j, y_j^k} & U_j(r_j) - \lambda_j^k y_j^k \\ s.t. & m_j \leq r_j \leq M_j \\ & r_j \leq y_j^k \end{cases} \quad (13)$$

Note that Problem (13) coincides with Problem (8) if we take Equation (10) into account and thus we keep the nomenclature, that is, we refer to Problem (13) as the dual subproblems of the CDM. As a result of computing the dual subproblems, the optimal values of the inner optimization variables y_j^k given λ_j^k , *i.e.*, $y_j^{*,k}(\lambda_j^k)$ are found as well. However, these values are not necessarily primal feasible, *i.e.*, they may not satisfy $\mathbf{A}\mathbf{y} \preceq \mathbf{c}$, and there is no guarantee that they accomplish the necessary slackness conditions in Equation (11). In the second step of the method, called the primal projection, we modify the resulting allocation \mathbf{y}^k in order to satisfy both conditions.

2.4.2. Primal Projection

In primal projection, the following optimization problem is solved,

$$\begin{aligned} \min_{\hat{\mathbf{y}}^k} & \quad \|(\mathbf{y}^k - \hat{\mathbf{y}}^k)\|^2 \\ s.t. & \quad [\mathbf{A}]_j \hat{\mathbf{y}}^k \leq c_j \quad \forall j \mid \mu_j^k = 0 \\ & \quad [\mathbf{A}]_j \hat{\mathbf{y}}^k = c_j \quad \forall j \mid \mu_j^k > 0 \end{aligned} \quad (14)$$

and we get the modified variables $\hat{\mathbf{y}}^k$ as the optimal solution of Problem (14). As it is discussed in Section 2.8, the algorithm guarantees that Problem (14) has at least one feasible point so that

Problem (14) is well-defined. However, we need a distributed solution of the primal projection in order to avoid sending all the information to the sink node and computing Problem (14) there. Fortunately, we can benefit from the tree structure of the network to find the optimal solution of Problem (14) distributedly. Note that in cluster-tree topologies, the subset of nodes associated with the i -th constraint is always a subset of the nodes related to the constraint immediately above c_i . See this in the small network example of Figure 3, where

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \succeq \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \hat{y}_4 \end{bmatrix} \quad (15)$$

For the sake of brevity, we describe the distributed computation of Problem (14) using the example in Figure 3, which can be easily extended to the general case. Let us assume that $[\mathbf{A}]_1 \hat{\mathbf{y}} = c_1$ (this is usually the case) and $[\mathbf{A}]_2 \hat{\mathbf{y}} \leq c_2$ [31]. For the sake of simplicity, we omit the index k in this example. As depicted in Figure 3, the idea is to compute the projection at the parent nodes, that is, first the sink allocates \hat{y}_1 and $\hat{y}_{2,ag}$ and thereafter, sensor 2 allocates $\hat{y}_{2,ag}$ to sensors 3-4 and itself, thus obtaining \hat{y}_2 , \hat{y}_3 and \hat{y}_4 . On the other hand, the application of the KKT optimality conditions to our simplified problem reveals us that the optimal solution must satisfy

$$\begin{aligned} \hat{y}_j &= y_j - \mu_1, & j &= 1, 2 \\ \hat{y}_k &= y_k - (\mu_1 + \mu_2), & k &= 3, 4 \end{aligned} \quad (16)$$

where $\mu_1 \in \mathbb{R}$ forces $\sum_i \hat{y}_i = c_1$ and $\mu_2 \in \mathbb{R}^+ \cup \{0\}$ prevents $\hat{y}_3 + \hat{y}_4$ from exceeding c_2 . In other words, μ_2 is bigger than 0 only when the constraint $\hat{y}_3 + \hat{y}_4 \leq c_2$ is violated given μ_1 . Therefore, there are only two possible projections, that is, either i) $\hat{y}_3 + \hat{y}_4 < c_2$ or ii) $\hat{y}_3 + \hat{y}_4 = c_2$. Let us consider a virtual sensor 2_{ag} that groups sensors 2-4 with $y_{2,ag} = \sum_{i=2}^4 y_i$ as in Figure 3. In case (i), the projection is straightforwardly computed [18] (Section 8.1.1) as

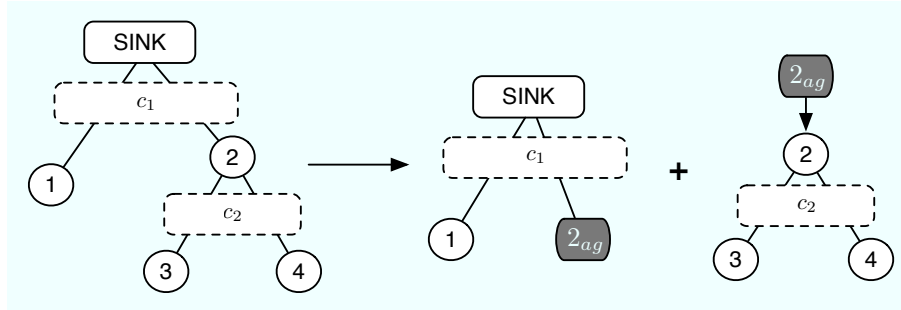
$$\hat{\mathbf{y}}_{ag} = \mathbf{y}_{ag} - \frac{(\mathbf{1}^T \mathbf{y}_{ag} - c_1) \mathbf{n}_{ag}}{\mathbf{1}^T \mathbf{n}_{ag}} \quad (17)$$

where $\hat{\mathbf{y}}_{ag} = [\hat{y}_{1,ag}, \hat{y}_{2,ag}]^T$ is the aggregated allocation to the virtual sensors, $\mathbf{y}_{ag} = [y_{1,ag}, y_{2,ag}]^T$ is their aggregated demand and $\mathbf{n}_{ag} = [n_{1,ag}, n_{2,ag}]^T$ is the number of sensors inside each virtual sensor. In our example, $\mathbf{n}_{ag} = [1, 3]^T$. However, if (ii) is the case, we allocate $c_1 - c_2$ to sensors 1-2 similarly as in Equation (17), as well as c_2 to sensors 3-4. Finally, we need to distinguish between situations (i) and (ii), which can be done using the following threshold for virtual node 2_{ag} ,

$$\hat{y}_{2,ag}^{th} = y_{2,ag} - \frac{n_{2,ag}}{n_{2,ag} - 1} (y_{2,ag} - y_2) \quad (18)$$

This value is the minimum necessary allocation to virtual node 2_{ag} so that the inner rate constraint, *i.e.*, $\hat{y}_3 + \hat{y}_4 \leq c_2$, is satisfied with equality.

Figure 3. Distributed primal projection. Initially, nodes 2, 3 and 4 are grouped together into the supernode 2'. The first allocation distributes c_1 between 1 and 2'. Then, the second allocation distributes the rate granted to 2' among nodes 2, 3 and 4 taking into account the maximum rate c_2 .



In summary, the primal projection is distributedly found as follows. First, we compute the initial allocation with Equation (17). If $\hat{y}_{2,ag} \leq \hat{y}_{2,ag}^{th}$, then we allocate $\hat{y}_{2,ag}$ to sensors 2-4 and we are done. If not, we allocate $c_1 - c_2$ to sensors 1-2 and c_2 to sensors 3-4. Finally, this approach can be scaled to any tree-deployed sensor network. In general, a parent node has to signal at most m_l 4-tuples, where m_l is the number of network levels below it. Each 4-tuple contains an aggregated rate demand, the number of nodes involved in that demand, a threshold value and the rate absorbed once the allocation is above it. Note that this threshold value indicates the maximum possible allocation for a given hypothesis, that is, which channels are congested and which not. Therefore, the absorbed rate must correspond to this specific hypothesis. Once this information is available at the sink node, it can make the first-level allocation and successively, all parent nodes can compute their own allocations.

2.4.3. Primal Subproblems

Once the sensors receive the corrected allocation \hat{y}_j^k , they solve the following optimization problem,

$$p_j(\hat{y}_j^k) = \begin{cases} \max_{r_j} & U_j(r_j) \\ s.t. & m_j \leq r_j \leq M_j \\ & r_j \leq \hat{y}_j^k \end{cases} \quad (19)$$

As above, note that Problem (19) coincides with the subproblems of a plain primal decomposition [22], thus we keep the nomenclature. As a result of computing Problem (19), the inner Lagrange multiplier $\hat{\lambda}_j^k$ associated with the constraint $r_j \leq \hat{y}_j^k$ is obtained. In the dual projection, we use the values of $\hat{\lambda}_j^k$ from all the subproblems to update the values of μ , that is, to obtain μ^{k+1} .

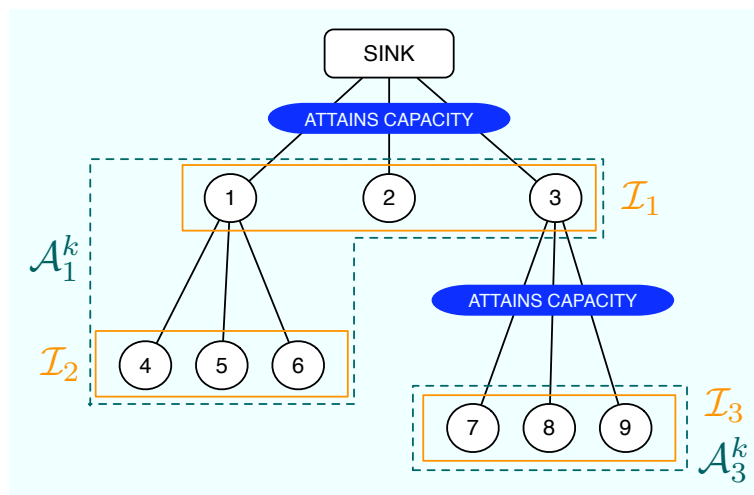
2.4.4. Dual Projection

The dual projection is the last step of the method and its mission is to update μ having the information in $\hat{\lambda}^k$. We already know that one of the optimality conditions of our problem is Equation (10). However, this linear system of equations is in general overdetermined and we can not use it to guess μ^{k+1} from $\hat{\lambda}^k$. Fortunately, it is possible to choose some of the entries in $\hat{\lambda}^k$ to obtain an alternative system of equations, say $\lambda_D^k = B_D^k \mu^{k+1}$, that determines the non-zero values in μ^{k+1} . Note that due to Equation (11), μ_j^{k+1}

is necessarily 0 if the corrected allocation after the primal projection verifies $[\mathbf{A}]_j \hat{\mathbf{y}}^k - c_j < 0$. In the following, we describe how λ_D^k and \mathbf{B}_D^k are constructed.

First of all, let us define the subsets of nodes \mathcal{I}_m and \mathcal{A}_n^k . The subset \mathcal{I}_m does not depend on the iteration number and includes all the nodes using the m-th wireless channel except for the parent node. On the contrary, \mathcal{A}_n^k depends on the subset of active constraints [32] in the network, which evolves on time. Specifically, if the n-th channel is congested, the subset \mathcal{A}_n^k is the union of \mathcal{I}_n with all the subsets \mathcal{I}_m that accomplish: (i) the path from any of the nodes in \mathcal{I}_m to the sink goes through one of the nodes in \mathcal{I}_n and (ii) none of these paths makes use a congested channel, that is, the channels in the path do not attain their rate constraints with equality. If the n-th channel is not congested, we do not define \mathcal{A}_n^k . A simple example can be found in Figure 4, where $\mathcal{I}_1 = \{1, 2, 3\}$, $\mathcal{I}_2 = \{4, 5, 6\}$ and $\mathcal{I}_3 = \{7, 8, 9\}$. In this example, we transmit at the maximum possible rate only in channels 1 and 3 whereas channel 2 is not congested. Therefore, we define $\mathcal{A}_1^k = \mathcal{I}_1 \cup \mathcal{I}_2$ and $\mathcal{A}_3^k = \mathcal{I}_3$ but not \mathcal{A}_2 .

Figure 4. The subsets \mathcal{I}_m and \mathcal{A}_n^k .



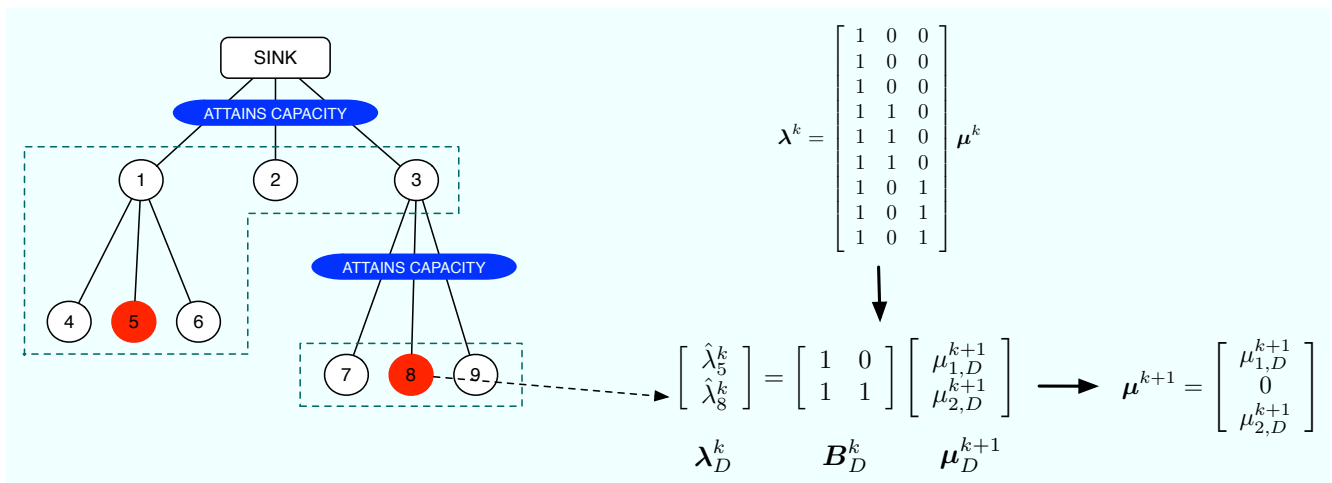
Once the subsets $\{\mathcal{A}_n^k\}$ are identified, we select one sensor from each group using the following selection rules:

- (SR1) Choose the node with the value of $\hat{\lambda}_j^k$ that is closest to λ_j^k (note that $\lambda_j^k = \lambda_i^k$ if $i, j \in \mathcal{A}_n^k$) and that verifies (SR2).
- (SR1) The corresponding value in $\hat{\mathbf{y}}_j^k$ attains $\hat{y}_j^k \in (m_j, M_j)$ (as shown in Proposition 3, there is at least one value of \hat{y}_j^k inside the interval).

(SR1) forces the smallest update from λ_j^k to λ_j^{k+1} whereas (SR2) is necessary to have the required linear dependance between λ and μ ; otherwise there exist more dual variables that come into the equation as shown in Section 2.8. Finally, the linear system of equations $\lambda_D^k = \mathbf{B}_D^k \mu_D^{k+1}$ is constructed by gathering all the selected values of $\hat{\lambda}^k$ in λ_D^k and the corresponding rows of \mathbf{A}^T in \mathbf{B}_D^k . Note that since μ_D^{k+1} does not include the values of μ^{k+1} that are known to be zero beforehand, it is necessary to discard the corresponding rows of \mathbf{A}^T when building \mathbf{B}_D^k . An example can be found in Figure 5, where we assume that the selected value from \mathcal{A}_1^k is $\hat{\lambda}_5^k$ and that the selected value from \mathcal{A}_3^k is $\hat{\lambda}_8^k$. Therefore, we take into account only rows 5 and 8 of \mathbf{A}^T in order to obtain \mathbf{B}_D^k . Furthermore, since channel 2 is not congested

(that's why we have not defined \mathcal{A}_2^k), we already know that $\mu_2^{k+1} = 0$. Consequently, we also discard the second column of \mathbf{A}^T . The resulting linear system of equations is determined and allows us to obtain $\mu_1^{k+1} = \mu_{1,D}^{k+1}$ and $\mu_3^{k+1} = \mu_{2,D}^{k+1}$. However, we want $\boldsymbol{\mu}^{k+1}$ to verify Equation (12) and thus, we propose next a distributed approach to compute $\boldsymbol{\mu}^{k+1}$ and $\boldsymbol{\lambda}^{k+1}$ in a cluster-tree WSN. Our mechanism solves $\boldsymbol{\lambda}_D^k = \mathbf{B}_D^k \boldsymbol{\mu}_D^{k+1}$ if $\boldsymbol{\mu}_D^{k+1} \succeq \mathbf{0}$ and finds an alternative positive solution otherwise, which guarantees the convergence of the method (see Section 2.8). Note that, due to the specific structure of \mathbf{A}^T and also of \mathbf{B}_D^k , we can apply some sort of Gauss elimination as discussed next.

Figure 5. Dual projection example. The reduced system of equations $\boldsymbol{\lambda}_D^k = \mathbf{B}_D^k \boldsymbol{\mu}_D^{k+1}$ is constructed and used in the dual projection to update the non-zero values in $\boldsymbol{\mu}^{k+1}$.



The proposed solution has two phases. The first phase begins at the lowest levels of the cluster-tree and each node sends its current value of $\hat{\lambda}_j^k$ to its parent. The parent, in turn, first checks if the channel with its child nodes is congested. If true, it sends its own value of $\hat{\lambda}_j^k$. If false, it takes the values from the child nodes and also its own value and selects one candidate according to (SR1)-(SR2). This is the value that is sent to the parent in this second case. This process is successively applied until the sink node is reached. Thereafter, the second phase begins. The sink node checks if it attains its own rate constraint. If true, it selects one value taking into account (SR1)-(SR2), fixes μ_1 to that value and sends it to its child nodes. If false, it fixes $\mu_1 = 0$ and sends it. When the child nodes receive that value, say v_i , select one $\hat{\lambda}_j^k$ among its gathered values using again (SR1)-(SR2) and compute μ_i (the dual multiplier associated with its rate constraint) as $\mu_i = \hat{\lambda}_j^k - v_i$. If the resulting $\mu_i \geq 0$, the parent node sends $\hat{\lambda}_j^k$ to its child nodes; if $\mu_i < 0$ or if the channel between the node and its children is not congested, the parent fixes $\mu_i = 0$ and sends the received v_i to its child nodes. This process is repeated until all the values in $\boldsymbol{\mu}^{k+1}$ are updated. Note that, as a result of the proposed distributed dual projection, each parent node only knows one value in $\boldsymbol{\mu}^{k+1}$ but each node receives its own λ_j^{k+1} so that the next iteration of the method can begin.

Finally, we have to decide when to stop the iterations of the CDM, that is, when the current solution is close enough to the optimum. One possibility is to monitor the distance between \mathbf{y}^k and $\hat{\mathbf{y}}^k$ and to stop when the relative distance between both quantities drops below a predefined value ϵ , that is, when

$$\frac{\|\mathbf{y}^k - \hat{\mathbf{y}}^k\|^2}{\|\hat{\mathbf{y}}^k\|^2} < \epsilon \tag{20}$$

However, this approach requires some signalling because we do not have all the necessary data available at a single node. An alternative approach with a very reduced signalling, which is preferred, is to compute the same relative distance at the sink node taking into account aggregated rate values of the child nodes. In this case, we use the previous definitions of $y_{j,ag}^k$ as the aggregated allocation demand (before the primal projection) at the j -th node and the k -th iteration and $\hat{y}_{j,ag}^k$ as the aggregated allocation, *i.e.*, after the primal projection. For example, in the small network in Figure 3, we have $y_{1,ag}^k = y_1^k$ and $y_{2,ag}^k = \sum_{j=2}^4 y_j^k$. Then, we collect at the sink node the values of $y_{j,ag}^k$ and $\hat{y}_{j,ag}^k$ for all its child nodes in the vectors \mathbf{y}_{ag}^k and $\hat{\mathbf{y}}_{ag}^k$, respectively. Using these vectors, we compute the relative distance between them as in Inequation (20). Note that using this approach, we do not introduce additional signalling as far as these quantities are necessary in the primal projection of the method. Furthermore, the decision to stop or continue with the iterations of the CDM can be piggybacked in the messages that send the values of $\hat{y}_{j,ag}^k$ from the parent nodes to the child nodes and thus, the impact in the signalling requirements of the method is very small.

2.5. Algorithmic Form

The CDM can be summarized in algorithmic form as follows:

Algorithm. Coupled-Decompositions Method.

Start with $k = 0$, $\boldsymbol{\mu}^0 = \mathbf{0}$ and repeat:

Dual Subproblems

1 Using $\lambda_j^k = [\mathbf{A}^T]_j \boldsymbol{\mu}^k$, compute

$$d_j(\lambda_j^k) = \begin{cases} \max_{r_j, y_j^k} & U_j(r_j) - \lambda_j^k y_j^k \\ \text{s.t.} & m_j \leq r_j \leq M_j \\ & r_j \leq y_j^k \end{cases}$$

and get y_j^k as the inner maximizer.

Primal Projection

2 Solve

$$\begin{aligned} \min_{\hat{\mathbf{y}}^k} & \quad \|(\mathbf{y}^k - \hat{\mathbf{y}}^k)\|^2 \\ \text{s.t.} & \quad [\mathbf{A}]_j \hat{\mathbf{y}}^k \leq c_j \quad \forall j \mid \mu_j^k = 0 \\ & \quad [\mathbf{A}]_j \hat{\mathbf{y}}^k = c_j \quad \forall j \mid \mu_j^k > 0 \end{aligned}$$

distributedly as sketched in Section 2.4 in order to get \hat{y}_j^k .

Primal Subproblems

3 Solve

$$p_j(\hat{y}_j^k) = \begin{cases} \max_{r_j} & U_j(r_j) \\ \text{s.t.} & m_j \leq r_j \leq M_j \\ & r_j \leq \hat{y}_j^k \end{cases}$$

and get $\hat{\lambda}_j^k$ as the dual multiplier associated with the constraint $r_j \leq \hat{y}_j^k$.

Dual Projection

4 Phase I. The lowest level nodes send $\hat{\lambda}_j^k$ to their parents. Then repeat:

- (a) Each parent node checks the congestion status of the channel between it and its children. If it is congested, the parent sends its own value of $\hat{\lambda}_j^k$ to its own parent. Otherwise, it considers also the values received from its child nodes, selects one candidate using (SR1)-(SR2) and sends it.

Until the sink node has received one value of $\hat{\lambda}_j^k$ from each of its child nodes.

Phase II. Start at the sink node with $v_1 = 0$ and repeat:

- (a) If the channel between the node and its children is congested, use (SR1)-(SR2) to select one candidate among the values received in Phase I, say $\hat{\lambda}_j^k$. Otherwise, $\hat{\lambda}_j^k = 0$.
- (b) Compute $\mu_i^{k+1} = \hat{y}_j^k - v_i$ (v_i is the message received from the parent node in Phase II, i is the channel index).
- (c) If the resulting value μ_i^{k+1} satisfies $\mu_i^{k+1} \geq 0$, send $\hat{\lambda}_j^k$ to the child nodes. Otherwise, fix $\mu_i^{k+1} = 0$ and send v_i .
- (d) Set the next update of λ_j at the child nodes, *i.e.*, λ_j^{k+1} , to the received value, either $\hat{\lambda}_j^k$ or v_i .

Until all the values in μ^k are determined and all the values in λ^k are updated to λ^{k+1} .

Until convergence.

Furthermore, Figure 6 summarizes in a flowchart all the tasks that a sensor has to execute in order to obtain its optimal rate allocation.

2.6. Signalling in the Coupled-Decompositions Method

In distributed implementations such as the one proposed in this paper, signalling plays a crucial role. In the following, we review a complete iteration of the proposed method from the point of view of signalling and we represent it graphically in Figure 7. Let us assume that at the k -th iteration, all the nodes know the values of λ_j^k and use them to compute the dual subproblems. As a result, they locally obtain the values of y_j^k involving no signalling. Thereafter, in primal projection, the parent nodes must send their aggregated request $y_{j,ag}^k$, *i.e.*, the sum of the values y_j^k that belong to their subtree, possibly together with a list that includes allocation threshold values and absorbed rates, as discussed in Equations (17) and (18). Note that the size of the list scales linearly with the number of levels in the subtree. Once the request reaches the sink node, it allocates resources to the first level of the tree. In turn, the nodes at the first level allocate their resources to the nodes at the second level and so forth. When all the nodes have received the corresponding value \hat{y}_j^k , they locally compute the primal subproblems and obtain the values $\hat{\lambda}_j^k$. Finally, in order to compute the dual projection, the nodes send the values $\hat{\lambda}_j^k$ to their parent nodes. Thereafter, the sink node computes μ_1 and broadcasts the value λ_j^{k+1} to their child nodes (note that all the child nodes receive the same value). In turn, the parent nodes compute their own

value μ_j and broadcast λ_j^{k+1} to their child nodes. This process is successively applied until all the nodes receive the updated value of the dual variable, which completes one iteration of the proposed CDM.

Figure 6. Flowchart of the tasks performed by the sensor.

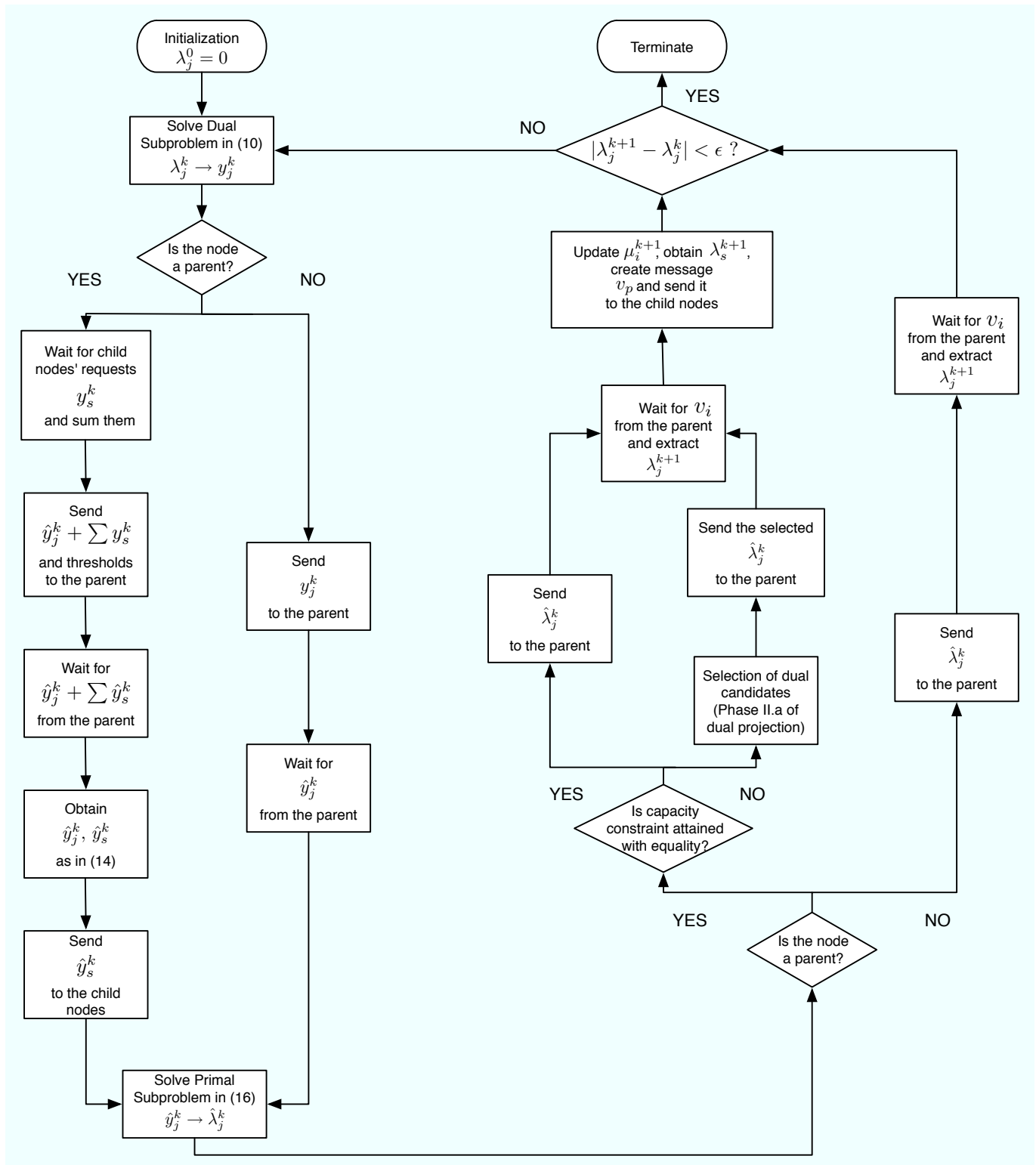
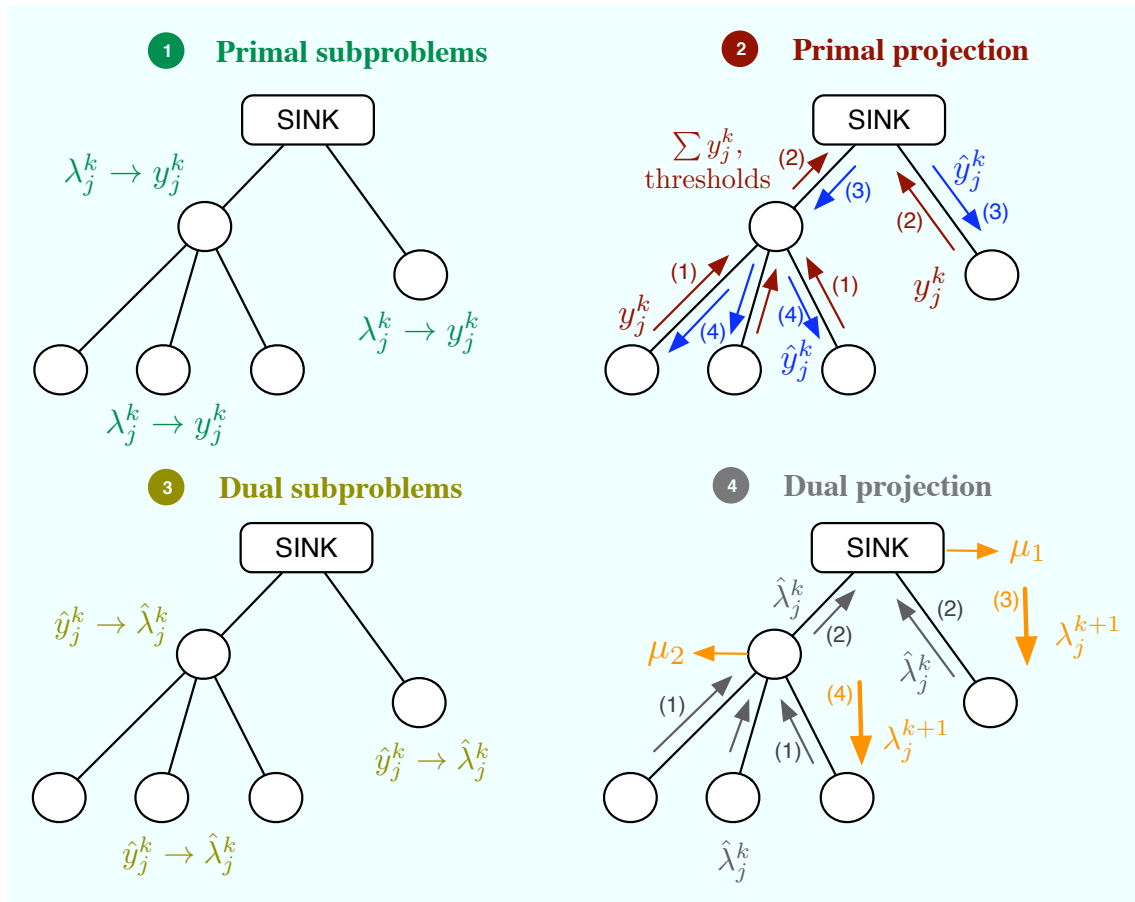


Figure 7. A complete iteration of the CDM from the point of view of signalling.



2.7. Differences with Classical Decompositions

From the point of view of the algorithm, two of the building blocks in the proposed method are the same as in the classical decompositions, that is, the definition of the primal and the dual subproblems has not been modified. On the contrary, the primal and the dual projections differ from the primal and dual master problems as far as subgradients are no longer used. Notwithstanding, the primal projection resembles the projection that appears in the primal master problem. The difference is that we force the constraints in $A\mathbf{y} \preceq \mathbf{c}$ to be attained with equality when the corresponding dual variable in $\boldsymbol{\mu}$ is not zero. Finally, the dual projection is completely different from the master dual problem and is indeed the key block in our proposal.

From the point of view of signalling, one iteration in traditional primal decomposition algorithms is equivalent to the first two steps in the proposed method. In other words, from y_j^k the nodes compute the primal subproblems and calculate $y_j^{k+1} = y_j^k + \alpha_p^k \lambda_j^k$, which requires no signalling. Thereafter, the projection $[\cdot]^\dagger$ in traditional primal decomposition algorithms requires the same amount of signalling as in the worst-case of the primal projection at the CDM, that is, when $[A]_j \mathbf{y} \preceq c_j, \forall j$. Similarly, one iteration in the traditional dual decomposition is equivalent to the last two steps of the CDM. That is, from $\boldsymbol{\mu}^k$, the nodes implementing the traditional dual decomposition approach obtain y_j^k , which requires no signalling, and afterwards send these values to the parent nodes. Using that information the parent nodes can extract the values of $\boldsymbol{\mu}^{k+1}$ and send the required linear combination of the values in $\boldsymbol{\mu}^{k+1}$ back to the child nodes. In this case, the total amount of signalling is exactly the same as in the dual projection of the

CDM, where the nodes send the current allocation, the parent nodes compute the required subgradient and send the updates of the values in λ afterwards.

In summary, the key step that allows us to combine traditional primal and dual decompositions in a single algorithm, the CDM, is the dual projection. Conceptually, the big difference is to interpret the primal variables $\mathbf{y}^k, \hat{\mathbf{y}}^k$ and the dual variables $\lambda^k, \hat{\lambda}^k$ as candidates to the optimal solution that must accomplish the well-known KKT optimality conditions. In terms of signalling, one iteration of the CDM is equivalent to two iterations of the traditional decomposition algorithms.

2.8. Convergence to the Optimal Rate Allocation

Let us start the iterations of the proposed CDM with $\mu^0 = \mathbf{0}$. In the following we show that $\mu^k \rightarrow \mu^*$ as k increases. Initially, we assume that at the k -th iteration $\mu_i^k > 0$ only if $\mu_i^* > 0$, that is, the active constraints (*i.e.*, attained with equality) at the k -th iteration are exactly the same as in the optimal solution and we prove that the algorithm converges to μ^* . Afterwards, we describe the mechanism used by the CDM in order to activate or deactivate constraints until the subset of active constraints in the optimal solution is found out.

Convergence to the Optimal Solution Given the Subset of Active Constraints at the Optimum

Let us first recall in Proposition 1, a known result that establishes the relationship between primal and dual variables in the primal and dual subproblems [11,12].

Proposition 1 *Let us consider the j -th primal subproblem $p_j(\hat{y}_j^k)$ in Problem (19) and the j -th dual subproblem $d_j(\lambda_j^k)$ in Problem (13) of the CDM. Then, the inner optimization variable $\hat{\lambda}_j^k$ is a decreasing function of \hat{y}_j^k in the primal subproblems and the inner optimization variable y_j^k is a decreasing function of λ_j^k in the dual subproblems.*

Proof See [12] (Lemma 2).

Note, in particular, that if $\hat{\lambda}_j^k \leq \lambda_j^* = \mathbf{A}^T \mu^*$ as a result of the primal subproblems, then it is true that $\hat{y}_j^k \geq y_j^*$ and vice versa. Similarly, if $y_j^k \leq y_j^*$ as a result of the dual subproblems, then $\lambda_j^k \geq \lambda_j^* = \mathbf{A}^T \mu^*$ and vice versa. Note also that given μ^k , all the values $\lambda_j^k = [\mathbf{A}^T]_j \mu^k$ with $j \in \mathcal{I}_i$ attain either $\lambda_j^k \geq \lambda_j^*$ or $\lambda_j^k \leq \lambda_j^*$ as far as $[\mathbf{A}^T]_j$ is exactly the same for all $j \in \mathcal{I}_i$. That is, if we start an iteration of the CDM with λ^k in the dual subproblems, it holds that $y_j^k \geq y_j^*$ for all $j \in \mathcal{I}_k$ if $\lambda_j^k \leq \lambda_j^*$ and vice versa due to Proposition 1.

Once we obtain the values \mathbf{y}^k from the dual subproblems, the primal projection in Problem (14) is computed. Since we assume that Slater's condition holds, Problem (14) defines a convex program where strong duality is fulfilled. Therefore, KKT optimality conditions can be applied and the solution to the primal projection can be expressed as

$$\hat{\mathbf{y}}^k = \mathbf{y}^k - \mathbf{A}^T \boldsymbol{\gamma} \quad (21)$$

where $\gamma_i \geq 0$ if $[\mathbf{A}]_i \hat{\mathbf{y}} \leq c_i$ and $\gamma_j \in \mathbb{R}$ if $[\mathbf{A}]_j \hat{\mathbf{y}} = c_j$. In the first case, $\gamma_i = 0$ if the corresponding rate constraint is not exceeded and $\gamma_i > 0$ if the opposite situation occurs in order to satisfy it with equality. In the second case, γ_j can take any value in order to satisfy the corresponding constraint. As a result of the primal projection, it is clear that all the nodes in every subset \mathcal{I}_i either increase or decrease the initial

allocation y_j^k in order to obtain \hat{y}_j^k . In other words, if $m, n \in \mathcal{I}_i$, then it is impossible, for example, that $\hat{y}_m^k = y_m^k - a_1$ and $\hat{y}_n^k = y_n^k + a_2$ with $a_1, a_2 \geq 0$. Moreover, if $\boldsymbol{\mu}^k$ is close enough to the optimal solution and assuming that $\mu_j^k > 0$ only if $\mu_j^* > 0$, the constraints that are attained with equality after the primal projection correspond only to the values $\mu_j^k > 0$.

In that situation and given the tree structure of the network, it is verified that

$$\sum_{j \in \mathcal{A}_m^k} \hat{y}_j^k = \sum_{j \in \mathcal{A}_m^k} y_j^* \quad (22)$$

For example, in Figure 4 we immediately notice that $\sum_{j \in \mathcal{A}_3^k} \hat{y}_j^k = c_3$ and $\sum_{j \in \mathcal{A}_1^k} \hat{y}_j^k = c_1 - c_3$. Therefore, we can conclude that i) some values of \hat{y}_j^k with $j \in \mathcal{A}_m^k$ attain $\hat{y}_j^k \geq y_j^*$ while the rest accomplish $\hat{y}_j^k \leq y_j^*$ and ii) all the values \hat{y}_j^k with $j \in \mathcal{A}_m^k$ either increase or decrease after the primal projection. This result is easily extrapolated to the dual values $\hat{\lambda}_j^k$ obtained from the primal subproblems when \hat{y}_j^k are the inputs if we take into account Proposition 1. We write it down in the following proposition.

Proposition 2 *The dual values $\hat{\lambda}_j^k$ that result from the computation of the primal subproblems of the CDM at the k -th iteration accomplish: (i) some values of $\hat{\lambda}_j^k$ with $j \in \mathcal{A}_m^k$ attain $\hat{\lambda}_j^k \geq \lambda_j^*$ while the rest accomplish $\hat{\lambda}_j^k \leq \lambda_j^*$ and (ii) all the values $\hat{\lambda}_j^k$ with $j \in \mathcal{A}_m^k$ either increase or decrease with respect to λ_j^k (note that $\lambda_l^k = \lambda_n^k$ if $l, n \in \mathcal{A}_m^k$).*

In the dual projection, we choose one value $\hat{\lambda}_j^k$ from each subset \mathcal{A}_m^k . In particular, the selected value has to accomplish (SR1)-(SR2), that is: (i) the corresponding \hat{y}_j^k verifies $\hat{y}_j^k \in (m_j, M_j)$ and (ii) it is the closest value in the subset to λ_j^k for any $j \in \mathcal{A}_m^k$. The following proposition guarantees the existence of the required value.

Proposition 3 *Let y^k be the result of the dual subproblems of the CDM at the k -th iteration. Then, the primal projection provides at least one node with $\hat{y}_j^k \in (m_j, M_j)$ at each subset \mathcal{A}_m^k .*

Proof See the Appendix.

Given that $U_j(r_j)$ is concave and strictly increasing, it is straightforward to see that y_j must equal r_j in Problem (3). Furthermore, some of the KKT optimality conditions of the problem reveal us that

$$\frac{\partial U_j}{\partial r_j} - \lambda_j + \alpha_j - \beta_j = 0 \quad (23)$$

$$\lambda_j - [\mathbf{A}^T]_j \boldsymbol{\mu} = 0 \quad (24)$$

$$\alpha_j(m_j - r_j) = 0 \quad (25)$$

$$\beta_j(M_j - r_j) = 0 \quad (26)$$

where α_j and β_j are the dual variables associated with the constraints $r_j \geq m_j$ and $r_j \leq M_j$, respectively. Therefore, if we fix y_j equal to \hat{y}_j^k , with $\hat{y}_j^k \in (m_j, M_j)$, then α_j and β_j must be 0 due to the slackness constraints, *i.e.*, Equations (25–26). Applying this observation and $r_j = \hat{y}_j^k$ to Equation (23), we realize that λ_j is uniquely determined. Besides, taking also Equation (24) into account, we can conclude that a good election of \hat{y}_j^k provides us useful information about the linear combination $[\mathbf{A}^T]_j \boldsymbol{\mu}$. In the CDM, this argumentation justifies our interest in the values of \hat{y}_j^k that attain $\hat{y}_j^k \in (m_j, M_j)$. Otherwise the values of α_j and β_j are unknown and we can not use $\hat{\lambda}_j^k$ to determine $\boldsymbol{\mu}^{k+1}$.

Finally, in the dual projection we construct the determined linear system $\lambda_D^k = B_D^k \mu_D^{k+1}$ with the selected values $\hat{\lambda}_j^k$, thus computing the values in μ^{k+1} corresponding to the active constraints (the remaining ones are fixed to zero). Combining the results in Proposition 2 with (SR1)-(SR2), we realize that each dual candidate $\hat{\lambda}_j^k$ approaches λ_j^* (w.r.t. λ_j^k) without overtaking it and that the candidate is valid to determine some of the variables in μ^{k+1} . Since in the dual projection we obtain a determined linear system using these dual candidates, the new update $\lambda^{k+1} = [A]^T \mu^{k+1}$ necessarily approaches $\lambda^k = [A]^T \mu^k$. Hence, we can state that $\mu^k \rightarrow \mu^*$. We summarize this result in the following lemma.

Lemma 1 *The iterations of the CDM satisfy $\mu^k \xrightarrow{k \rightarrow \infty} \mu^*$ given the subset of active constraints at the optimum.*

Activation/Deactivation of Constraints

In the following, we use the partial result obtained in Lemma 1 to show that the proposed version of the CDM converges to the optimal solution. Let us state this in the following theorem.

Theorem 1 *The proposed CDM is able to determine the subset of active constraints of Problem (3) at the optimal solution. Hence, the method converges to the optimal solution.*

Proof We have proved in Lemma 1 that, given the subset of active constraints at the optimum, the CDM converges to μ^* . Therefore, we need to prove that this subset is eventually found out. Let us start with the activation of a non-active constraint and let us assume that the l -th constraint is active at the optimal solution but not at the k -th iteration. This implies that, if we remove the constraint $[A]_l \mathbf{y} \leq c_l$ from the original problem formulation in Problem (3), then the optimal solution to the modified problem \mathbf{y}^\ddagger would satisfy $[A]_l \mathbf{y}^\ddagger \geq c_l$. In other words, the successive updates of the CDM tend to \mathbf{y}^\ddagger when the l -th constraint is not included in the active subset of constraints. However, there will be an iteration q where $[A]_l \mathbf{y}^q \geq c_l$ so that the primal projection will detect that constraint violation, correct it and the l -th constraint will be activated.

The opposite situation arises when the l -th constraint is active at the k -th iteration but not at the optimal solution. In that case, if the l -th constraint is not deactivated, *i.e.*, the primal projection keeps forcing $\mathbf{A}\mathbf{y} = c_l$, the CDM would converge to a solution μ^* with $\mu_l^* < 0$ (assuming that the dual projection allows negative values). Note that the allocation to the variables involved in the l -th constraint increases with respect to the optimum whereas the allocation to the remaining variables is reduced or remains unchanged in order to satisfy the rate constraints. Looking at the situation from the dual point of view, that is, taking into account Proposition 1, we notice that the values λ_j^* involved in the l -th constraint must converge to a value that is below λ_j^* . Also, the remaining values remain unchanged or above the corresponding λ_j^* . If we compare those values with $\lambda^* = A^T \mu^*$ and we know that $\mu_l^* = 0$, then we realize that $\mu_l^* < 0$. At this point, let us compute one more iteration of the CDM, thus obtaining λ^{*+1} . In this case, since the values λ_j^{*+1} involved in the l -th constraint increase with respect to $\lambda^* = A^T \mu^*$ because μ_l^{*+1} is now 0, the corresponding values y_j^{*+1} decrease. Therefore the l -th rate constraint will not be exceeded, preventing its reactivation at the next primal projection.

3. Application to Cluster-Tree Wireless Sensor Networks

The algorithm developed in the previous section can be applied to cluster-tree WSNs with the aim of providing global fair resource allocations with a reduced impact in terms of energy consumption due to signalling. In this work, in particular, we focus on the beacon-enabled mode of IEEE 802.15.4 [1]. This mode divides the transmission through time slots and, besides, two periods are defined depending on the kind of access used by the nodes in the network: contention access period (CAP), where slotted CSMA/CA is adopted, and contention-free period (CFP), where GTS slots are distributed to the nodes (see Figure 8). Since time slots can have different sizes from one cluster to another, we distribute resources in terms of rate, *i.e.*, r_j is the rate allocated to the j -th sensor. Thereafter, we use these rates to derive the time slot allocation. Note that depending on each specific Superframe (SF) configuration, the maximum transmission rate in each cluster changes. Furthermore, we will also take into account the quality of the radio links by means of the Packet Delivery Ratio (PDR), that is, PDR_j is the ratio between correctly received packets and sent packets at the j -th sensor. As previously commented, beacon collision problems can appear in a cluster-tree WSN based on the beacon enabled mode. For that reason, we take into account a beacon collision mechanism. In particular, we consider the time-division approach adopted by the Zigbee specification [9], *i.e.*, time is divided in such a way that the beacon frame of each coordinator is sent during the inactive periods of the rest of coordinators (see Figure 9).

Figure 8. Beacon frame configuration adopted by the IEEE 802.15.4 standard [1].

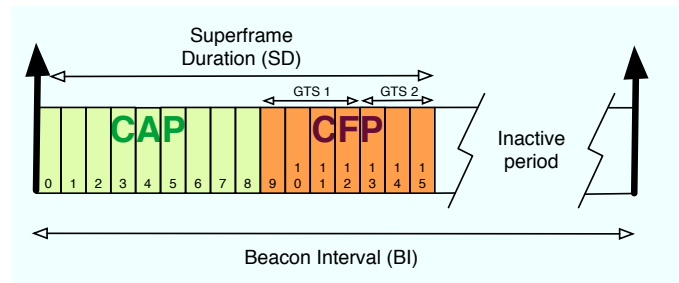
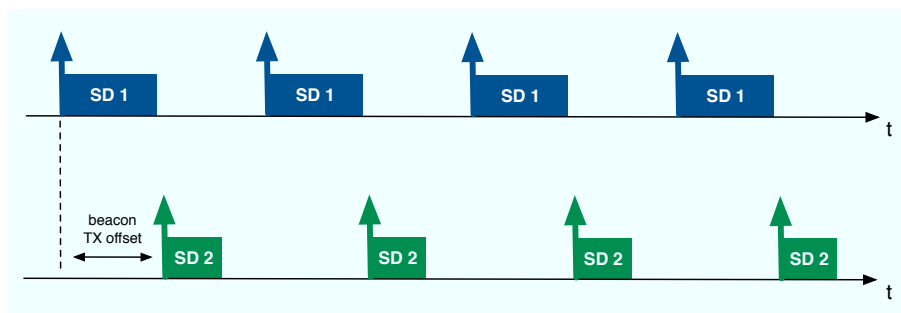


Figure 9. Beacon frame collision avoidance based on the time division approach adopted by the Zigbee specification [9].



Given the sensor demands in $\mathbf{M} = [M_1, \dots, M_N]^T$ and the minimum guaranteed allocation in $\mathbf{m} = [m_1, \dots, m_N]^T$ (both in terms of rate), the optimum allocation is computed by solving the following optimization problem using the CDM,

$$\begin{aligned} \max_{\{r_j\}} \quad & \sum_{j=1}^N U_j(r_j \cdot PDR_j; w_j, \gamma) \\ \text{s.t.} \quad & m_j \leq r_j \leq M_j, \quad j = 1, \dots, N \\ & \mathbf{Ar} \preceq \mathbf{c} \end{aligned} \quad (27)$$

where the priority values w_j allow us to balance the allocation towards some sensors in the network and γ fixes the degree of fairness as discussed in Equation (2). Moreover, the radio link qualities are also considered as far as $r_j \cdot PDR_j$ is the effective data rate assigned to the j -th user. In particular, note that when $U_j(r_j) = w_j \log(r_j \cdot PDR_j)$ the resulting allocation is insensitive to PDR_j because $U_j(r_j \cdot PDR_j) = w_j [\log(PDR_j) + \log(r_j)]$ and thus, proportional fairness does not benefit sensors with bad channel condition. If we fix $\gamma > 1$ instead, the resulting allocation is balanced towards the bad sensors. Note that in the limit, *i.e.*, with $\gamma \rightarrow \infty$, max-min fairness does not admit low effective data rates even if this implies a significant reduction in terms of the network throughput.

Once the optimal allocation is obtained, it has to be mapped onto a time slot assignment. Since the WSN scenario can be assumed quasi-static in most of the real-life applications, it is reasonable to sustain an allocation for a given number of Beacon Intervals (BI), say nBI . Therefore, the real-valued number of time slots assigned to the j -th sensor on the nBI periods is

$$TS_j = \frac{r'_j \cdot nBI \cdot T_{BI}}{b_j} \quad (28)$$

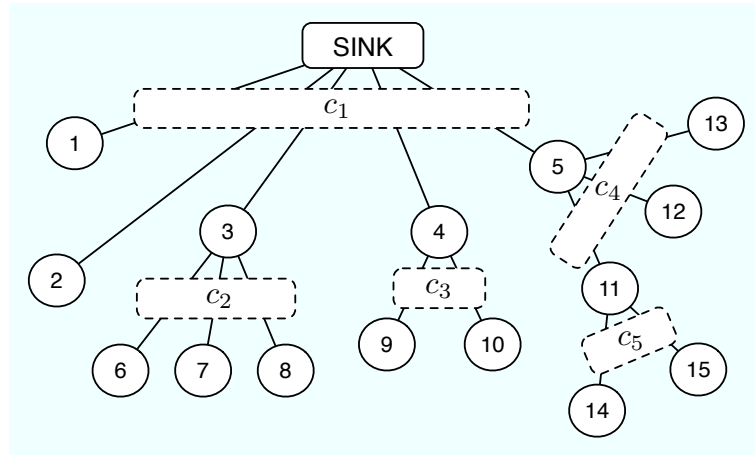
where r'_j is the rate across the j -th sensor, that is, its own rate plus the rate of all the nodes below it. Furthermore, T_{BI} is the BI duration and b_j is the packet size in bits at the j -th sensor. Since TS_j in Equation (28) is real-valued, the cluster head has to round the allocation of its nodes. We propose to allocate first $\lfloor TS_j \rfloor$ time slots to each node and then successively allocate one extra time slot to the sensors with the biggest value of $TS_j - \lfloor TS_j \rfloor$, where $\lfloor \cdot \rfloor$ represents the floor function. Note that the higher the nBI , the higher the optimality of the time slot allocation. However, if the network is substantially dynamic, we shall fix $nBI = 1$ and implement other rounding strategies that are out the scope of this paper. In any case, we need algorithms that converge fast to the optimal solution if we want to track the changes in the network efficiently. This is achieved with the proposed CDM as shown in the next simulation results.

4. Numerical Results

Let us consider the network configuration depicted in Figure 10, with 15 sensor nodes, one sink node and 5 capacity constraints that deploy a three-level cluster tree. Note that the following results can be extrapolated to WSNs with a larger number of nodes and a larger number of levels and thus, our simplified example has no loss in generality. Specifically, the Superframe (SF) is configured as follows: T_{BI} is set to 245.76 ms and is the same at all levels; the SF duration is 61.44 ms at the highest level, 30.72 ms at the intermediate level and 15.36 ms at the lowest level. According to [33], this configuration provides 9.38 kbps, 10.94 kbps and 13.02 kbps of data rate, respectively. Furthermore,

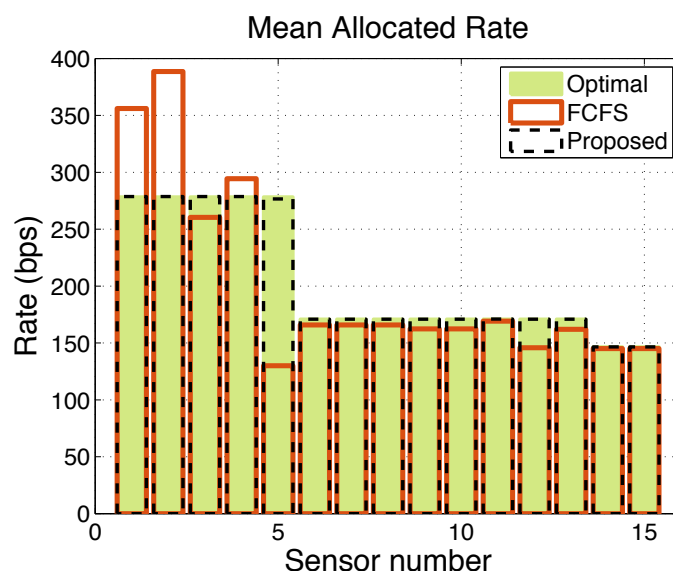
if we consider 15 slots per BI dedicated to the Guaranteed Time Slot (GTS) allocation, the size of the slots in bits is 9, 21 and 50, respectively. Accordingly, the rate capacities of the clusters are set to $c = [3.0516, 1.2820, 1.2820, 1.2820, 0.5496]^T$ kbps.

Figure 10. Network configuration used in the numerical results.



In our simulations, we compare our CDM-based approach to the widely-used First Come First Serve (FCFS) policy. We assume, without loss of generality, that the values of w_j and PDR_j are set to 1 and that all the sensors are of the same type and thus, all of them need to transmit n bits per BI. However, the rate requests are quantized by the time slot duration at each sensor so that each node asks for \bar{n}_i/T_{BI} kbps, where \bar{n}_i is the nearest multiple (above) of the packet size in bits at the i -th node. Figure 11 shows an allocation example for $n = 20$ bits. We notice that the CDM-based policy provides a rate allocation that is very close to the optimal, whereas the FCFS solution does not guarantee a fair resource allocation, even when the throughput is not penalized. Note that, in the example under consideration, there exists no trade-off between fairness and throughput because we have set error-free links. Note also that our proposed time slot allocation does not provide the optimum fair rate allocation due to the rounding effects.

Figure 11. Allocation example for $n = 20$ bits.

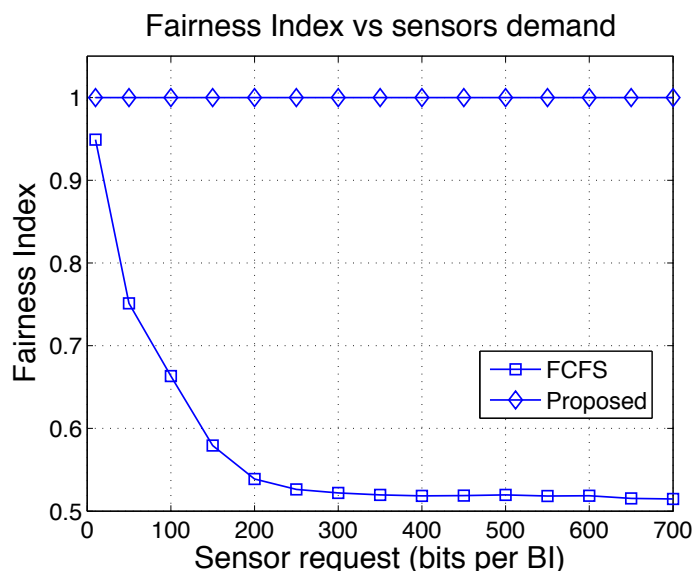


In order to study the fairness degree provided by both solutions, we plot in Figure 12 the Fairness Index (FI) as a function of the requested bits n . The FI takes values between 0 and 1 and measures how far is a certain resource allocation with respect to the one that is considered optimal or most fair. If $FI = 0$ the allocation is totally unfair and if $FI = 1$, the resulting allocation is completely fair. A value that is in-between such as 0.6 may be interpreted as follows: the 60% of allocations are considered fair in mean. The FI is computed as [34]

$$FI = \frac{\sum_j z_j^2}{N(\sum_j z_j)^2} \quad (29)$$

where $z_j = \frac{x_j}{r_j^*}$, being r_j^* the optimal fair allocation and x_j the allocation under test. Therefore, x_j represents the rate allocation that results from either our CDM-based time slot distribution or the standard-proposed FCFS approach. Furthermore, r_j^* is the real-valued optimal rate allocation, that is, without rounding. Note in the figure that the CDM-based approach provides a solution that is very close to the optimal whereas the FCFS allocation tends to be more unfair as the transmission requirements of the sensors grow until it stabilizes around 0.5, which is pretty unfair. Therefore, FCFS is an acceptable strategy for small traffic loads, that is, when the network resources suffice to satisfy all the sensor requirements. However, the allocation is unfair at high loads because the mechanism does not distinguish sensors that are parents of a sub-cluster tree from sensors that are not. Since the two receive the same amount of resources in mean, the sensors in the sub-cluster tree (including the parent node) are thus penalized. In our example, the routing policy of a sensor is to first relay the packets with a lower time stamp and then send its own packets, which explains the higher allocation of sensors 1 and 2 in Figure 11.

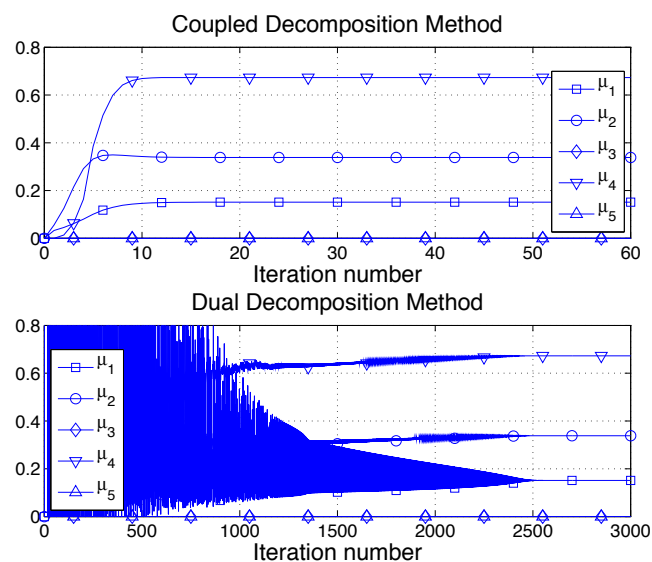
Figure 12. Fairness comparison.



Finally, signalling is a very important aspect to take into account when considering distributed optimization techniques. The advantage is clear: the network operates at the desired optimal point (proportional fairness in our examples). Notwithstanding, as opposite to other policies such as FCFS, sensors need to exchange information thus draining the sensor batteries. Therefore, distributed optimization approaches make sense if the amount of signalling information is kept small. In Figure 13

we compare the proposed CDM against the classical dual decomposition approach in terms of iterations required to converge to the optimal solution. In this example, requests, maximum transmission rates, minimum guaranteed rates and sensor priorities are picked at random using uniform distributions. In particular, $d_i, c_i \sim \mathcal{U}[0, 50]$ kbps, $m_i \sim \mathcal{U}[0, 0.5]$ kbps and $p_i \sim \mathcal{U}[0, 2]$. In our simulations, the step-size in dual decomposition is adjusted to provide good performance in general, fixing it to $\alpha^k = \frac{0.5}{\sqrt{k}}$ (note that the CDM has no parameter to configure). See in Figure 13 one numerical example. Note that although one iteration of the CDM implies twice the amount of signalling required in one iteration of dual decomposition, the number of iterations in the latter is excessively large and makes the technique impractical in the WSN context. On the contrary, the CDM finds the optimal solution using 10–30 iterations in general. In other words, the CDM requires about 40 N to 120 N messages to achieve a fair distribution of the available rate whereas the so-called dual decomposition solution needs to exchange up to 5,000 N in our example. If we consider 32-bit messages, our solution requires 56.25 kb in total whereas the dual decomposition approach needs 2343.75 kb of signalling.

Figure 13. Comparison between the CDM and the dual decomposition.



5. Conclusions

In this paper, we present a resource allocation technique that is designed to fairly distribute the available time slots among the sensors of a cluster-tree WSN. Our solution is based in convex decomposition and, in particular, it combines the so-called primal and dual decompositions in a single algorithm that provides superior performance in terms of convergence speed. Thanks to the proposed Network Utility Maximization (NUM) formulation, it is possible to choose the most adequate definition of fairness in our application, going from max-min fairness to max-sum rate. As shown in the document, the standard-proposed First Come First Serve policy provides pretty unfair allocations when the traffic load in the network increases and proportional fairness is considered. Moreover, the NUM formulation takes into account the different rate capacities in the network, which are due to different superframe configurations, as well as the radio link qualities by means of the Packet Delivery Ratio.

Throughout the document, we focus our attention on the practical application of the proposed method. On one hand, the conversion from rate (required by the algorithm) to time slots is addressed. On the other hand, the signalling involved in the computation of the global optimal solution is detailed. Specifically, our method requires the double amount of signalling per iteration when compared to dual decomposition. Notwithstanding, the number of iterations can be reduced up to a factor of 1,000 so that the total amount of signalling is reduced about 500 times. Besides, the CDM has no parameter to be adjusted, which is a true impairment in practice.

In summary, we can conclude that the most remarkable benefits of our solution are:

- The network can be adjusted to operate with several fairness definitions by fixing the value of γ .
- The optimal solution is distributedly computed in the cluster-tree network.
- The price to pay to compute that optimal solution in terms of signalling is minimized with respect to existing techniques.

Acknowledgements

This work was supported by the Spanish Government Project TEC2008-06305/TEC, the Catalan Government under Grant 2009 SGR 298, and the Chair of Knowledge and Technology Transfer Parc de Recerca UAB—Santander. I.V. is partially funded by Torres Quevedo grant from the Ministerio de Ciencia e Innovación 2008-03-08109. We are also grateful to the HAROSA Knowledge Community (<http://dpcs.uoc.edu/>) and to FP7-IAPP-SWAP EU contract number 251557 that partially funded our research.

References and Notes

1. IEEE. *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*; IEEE Std 802.15.4; IEEE: New York, NY, USA, 2006.
2. Koubaa, A.; Alves, M.; Tovar, E. *IEEE 802.15.4 for Wireless Sensor Networks: A Technical Overview*; IPP-HURRAY Technical Report; HURRAY-TR-050702; Polytechnic Institute of Porto: Porto, Portugal, 2005.
3. Chen, F.; German, R.; Dressler, F. Towards IEEE 802.15.4e: A Study of Performance Aspects. In *Proceedings of 8th IEEE PERCOM*, Mannheim, Germany, 29 March–2 April 2010.
4. Kim, A.; Hekland, F.; Petersen, S.; Doyle, P. When HART Goes Wireless: Understanding and Implementing the WirelessHART Standard. In *Proceedings of IEEE ETFA*, Hamburg, Germany, 15–18 September 2008.
5. ISA. *Wireless Systems for Industrial Automation: Process Control and Related Applications*; ISA: Research Triangle Park, NC, USA, 2009.

6. Mishra, A.; Chewoo, N.; Rosenburgh, D. On Scheduling Guaranteed Time Slots for Time Sensitive Transactions in IEEE 802.15.4 Networks. In *Proceedings of IEEE MILCOM*, Orlando, FL, USA, 29–31 October 2007.
7. Chitnis, M.; Pagano, P.; Lipari, G.; Liang, Y. A Survey on Bandwidth Resource Allocation and Scheduling in Wireless Sensor Networks. In *Proceedings of IEEE International Conference on Network-Based Information Systems*, Indianapolis, IN, USA, 19–21 August 2009.
8. IEEE 802.15 WPAN™ Task Group 4b. Available online: <http://grouper.ieee.org/groups/802/15/pub/TG4b.html> (accessed on 21 March 2011).
9. Zigbee-Alliance. ZigBee Specification. Available online: <http://www.zigbee.org> (accessed on 21 March 2011).
10. Koubaa, A.; Alves, M.; Attia, M.; Van Nieuwenhuysse, A. Collision-Free Beacon Scheduling Mechanisms for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks. In *Proceedings of 7th International Workshop on Applications and Services in Wireless Networks*; Santander, Spain, 24–25 May 2007.
11. Morell, A.; Seco-Granados, G.; Vicario, J. Fair Adaptive Bandwidth and Subchannel Allocation in the WiMAX Uplink. *EURASIP J. Wireless Communications and Networking* **2009**, doi:10.1155/2009/918261.
12. Morell, A. *A Convex Decomposition Perspective on Dynamic Bandwidth Allocation and Applications*; Ph.D. Thesis, 2008. Available online: <http://spcomnav.uab.es> (accessed on 21 March 2011).
13. The ROLL Design Team. RPL: Routing Protocol for Low Power and Lossy Networks (work in progress), 2010. Available online: <http://tools.ietf.org/html/draft-ietf-roll-rpl-17> (accessed on 21 March 2011)
14. Gnawali, O.; Fonseca, R.; Jamieson, K.; Moss, D.; Levis, P. The Collection Tree Protocol (CTP). In *Proceedings of 7th ACM Conference on Embedded Networked Sensor Systems (SenSys)*; Berkeley, CA, USA, 4–6 November 2009.
15. Vilajosana, X.; Llosa, J.; Pacho, J.; Vilajosana, I.; Juan, A.A.; Vicario, J.; Morell, A. ZERO: Probabilistic Routing for Deploy and Forget Wireless Sensor Networks. *Sensors* **2010**, *10*, 8920-8937.
16. Notation: \preceq , \prec , \succeq and \succ stand for component-wise inequalities.
17. Bertsekas, D. *Nonlinear Programming*; Athena Scientific: Belmont, MA, USA, 1999.
18. Boyd, L.; Vandenberghe, S. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
19. Kelly, F. Charging and Rate Control for Elastic Traffic. *Eur. Trans. Telecommun* **1997**, *8*, 33-37.
20. Kelly, F.; Maulloo, A.; Tan, D. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *J. Oper. Res. Soc.* **1998**, *49*, 237-252.
21. Mo, J.; Walrand, J. Fair End-to-End Window-based Congestion Control. *IEEE/ACM Trans. Netw.* **2000**, *8*, 556-567.

22. Palomar, D.; Chiang, M. Alternative Decompositions for Distributed Maximization of Network Utility: Framework and Applications. *IEEE Trans. Autom. Control* **2007**, *52*, 2254-2269.
23. Xiao, L.; Johansson, M.; Boyd, S. Simultaneous Routing and Resource Allocation via Dual Decomposition. *IEEE Trans. Commun.* **2004**, *52*, 1136-1144.
24. Lee, J.; Chiang, M.; Calderbank, A. Network Utility Maximization and Price-Based Distributed Algorithms for Rate-Reliability Tradeoff. In *Proceedings of IEEE Infocom*, Barcelona, Spain, 23–29 April 2006.
25. Tseng, P.; Bertsekas, D. On the Convergence of the Exponential Multipliers Method for Convex Programming. *Math. Program.* **1993**, *60*, 1-19.
26. Polyak, R. Primal-Dual Exterior Point Method for Convex Optimization. *Optim. Method. Softw.* **2008**, *23*, 141-160.
27. Holmberg, K. Primal and Dual Decomposition as Organizational Design: Price and/or Resource Directive Decomposition. In *Design Models for Hierarchical Organizations: Computation, Information, and Decentralization*; Kluwer Academic Publishers: Boston, MA, USA, 1995; pp. 61-92.
28. Nedic, A.; Ozdaglar, A. Subgradient Methods for Saddle-Point Problems. *JOTA* **2009**, *142*, 205-228.
29. The vector \mathbf{s} is a subgradient of the function $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}$ at $\mathbf{x} \in \mathbb{R}^n$ if $\mathbf{f}(\mathbf{y}) \geq \mathbf{f}(\mathbf{x}) + (\mathbf{y} - \mathbf{x})^T \mathbf{s}$, $\forall \mathbf{y} \in \mathbb{R}^n$. If \mathbf{f} is differentiable at \mathbf{x} , the subgradient \mathbf{s} and the gradient $\nabla \mathbf{f}(\mathbf{x})$ coincide. Otherwise, there exist many subgradients.
30. Notation: δ_j is a zero-valued column vector with its j -th entry equal to 1.
31. Note that in the case $[\mathbf{A}]_2 \hat{\mathbf{y}} = c_2$, Problem (14) is splitted into 2 independent projections: (i) distribute c_2 between sensors 3 and 4 and (ii) distribute $c_1 - c_2$ between sensors 1 and 2.
32. We say that a constraint is active when it is attained with equality.
33. Koubaa, A.; Alves, M.; Tovar, E. GTS Allocation Analysis in IEEE 802.15.4 for Real-Time Wireless Sensor Networks. In *Proceedings of Parallel and Distributed Processing Symposium (IPDPS)*, Rhodes Island, Greece, 25–29 April 2006.
34. Jain, R.; Chiu, D.; Hawe, W. *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems*; Technical Report; DEC TR-301; Digital Equipment Corp.: Littleton, MA, USA, 1984.

Appendix

Proof of Proposition 3

As it is discussed after Problem (21), the step from \mathbf{y}^k to $\hat{\mathbf{y}}^k$ forces all the values in each subset \mathcal{A}_m^k to either increase or decrease their allocation in order to fulfill the corresponding rate constraint. Let us consider the first case, *i.e.*, all the values of \hat{y}_j^k with $j \in \mathcal{A}_m^k$ satisfy $\hat{y}_j^k < y_j^k$. Since $y_j^k \in [m_j, M_j]$, it is straightforward to see that $\hat{y}_j^k < d_j$. However, it is possible that $\hat{y}_j^k \leq m_j \forall j \in \mathcal{A}_m^k$. Notwithstanding, this implies that there is no strictly feasible point in the original problem, which contradicts the initial hypothesis. Note that we need to increase all the values in $\hat{\mathbf{y}}^k$ to reach a strictly feasible point but this violates the rate constraint. Therefore, there is at least one value that accomplishes $\hat{y}_j^k \in (m_j, M_j)$.

Similarly, we can consider the second case, that is, the values of \hat{y}_j^k with $j \in \mathcal{A}_m^k$ are increased in the primal projection. In this occasion, if the resulting point \hat{y}_j^k verifies $\hat{y}_j^k \geq d_j \forall j \in \mathcal{A}_m^k$, it implies that the associated constraint is not really necessary. However, this contradicts our hypothesis that the subset of active constraints at the optimum is known. This concludes the proof of the proposition.

© 2011 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license ([http://creativecommons.org/licenses/by/3.0/.](http://creativecommons.org/licenses/by/3.0/))