# Distributed Algorithm for Uplink Scheduling in WiMAX Networks

Antoni Morell, Gonzalo Seco-Granados and José López Vicario
Universitat Autònoma de Barcelona (UAB)
Telecommunications and System Engineering Department (TES)
e-mail: {Antoni.Morell,Gonzalo.Seco,Jose.Vicario}@uab.cat

***Abstract*— This work proposes an algorithm to perform the resource allocation in the uplink of an IEEE802.16 standard-based system. The approach is valid for Point to Multi-Point (PMP) and also for tree-deployed mesh networks, already defined for the Worldwide Interoperability for Microwave Access (WiMAX). Our solution is based on a proportionally fair distribution of resources and it is formulated using the Network Utility Maximization (NUM) framework. Thanks to convex decomposition techniques, we derive a novel way of solving the NUM problem in a distributed manner. The goal is to attain the global optimal scheduling at the Subscriber Stations (SS) without the need of gathering information at a central node in the network. The results show significant gains in the time required to reach the optimal resource allocation for a given set of demands.**

## I. Introduction

The wireless community has recently directed much attention on a variety of topics related to Worldwide Interoperability for Microwave Access (WiMAX) technologies as a broadband solution. Two different standards are under this commercial nomenclature: the IEEE 802.16 [1], with its extension to mobile scenarios IEEE 802.16e [2], and the ETSI HiperMAN [3]. Operating in the range of 2GHz to 11GHz, WiMAX enables a fast deployment of the network even in remote locations with low coverage of wired technologies, such as the DSL (Digital Subscriber Loop) family. WiMAX extends the widely-used WLAN (Wireless Local Area Network) coverage to tens of kilometers, and thus the interest to use such platform to bring internet access to rural and isolated places.

Focusing on WiMAX network aspects, we distinguish between two possible architectures: point-to-multipoint (PMP) and mesh. In PMP mode, one Base Station (BS) serves a certain amount of Subscriber Stations (SSs) using direct links like in traditional cellular networks, whereas in mesh mode, SSs can be linked directly to the BS or routed through other SSs in the network. Terminals use OFDM/OFDMA (Orthogonal Frequency Division Multiplexing/Multiple Access) in mobile and also in fixed WiMAX, although fixed terminals employ mainly TDM/TDMA (Time Division Multiplexing/Multiple Access) as the access technique. As defined in the standard [1], transmission scheduling in mesh mode can be centralized in the mesh BS or distributed among the network. However, the SSs are always in charge of allocating granted resources among their services. The allocation is the result of a three-way handshake process whereby transmission rights are requested and granted, so it constitutes a Dynamic Assignment Multiple Access (DAMA) policy.

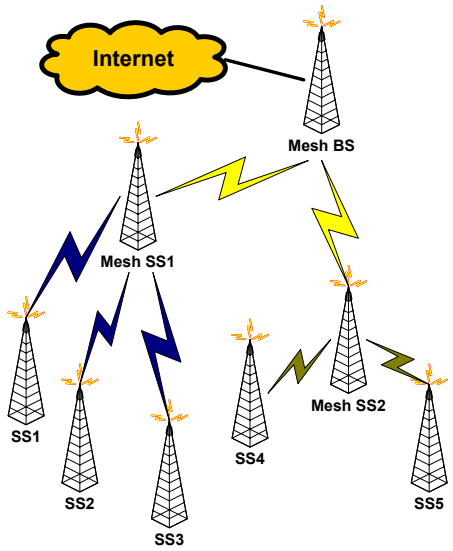


Fig. 1. WiMAX mesh network.

Previous works related to resource allocation in WiMAX networks address a variety of scenarios, from PMP to mesh, from TDMA to OFDMA access types, and distinguish single-channel from multi-channel networks, most of them from a physical (PHY) layer perspective. At the best of our knowledge, two main scheduling approaches are found in the literature, namely: i) formulate the problem in a mathematical optimization framework and ii) develop heuristic algorithms. In the sequel, we review some of the works. In [4], the authors propose an heuristic solution for the case of a single cell OFDMA WiMAX network that maximizes the network sum-rate under some fairness considerations. The authors in [5] analyze how concurrent transmissions boost performance in mesh-type networks by proposing an interference-aware routing and scheduling mechanism. In [6], one can find a discussion about the advantages of a multi-channel network. Finally, [7] contributes with a mathematical optimization solution that falls into the Network Utility Maximization (NUM) framework, where a distributed optimal solution to the established NUM problem is obtained using a convex decomposition approach [8]. It combines PHY and Medium Access Control (MAC) scheduling aspects.

In this paper we concentrate on the scheduling design of the

uplink of a WiMAX network from a MAC layer perspective, i.e. we assume that the actual PHY layer adjustments of the terminals provide fixed averaged capacities in the mid-term. We consider either a PMP or a tree-deployed mesh network; the later being useful for instance when WiMAX is employed as the backhaul network [9]. Our solution can be sorted into the class of proportionally fair schedulers [10] and it is formulated as a NUM problem. The objective is to fairly allocate transmission rates to all the connections or services in the system depending on the mid-term terminal rate defined by the PHY layer set-up. The proposed solution is distributed in the sense that it allows to jointly optimize the entire network without the need of a central node (and subsequent signalling requirements), and provides faster convergence times than other known distributed techniques. A possible network configuration is depicted in Figure 1 with a Mesh Base Station (BS), two Mesh Susbscriber Stations (SSs) and five SSs. We can further assume that each SS has several services that communicate with the BS.

## II. Bandwidth Request and Allocation in the WiMAX Uplink

In WiMAX each SS may support many connections, each one described by a Connection Identifier (CID). There is a primary CID (which is in charge of MAC messaging) and several secondary CIDs, all devoted to different services. All CIDs use a three-way handshake in which they request uplink bandwidth, wait for the BS to compute the allocation and receive their grants in the Uplink (UL) MAP messages. Requests are made in terms of bytes of information and can be incremental (if they add to the previous ones) or aggregate (if they replace them). The way the SSs ask for resources is either using a specific bandwidth-request MAC Packet Data Unit (PDU) or piggybacking on a generic MAC PDU. The UL MAP defines the dedicated or shared UL resources that the SSs can use to emit their bandwidth requests (both types). This mechanism is known as polling in the WiMAX context. If there are enough available resources to poll each SS separately, then we have unicast polling. On the contrary, a subset of terminals or even all terminals enter in a contention process and we have multicast/broadcast polling. Resources are requested and granted in WiMAX per SS and it is the SS that distributes resources among attached CIDs. Therefore, distributed solutions are crucial to perform a joint network optimization.

In order to provide Quality of Service (QoS), WiMAX defines five different scheduling services, namely: i) the unsolicited grant service (UGS), to support real-time service flows, offers fixed-size grants periodically without requiring explicit requests; ii) the real-time polling services (rtPS), to be used in real-time services that generate variable-size packets, provides unicast polling opportunities to the SS; iii) the non-real-time polling services (nrtPS), which is similar to rtPS except that the BS can also use contention-based polling and that unicast polling is made less frequently; iv) the best-effort service (BE), for traffic with non-strict QoS, uses only contention-based polling; and v) the extended real-time polling service (ertPS) is like UGS except that the BS allocates periodical resources that can be used to transmit data or to request additional resources. It is half way between UGS and rtPS to accomodate services whose requirements change in time but not so frequently as with a rtPS. Further details on WiMAX aspects can be found in [11] and references therein.

Let us formulate the scheduling as a NUM problem,

$$\begin{aligned} \max_{\{\boldsymbol{r}_i\}} \quad & \textstyle\sum_{i=1}^{N} U_i(\boldsymbol{r}_i) \\ s.t. \quad & \boldsymbol{r}_i \in \mathcal{R}_i \qquad i = 1 \dots N \\ & \textstyle\sum_{i=1}^{N} h_i(\boldsymbol{r}_i) \leq c \end{aligned} \tag{1}$$

where $U_i(\boldsymbol{r}_i)$ is the utility function perceived at entity $i$ (mesh SS, SS or CID) and depends on the granted rates $\boldsymbol{r}_i$. Note that $U_i(\boldsymbol{r}_i)$ may have an analytical expression or it can be the result of an optimization problem with the same structure of (1). The functions $h_i(\boldsymbol{r}_i)$ are convex on the rates and $c$ is the total amount of available resources. The convex subsets $\mathcal{R}_i$ are cartesian products that define the maximum and minimum rates that each element in $\boldsymbol{r}_i$ can take.

An illustrative example can be derived from the network configuration in Figure 1. Assume that we want to perform a joint and distributed allocation for all the CIDs in the network. First, let us consider the scheduling at the highest level, i.e. within the links Mesh SS1-Mesh BS and Mesh SS2-Mesh BS, and define accordingly $U_1(\boldsymbol{r}_1)$ and $U_2(\boldsymbol{r}_2)$. Note that $\boldsymbol{r_1}$ contains the rates of the links from SSs 1, 2 and 3 to Mesh SS1, i.e. $\boldsymbol{r}_1 = [\boldsymbol{r}_1^{1^T}, \boldsymbol{r}_1^{2^T}, \boldsymbol{r}_1^{3^T}]^T$. Furthermore, each $\boldsymbol{r}_1^j$ contains at its turn the rates from the CIDs attached to the SS$j$ that take the route SS$j$-Mesh SS1-MeshBS, so that $U_1(\boldsymbol{r}_1)$ is a convex optimization problem that models the scheduling in the second level, i.e. from Mesh SS1 to SSs 1 to 3. The parameter $c$ models the total rate amount that the Mesh BS can send to the global network.

In this way, the joint problem is described as the concatenation of several PMP scheduling problems, as Figure 2 shows. Moreover, as we will see, it is only necessary that each node exchanges information with the node above it with the subsequent reduction in signalling with respect to a centralized optimization approach (although it is centralized scheduling). In the results section we propose an example that shows a possible connection between the proposed formulation and the scheduling services in WiMAX using a specific definition of utility functions and feasible allocation subsets.

In the next section, we develop a novel, efficient and distributed optimization algorithm to solve (1) based on convex decomposition techniques.

## III. Coupled Primal-Dual Decompositions Method

Let us consider again the problem in (1). It has optimization variables $\{\boldsymbol{r}_i\}$ and objective function $F = \sum_{i=1}^{N} U_i(\boldsymbol{r}_i)$. Each group of variables $\boldsymbol{r}_i$ is restricted to lie on the convex set defined in $\mathcal{R}_i$. If there were no additional constraints, the problem could be solved separately for each group of variables $\boldsymbol{r}_i$ as it is fully decoupled. However, there exists a coupling

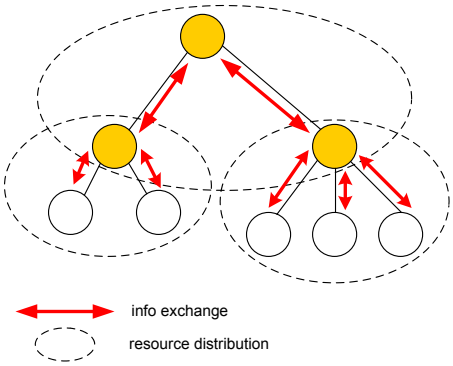


Fig. 2. Proposed distributed bandwidth allocation.

constraint that sums all the functions $h_i(\boldsymbol{r}_i)$. In the sequel, we will talk indistinctly about the minimization of a convex objective function or the maximization of a concave one as they are equivalent problems [12]. However, we remark that in the NUM context, the maximization of the utility function is usually employed since it is intuitively related to the operation of the network. There are mainly two formalized procedures to take advantage of the semi-decoupled nature of the problem, namely primal and dual decompositions. We first review these two procedures before presenting our proposal.

### A. Primal Decomposition

To understand the basics of primal decomposition, let us rewrite the problem in (1) for a fixed link capacity as

$$\begin{aligned} \max_{\{\boldsymbol{r}_i, y_i\}} \quad & \textstyle\sum_{i=1}^{N} U_i(\boldsymbol{r}_i) \\ s.t. \quad & \boldsymbol{r}_i \in \mathcal{R}_i \qquad i = 1 \dots N \\ & \boldsymbol{h}_i(\boldsymbol{r}_i) \leq y_i \\ & \textstyle\sum_{i=1}^{N} y_i \leq c \end{aligned} \tag{2}$$

Clearly, fixing the values of the variables $y_i$ fully decouples the main problem. In other words, knowing the optimal values of $y_i$ reduces the resolution of the main problem to the resolution of $N$ smaller problems in the variables $\boldsymbol{r}_i$. The problem can be interpreted in the following manner,

$$\begin{aligned} \max_{\{y_i\}} \quad & \textstyle\sum_{i=1}^{N} U_i^P(y_i) \\ s.t. \quad & \textstyle\sum_{i=1}^{N} y_i \leq c, \end{aligned} \tag{3}$$

where the functions $U_i^P(y_i)$ are defined as

$$\begin{aligned} U_i^P(y_i) = \quad & \max_{\{\boldsymbol{r}_i\}} \qquad U_i(\boldsymbol{r}_i) \\ & \boldsymbol{r}_i \in \mathcal{R}_i \\ & h_i(\boldsymbol{r}_i) \leq y_i \end{aligned} \tag{4}$$

Problem (3) is usually referred as the primal master problem, while (4) are known as the primal subproblems. One possible way to numerically solve the primal master problem is using the projected subgradient method. The idea of the method is quite intuitive. It basically updates the values of $\{y_i\}$ towards the opposite direction to the subgradient of (3) and projects these values to the half-space defined by the second line of (3) in the case the previously updated values are unfeasible. Details about projections into subsets can be found in [12]. The resulting update equation is

$$y_i^{t+1} = \left[ y_i^t + \alpha(t) \cdot s_i(y_i^t) \right]^P, \tag{5}$$

where $\alpha(t)$ is the adaptation step-size, $s_i(y_i^t)$ stands for the subgradient of $U_i^P$ at the point $y_i = y_i^t$ and $[\cdot]^P$ refers to the projection on the feasible set. The superscript $t$ indicates the iteration number.

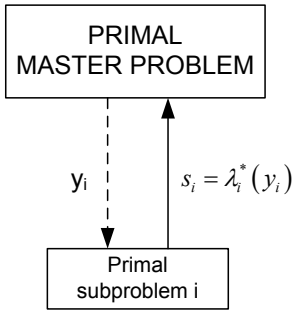


Fig. 3. System view of primal decomposition.

The subgradient of a function can be conceptually interpreted as the gradient. The question is how to find a gradient of the subproblems $U_i^P$, which are defined as convex optimization problems. In this case, we resort to [13, Sec. 5.4.4] and use the subgradient as a generalization of the gradient of a function. The strenght of the technique is that a subgradient is directly given by the Lagrange multipliers associated to the coupling constraint $h_i(\boldsymbol{r}_i) \leq y_i$ in (4), [14]. Later on, this Lagrange multiplier is referred to as $\lambda_i$, and its optimal value is referred to as $\lambda_i^*(y_i)$ for a given $y_i$. For further details on the projected gradient method, the step size and the subgradients, please refer to [8], [13] and [14].

The logical procedure of a primal decomposition algorithm is as follows: the master subproblem sends the $y_i$ values to the subproblems. These compute the associated subgradients and return these values to the master problem. Now, the master updates the $y_i$'s. A system view of a primal decomposition can be found in Figure 3.

### B. Dual Decomposition

Consider now dual decomposition, which decomposes the dual function of the original problem (1). Construct the Lagrangian of (1) relaxing only the coupling constraint as

$$L(\{\boldsymbol{r}_i\, y_i\}, \mu) = -\sum_{i=1}^{N} U_i(\boldsymbol{r}_i) + \mu^T \left(\sum_{i=1}^{N} h_i(\boldsymbol{r}_i) - c\right) \tag{6}$$

The minimization of the Lagrangian with respect to the primal variables results in the dual function, which is a concave function of the dual variables. As the constraints $\boldsymbol{r}_i \in \mathcal{R}_i$ have not been relaxed, the dual function is

$$g(\mu) = \left( \sum_{i=1}^{N} \min_{\substack{\{\boldsymbol{r}_i\} \\ \boldsymbol{r}_i \in \mathcal{R}_i}} \left(-U_i(\boldsymbol{r}_i) + \mu^T h_i(\boldsymbol{r}_i)\right) \right) - \mu^T c \tag{7}$$

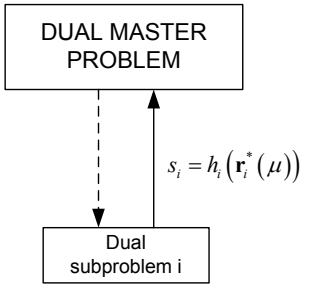


Fig. 4. System view of dual decomposition.

The optimal value for the dual variable $\mu$ is the one that maximizes the dual function [12]. Note that the problem in (7) can be expressed as

$$g(\mu) = \sum_{i=1}^{N} U_i^D(\mu) - \mu^T c, \tag{8}$$

where now

$$U_i^D(\mu) = \min_{\substack{\{\boldsymbol{r}_i\} \\ \boldsymbol{r}_i \in \mathcal{R}_i}} -U_i(\boldsymbol{r}_i) + \mu^T h_i(\boldsymbol{r}_i) \tag{9}$$

are the dual subproblems. The subgradient concept applies also in this case. We have as subgradient $s_i = h_i(\boldsymbol{r}_i^*)$, being $\boldsymbol{r}_i^*(\mu)$ the optimal value of the primal variables in subproblem (9) for a given value of $\mu$ [13, Sec. 6.1].

Finally, to solve the dual problem, (8) must be maximized with the constraint that $\mu \geq 0$. This is often called the dual master problem:

$$\begin{aligned} \max_{\mu} \quad & g(\mu) \\ s.t. \quad & \mu \geq 0 \end{aligned} \tag{10}$$

The dual master problem can also be solved using the projected subgradient method. Note that the projection into the feasible set is easier than in the primal decomposition as we only have to set $\mu$ to 0 when a negative value is computed. The $\mu$ updates are

$$\mu^{t+1} = \left[\mu^t + \alpha(t) \cdot (\sum_{i=1}^{N} h_i(\boldsymbol{r}_i^*(\mu^t)) - c)\right]^+ \tag{11}$$

where $[\cdot]^+$ stands for the aforementioned projection into the non-negative orthant.

The dual decomposition is the decomposition technique most used in the literature. From a system-level point of view, it resembles to the primal decomposition one (see Figure 4).

The major advantage of using a decomposition technique is that distributed solutions may be naturally obtained, which sometimes is required by some problems. For example, in the NUM context, dual decomposition techniques obtain fully distributed solutions. On the contrary and generally speaking, the main disadvantage is the slow speed of convergence of the resulting algorithms, mainly due to the projected subgradient approach. Moreover, speed of convergence depends on the step-size parameter $\alpha(t)$, which must be tuned by the user.

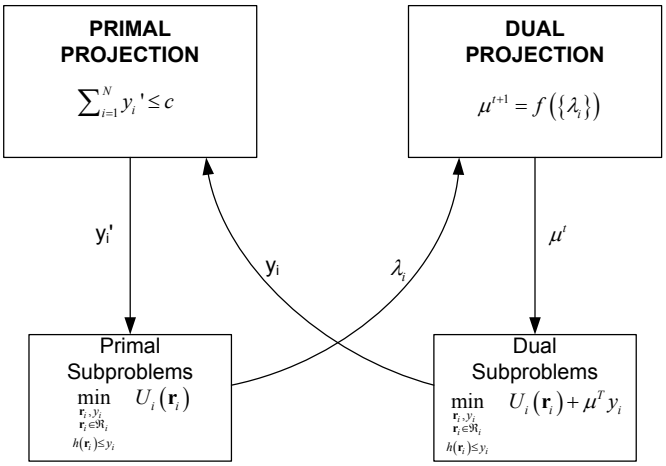


Fig. 5. System view of the Primal-Dual Decompositions method.

### C. Coupled Primal-Dual Decompositions

In the light of the previous results, we observe that both approaches manage complementary information. We formulate then the following question: is it possible to find out an hybrid technique with advantages over the single approaches?

The answers is yes. The observation that the dual of the primal subproblem (4) is the dual subproblem (9) is the key to do that. The basic idea behind the proposed method is hence to couple the primal and dual decompositions, so they are iterated in the following way: Dual Master → Dual Subproblems → Primal Master → Primal Subproblems, and so on (see Figure 5). Note, however, that the procedure is not as immediate as combining both decompositions since modifications, specially in the Dual Master, are needed. This is the reason why we will introduce the concept of Dual Projection. In the Primal Master problem, the updating towards the subgradient is no longer used and only the projection on the feasible set remains. This approach will result in faster convergence and will allow us to avoid both the gradient method and the choice of an adaptation step-size. It is possible to find out in the literature other uses of combined primal and dual approaches based on the algorithm so-called cross-decomposition [15]. Our solution goes in a different direction as the proposed interactions between primal and dual versions of the problem are constructed in a different way. As a result, we find that in [15] both primal and dual variables have to be updated by averaging new candidates with old ones, whereas our method uses only instant updates without averaging. In the simulation section, we show how this affects both strategies in a practical example.

The algorithm can be divided into two conceptually different parts, namely *proposal of candidates* and *correction*. The proposal of candidates is the task done by both the primal and dual subproblems. There are two correction steps; they replace the master problems in the primal and dual standard decompositions. The correction steps are in charge of adjusting the proposed candidate solutions according to the primal or dual feasible sets. These steps are interpreted as projections into the corresponding feasible sets in a wide sense.

Let us describe at high level a complete iteration of the proposed method for the problem under consideration. Let us start with an initial value of $\mu$, called $\mu^t$, which is passed to the dual subproblems. Using that value, the subproblems make their particular guess for the primal coupling variables $\{y_i\}$ as $h_i(\boldsymbol{r}_i^*(\mu^t))$. The proposed values may exceed the convex subset defined by $\sum_{i=1}^N y_i \leq c$ and hence be unfeasible. Primal Projection corrects this situation by projecting $\{y_i\}$ into the feasible subset. Thus we obtain the corrected values $\{y_i'\}$, which are given to the primal subproblems. In turn each primal subproblem computes its own candidate ($\lambda_i$) for the dual variable ($\mu$). Similarly to what happens with the primal part of the problem, the solution may be unfeasible from the dual point of view (defined later) and requires correction. The Dual Projection computes this correction as a function of the previous values (either the min or max can be chosen, but once it is chosen the algorithm is pegged to it) and updates the dual coupling constraint, i.e. $\mu^{t+1} = f(\{\lambda_i\})$.

## IV. Method Analysis

In this section we detail the Coupled Primal-Dual Decompositions method shown in Figure 5 and analyze its convergence.

### A. Detailed Description of the Method

Let us consider again the problem in (1), where variables $\boldsymbol{r}_i$ are end-user rates, and primal variables $y_i$ constraint these by means of any convex function $h_i(\boldsymbol{r}_i)$. We refer to $\{\boldsymbol{r}_i\}$ as non-coupling variables and to $\{y_i\}$ as coupling variables. The dual variables associated to $h_i(\boldsymbol{r}_i) \leq y_i$, i.e. $\lambda_i$, are called non-coupling dual variables, while $\mu$ is the dual coupling variable associated to the coupling constraint $\sum_{i=1}^N y_i \leq c$. The motivation for this nomenclature is that without $\sum_{i=1}^N y_i \leq c$, the problem is only constrained by the local subsets $\mathcal{R}_i$, so it becomes a set of non-coupled problems.

The basics of the method have been already introduced in the previous section with Figure 5. Let us now detail each of the building blocks of the algorithm in the order they are executed.

First, the dual subproblems compute their group bandwidth allocation candidates $y_i$ depending on the value of $\mu^t$ as

$$\boldsymbol{r}_i^*(\mu^t) = \arg_{\boldsymbol{r}_i} \min_{\substack{\boldsymbol{r}_i \\ \boldsymbol{r}_i \in \mathcal{R}_i \\ y_i = h_i(\boldsymbol{r}_i)}} -U_i(\boldsymbol{r}_i) + \mu^t \cdot y_i \tag{12}$$

Note that in the optimal solution $y_i = h_i(\boldsymbol{r}_i)$.

We assume that the coupling constraint is active, otherwise the problem is not coupled and readily solved. Next, the Primal Projection updates the $y_i$ values to $y_i'$ and ensures feasibility, i.e. $\sum_{i=1}^N y_i' \leq c$. The projection into $\sum_{i=1}^N y_i' = c$ can be analytically computed as the point in the surface that minimizes the Euclidean distance to the point $\boldsymbol{y} = [y_1, \ldots, y_N]^T$:

$$y_i' = y_i - \frac{\sum_{i=1}^N y_i - c}{N} \tag{13}$$

The updated primal variables $y_i'$ are feed to the primal subproblems to obtain the dual variables $\lambda_i$ as

$$\lambda_i(y_i') = \arg_{\lambda_i} \max_{\lambda_i \geq 0} \min_{\substack{\boldsymbol{r}_i \\ \boldsymbol{r}_i \in \mathcal{R}_i}} -U_i(\boldsymbol{r}_i) + \lambda_i[h_i(\boldsymbol{r}_i) - y_i'] \tag{14}$$

Note that solving the minimization problem inside (14) implies obtaining the primal variables $\boldsymbol{r}_i$ and also the dual variable $\lambda_i$, associated to the constraint $h_i(\boldsymbol{r}_i) \leq y_i'$.

Finally, in the Dual Projection we get $\mu^{t+1}$ as a function of the candidate values $\lambda_i$. We can choose between the minimum or the maximum to compute the iterations (once the algorithm starts, it must not be changed) and therefore

$$\mu^{t+1} = f(\{\lambda_i|_{act}\}) = \begin{cases} \min(\{\lambda_i|_{act}\}) \\ \\ \max(\{\lambda_i|_{act}\}) \end{cases} \tag{15}$$

where $\{\lambda_i|_{act}\}$ defines the subset of the $\{\lambda_i\}$ values that are active. A multiplier $\lambda_i$ is defined as active when eliminating the related constraint $h_i(\boldsymbol{x}_i) \leq y_i'$ in (14) changes the solution of the aforementioned problem. In the following, we prove the convergence of the algorithm.

### B. Convergence analysis

It is assumed in the rest of the section that the optimal solution to the problem in (1) is unique, which holds for most of the convex problems that are of interest in engineering.

Until this point we have seen the motivation of the proposed method and also the role of most of the building blocks, namely dual and primal subproblems and Primal Projection. We want to show now the role of the Dual Projection. For that purpose we use the KKT conditions [12]. First, let us construct the Lagrangian of (2)

$$\begin{aligned} L(\{\boldsymbol{r}_i, y_i, \lambda_i\}, \mu) &= \sum_{i=1}^N -U_i(\boldsymbol{r}_i) \\ &+ \sum_{i=1}^N \lambda_i(h_i(\boldsymbol{r}_i) - y_i) \\ &+ \sum_{i=1}^N \sum_j \gamma_i^j g_i^j(\boldsymbol{r}_i) + \mu(\sum_{i=1}^N y_i - c) \end{aligned} \tag{16}$$

where we have relaxed all explicit and implicit constraints and the arbitrary number of convex functions $\{g_i^j\}$ defines the convex set $\mathcal{R}_i$. Among the KKT conditions for optimality of the solution, we are interested in the conditions that require

$$\frac{\partial L}{\partial y_i} = 0 = \mu - \lambda_i \tag{17}$$

which force the solution to fulfill

$$\mu = \lambda_1 = \ldots = \lambda_N \tag{18}$$

Therefore, $\mu$ must be chosen from the candidates $\lambda_i$, as in the optimal solution all $\lambda_i$ and $\mu$ must be equal. This is the role of the Dual Projection in (15).

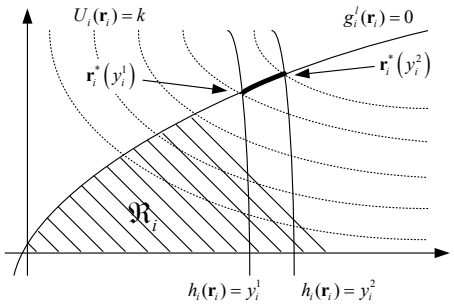


Fig. 6. Study of the subproblems.

The following Lemma is needed as an intermediate step to prove the convergence.

*Lemma 1:* Assuming that $h_i(\boldsymbol{r}_i) \leq y_i$ is active in the interval $y_i \in (y_i^1, y_i^2)$, the dual variable $\lambda_i$ is a decreasing function of the primal variable $y_i$ in the subproblem

$$U_i^{P'}(y_i) = \begin{array}{c} \min \\ \boldsymbol{r}_i \\ \boldsymbol{r}_i \in \mathcal{R}_i \\ h_i(\boldsymbol{r}_i) \leq y_i \end{array} \quad -U_i(\boldsymbol{r}_i) \tag{19}$$

*Proof:* Consider the following modification of the minimization problem in (19). Assume that $U_i(\boldsymbol{r}_i)$ is evaluated only in the curve described by $\boldsymbol{r}_i^*(t_i)$ with $t_i \in [y_i^1, y_i^2]$, where $\boldsymbol{r}_i^*(t_i)$ is defined as the optimal value of $\boldsymbol{r}_i$ in (19) when $y_i = t_i$. The result is the one-dimensional function $U_i^P(t_i)$. To fix concepts, see Figure 6. In dashed lines, we plot the contour plot of the objective function, i.e. $-U_i(\boldsymbol{r}_i) = k$ ($k$ is an arbitrary constant) and in solid lines the contour plots of both the coupling constraint and a given local constraint in $\boldsymbol{r}_i \in \mathcal{R}_i$ (whose expression is $g_i^l(\boldsymbol{r}_i) \leq 0$, assuming that there is a single $g_i^l(\boldsymbol{r}_i)$ for simplicity). The darkest curve corresponds to $\boldsymbol{r}_i^*(t_i)$.
Under this modification, the problem in (19) turns into

$$\begin{array}{ll} \min\limits_{t_i} & -U_i(\boldsymbol{r}_i^*(t_i)) \\ s.t. & h_i(\boldsymbol{r}_i^*(t_i)) \leq y_i \end{array} \tag{20}$$

Note that constraints in $\boldsymbol{r}_i \in \mathcal{R}_i$ are no longer necessary as they are included in $\boldsymbol{r}_i^*(t_i)$. Furthermore, $-U_i(\boldsymbol{r}_i^*(t_i))$ is guaranteed to be convex as it is the minimization of a function of variables $(\boldsymbol{r}_i, y_i)$ over $\boldsymbol{r}_i$ in a convex set [12, Section 3.2.5]. And finally, as we assume $h_i(\boldsymbol{r}_i) \leq y_i$ to be active, we have

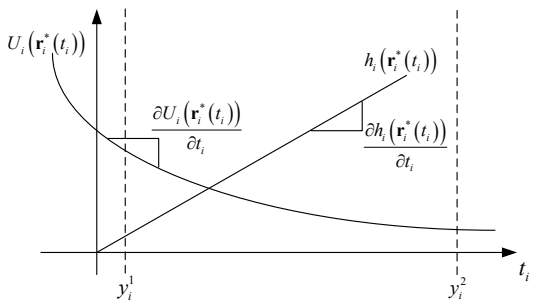


Fig. 7. Study of the subproblems (1-D interpretation).

that $h_i(\boldsymbol{r}_i^*(t_i)) = t_i$.
KKT conditions applied to (20) state that

$$\frac{\partial L(t_i, \lambda_i)}{\partial t_i} = \frac{\partial - U_i(\boldsymbol{x}_i^*(t_i))}{\partial t_i} + \lambda_i \cdot \frac{\partial h_i(\boldsymbol{r}_i^*(t_i))}{\partial t_i} = 0 \tag{21}$$

and therefore, it holds that

$$-\frac{\partial U_i(\boldsymbol{r}_i^*(t_i))}{\partial t_i} = -\lambda_i \tag{22}$$

Finally, concavity of $U_i(\boldsymbol{r}_i^*(t_i))$ assures that $\frac{-\partial U_i(\boldsymbol{r}_i^*(t_i))}{\partial t_i}$ is an increasing function of $t_i$, and therefore $\lambda_i$ is a decreasing function of $t_i$. We conclude this proof by noting that by definition the optimal solution of (20) is attained at $t_i = y_i$. Figure 7 plots the graphical representation of the one-dimensional problem discussed in this proof. ■

*Corollary 1:* Using similar arguments, it can be proved that in the dual subproblem $y_i = h_i(\boldsymbol{r}_i^*(\mu))$ is a decreasing function of $\mu$.

Once we have studied the subproblems and the relations that exist between dual and primal coupling variables, we are ready to outline the proof of the convergence of the algorithm, which studies the convergence of $\mu^t$ to its optimal value, i.e. $\mu^t \overset{t\to\infty}{\longrightarrow} \mu^*$. Since the problem is convex, finding the optimal values of the dual variables implies finding the optimal values for the primal ones.

Let us study two cases, namely:

1) $\mu^t > \mu^*$
2) $\mu^t < \mu^*$

Assume now that $\mu^t > \mu^*$. Then, after the application of the dual subproblems and corollary 1, it holds that

$$y_i \leq y_i^*, \qquad i = 1, \ldots, N \tag{23}$$

This result assumes that $h_i(\boldsymbol{r}_i) = y_i$ for all subproblems, which is imposed by the following KKT optimality condition

$$\lambda_i \cdot (h_i(\boldsymbol{r}_i) - y_i) = 0, \quad i = 1 \ldots N \tag{24}$$

together with the result in (18).

In the Primal Projection, a certain quantity $k$ is added to the obtained $y_i$ values, so that

$$y_i' = y_i + k, \qquad s.t. \sum_{i=1}^{N} y_i' = c \tag{25}$$

and therefore, some $y_i' \leq y_i^*$ and some $y_i' > y_i^*$; otherwise $y_i'$ would be the optimal solution.

Applying $\{y_i'\}$ to the primal subproblems, we obtain the dual variables $\lambda_i$, which may be interpreted as candidates for $\mu^*$. Resorting now to Lemma 1, it holds that there exist some $\lambda_i \leq \mu^*$ and some $\lambda_i > \mu^*$.

Finally, the key point is the Dual Projection,

$$\mu^{t+1} = f(\{\lambda_i|_{act}\}) \tag{26}$$

Let us discuss a detail here. The $\lambda_i$ values associated with $y_i$ values that do not really constraint the solution must not be taken into account because the solution is actually constrained by the local constraints, i.e. $\boldsymbol{r}_i \in \mathcal{R}_i$. Note that, in terms

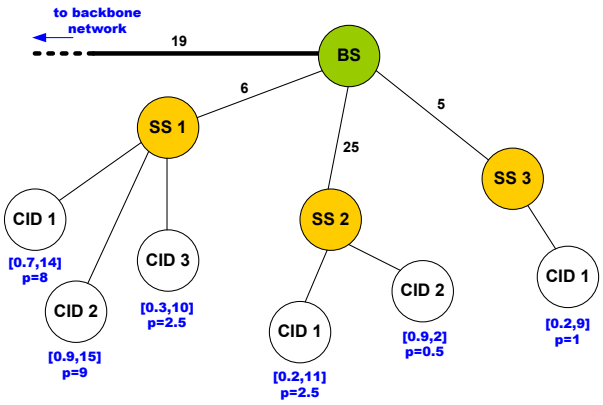


Fig. 8. Network example under test.

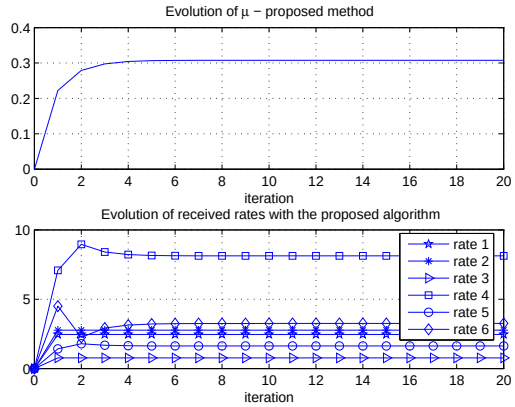


Fig. 9. Evolution of rates and dual variable $\mu$ with the proposed method.

of KKT conditions, the value of $\lambda_i$ is not significant because whatever this value is, an adequate Lagrange multiplier for the local constraints can be chosen to maintain all of the optimality conditions and hence, the optimal solution.

Continuing with the convergence proof, if $\mu^{t+1} = \max(\lambda_i^{act})$, it holds that $\mu^{t+1} \geq \mu^*$. Furthermore, since $y_i' \geq y_i$, it is also true that $\mu^{t+1} \leq \mu^t$. Therefore, we can summarize that

$$\mu^* \leq \mu^{t+1} \leq \mu^t \tag{27}$$

where $\mu^t$ can not tend to any value $\mu' \neq \mu^*$ as the optimal solution is unique. In the case where $\mu^t < \mu^*$, the same results hold if we choose $\mu^{t+1} = \min(\lambda_i|_{act})$ and the proof is similar. In the general case, any of the two functions is valid. If $\mu^0 > \mu^*$ and we choose $\mu^{t+1} = \min(\lambda_i|_{act})$, then $\mu^1 < \mu^*$ and we take up again one of the previous cases.

In the next section, we present some results comparing the performance of the proposed algorithm to other solutions in the literature.

## V. Results

Consider the PMP network example in Figure 8, with three SSs and six CIDs that manage different services. The links are labelled with their maximum rate, which is determined by the PHY-layer mode used in each one. Two scheduling levels can be identified, namely: i) from the SSs to the BS and ii) from the CIDs to the corresponding SS. At the highest level, we compute (1) with

$$U_i(c_i) = \left\{ \begin{array}{ll} \max\limits_{\{r_i^j\}} & \sum_j U_i^j(r_i^j) \\ s.t. & \sum_j r_i^j \leq c_i \end{array} \right\}, \quad h_i(\boldsymbol{r}_i) = \sum_j r_i^j, \tag{28}$$

where $r_i^j$ is the transmission rate of CID$j$ at SS$i$ and $c_i$ is the rate capacity from SS$i$ to the BS. At the lowest level, problem (1) is solved for the $i^{th}$ SS using

$$U_i^j = p_i^j \log r_i^j, \qquad h_i^j(r_i^j) = r_i^j. \tag{29}$$

In both cases, the subsets $\mathcal{R}_i$ and $\mathcal{R}_i^j$ contain the maximum and minimum rate values of the CIDs within them. However, note that at the highest level and from a practical point of view, we need to know only the sum of maximums and minimums since these two values suffice to obtain the primal variables $\{y_i\}$ and the dual variable $\mu$ of the proposed algorithm. The specific quantities per CID, i.e. $\mathcal{R}_i^j$, are only required at the lowest level to obtain the scheduling of CIDs.

Intuitively, the proposed method tries to find a consensus on the value of the dual variable $\mu$ across the entire network, often interpreted as the price to be paid for the resources (rates in our case). However, some subsets of terminals (in certain PMP sub-pieces) may remove from the negotiation if their local constraints make the current global $\mu$ value not feasible therein. In this case, those zones in the network negotiate their particular consensus price, which is different from $\mu$.

The election of logarithmic functions of the rates responds to a proportional fair criterion as it is discussed in [10], but other utility functions can be used. We further use the priority values $p_i^j$ to balance the scheduling towards some services depending on the specific QoS policy and thus the solution is asymmetric proportionally fair. These values are depicted in blue in Figure 8 at each CID. The max and min values in $\mathcal{R}_i^j$ (in brackets in the figure) define the requested and minimum granted rates of each service, respectively. For example, with UGS one can map the request to the minimum guaranteed rate in our model (which is always assigned) whereas the ertPS can be configured granting part of the requested rate as in UGS and competing for the remaining part (prioritized with $p_i^j$). The original requests in bytes of information can be transformed to rates taking into account the time basis of such requests.

We assess now the convergence speed terms of three different solutions, namely: i) a two-level dual decomposition approach [8], ii) a mean value cross-decomposition approach [15] and iii) the proposed technique.

The results of the proposed method are depicted in Figure 9. The first subplot contains the evolution of the dual variable at the highest allocation level and the second subplot shows the evolution of the allocated rates at the CIDs (rates are ordered from left to right according to the CIDs in Figure 8). The same results with a two-level dual decomposition approach are plotted in Figure 10. Dual or primal decompositions require a user-defined adaptation step and in this case we choose a

diminishing step size of the form $\alpha(t) = \frac{\alpha_0}{\sqrt{t}}$ with $\alpha_0 = 0.5$. Note that the proposed method does not require the choice of any parameter. In both cases, at each iteration at the highest level, it is required to attain the solution at each CID at the lowest level, which enforces different updating rates. Therefore, a fast convergence of the lower level is more necessary as the tree size grows. In the light of results, it is clear that our algorithm converges with a number of iterations orders of magnitude lower.

For the sake of completeness, we compare our method with the Mean Value Cross (MVC) decomposition method, which is described in [15]. It is not distributed but uses also the idea of combining primal and dual decompositions of the problem in a single approach. The evolution of the rates at the CIDs is plotted in Figure 11 and once more, the proposed method converges to the optimal solution much faster.

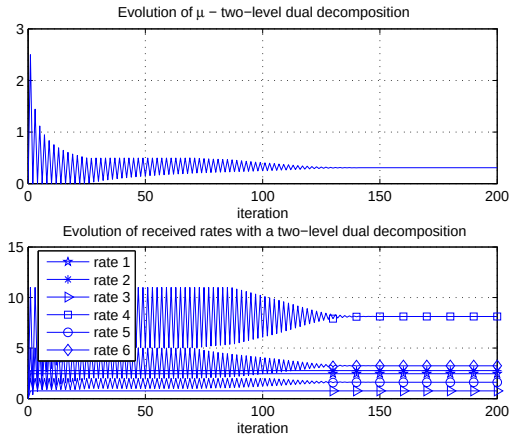


Fig. 10. Evolution of rates and dual variable $\mu$ using a two-level dual decomposition approach.

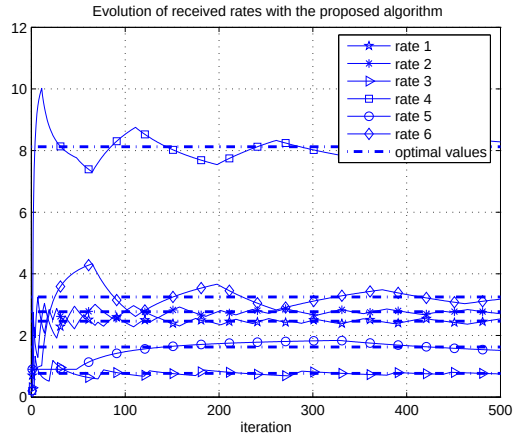


Fig. 11. Evolution of rates using a mean value decomposition approach.

## VI. Conclusions

In this paper we have derived a novel decomposition method, the coupled primal-dual decompositions, with direct application to the problem of bandwidth allocation or flow control in the uplink of WiMAX PMP or tree-deployed mesh networks, the later being suitable for instance for backhaul. As a result, the global solution is computed exchanging information only locally (inside each PMP subpart) with promising results in terms of iterations required to converge.

The problem is formulated as a NUM problem with a proportional fairness criterion and thanks to the proposed decomposition, the optimal solution is computed in a distributed manner, as enforced by the standard (each BS schedules its CIDs with the granted resources). The whole network optimization is broken into several PMP scheduling levels in a top-bottom design, where each terminal interchanges the resource allocation and the Lagrange multiplier (often interpreted as the price to be paid for the resource) with the node above. The process is repeated until a consensus is found.

The convergence of the method has been proved and simulation results show that it converges faster than other approaches in the literature. The issue is specially relevant as the number of scheduling levels grow. Additional advantages are provided by the NUM framework, since an adequate selection of the utility functions used allows us to reach fair solutions or to sustain QoS definitions.

## References


[1] IEEE, "Air Interface for Fixed Broadband Wireless Access Systems," *IEEE Standards*, Oct 2004.

[2] IEEE, "Air Interface for Fixed and Mobile Broadband Wireless Access Systems; Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Band and Corrigendum 1," *IEEE Standards*, Feb 2006.

[3] ETSI, "Broadband Radio Access Networks (BRAN); HIPERMAN; Data Link Control (DLC) Layer.," *ETSI TS 102 178*, Mar 2003.

[4] B. Makarevitch, "Adaptive Resource Allocation for WiMax," *in proc. IEEE Int. Symp. on Personal, Indoor and Mobile Radio Comm. (PIMRC), Athens (Greece)*, Sep 2007.

[5] H.-Y. Wei, S. Ganguly, R. Izmailov and Z.J. Hass, "Interference-Aware IEEE 802.16 WiMax Mesh Networks," *in proc. IEEE Vehicular Tech. Conf (VTC'05 Spring), Stockholm (Sweden)*, May 2005.

[6] P. Du, W. Jia, L. Huang and W. Lu, "Centralized Scheduling and Channel Assignment in Multi-Channel Single-Transceiver WiMax Mesh Network," *in proc. IEEE Wireless Comm. and Net. Conf (WCNC), Hong Kong (China)*, Mar 2007.

[7] P. Soldati, B. Johansson and M. Johansson, "Distributed Optimization of End-to-End Rates and Radio Resources in WiMax Single-Carrier Networks," *in proc. IEEE Global Telecomm. Conf. (GLOBECOM), San Francisco (USA)*, Nov 2006.

[8] D.P. Palomar and M. Chiang, "Alternative Decompositions for Distributed Maximization of Network Utility: Framework and Applications," *in IEEE Tran. on Automatic Control, Vol. 52, No. 12, pp. 2254-2269*, Dec 2007.

[9] S. Lee, G. Narlikar, M. Pal, G. Wilfong and L. Zhang, "Admission control for multihop wireless backhaul networks with QoS support," *in proc. IEEE Wireless Comm. and Net. Conf (WCNC), Las Vegas (USA)*, Apr 2006.

[10] F.P. Kelly, A. Maulloo and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of Operations Research Society, Vol. 49, No. 3, pp. 237-252*, Mar 1998.

[11] J.G. Andrews, A. Ghosh, and R. Muhamed, *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*, Prentice-Hall, 2007.

[12] L. Boyd and S. Vandenberghe, *Convex optimization*, Cambridge University Press, 2003.

[13] D. P. Bertsekas, *Nonlinear Programming*, Belmont, MA, USA: Athena Scientific, 1999.

[14] D. P. Bertsekas, A. Nedić and A. E. Ozdaglar, *Convex Analysis and Optimization*, Belmont, MA, USA: Athena Scientific, 2003.

[15] K. Holmberg and K.C. Kiwiel, "Mean Value Cross Decomposition for Nonlinear Convex Problems," *Optimization Methods and Software, Vol. 21, No. 3, pp. 401-417*, Jun 2006.