Computationally Efficient Cross-Layer Algorithm for Fair Dynamic Bandwidth Allocation

Antoni Morell, Gonzalo Seco-Granados and María Ángeles Vázquez-Castro
Universitat Autònoma de Barcelona (UAB)
Telecommunications and System Engineering Department (TES)
e-mail: {Antoni.Morell,Gonzalo.Seco,Angeles.Vazquez}@uab.es

***Abstract*— The problem of Dynamic Bandwidth Allocation (DBA) is inherent to systems that employ Bandwidth on Demand (BoD). An important issue in such systems is to be able to react efficiently to the always-changing traffic requests of users. Moreover, it is realistic to assume large populations sharing system resources and thus efficient methods to distribute bandwidth are mandatory.**

Further desirable system features include guarantees on fairness and on Quality of Service (QoS). Actual trends propose to reach convergence among networks at IP-level. This encourages the design of algorithms that sustain IP-defined QoS (e.g. in DiffServ) and forces to exchange information between layers. We talk then about cross-layer designs.

In this paper, we propose a novel method to compute the allocation accomplishing the previous requirements of fairness, QoS and time efficiency. Our work departs from known results on decomposition techniques (primal and dual) and combines these in a novel, interleaved and coupled fashion. In the dual decomposition technique, the subgradient method is typically used to adpatively compute the price the resource is charging to the users. In our approach, the price is selected taking into account the value that users are willing to pay, which comes from the primal decomposition. The method is compared to the well-known bisection one and results effectively demonstrate superior performance in terms of convergence speed and computational complexity.

***Keywords*— DBA, efficient optimization algorithms, cross-layer, QoS, fairness.**

I. Introduction

As established by the OSI protocol stack, multiple access of users in any system has to be considered as a link layer funtionality. More precisely, we define such procedures inside the Multiple Access Control or MAC sublayer. Traditional approaches force an a priori subdivision of system resources and on that basis, users are allocated into the system when available resources are left. Classical approaches such as Time Division Multiple Access (TDMA), Frequency Division Multiple Access (FDMA) or Code Division Multiple Access (CDMA) are thus grouped under the concept of static bandwidth allocation. Another possibility is to dynamically assign resources as they are needed and we talk about Dynamic Bandwidth Allocation (DBA). Associated to DBA is the process of requesting system resources and thus the idea of Bandwidth on Demand (BoD) systems.

[0]This work was supported in part by the IST-507052 SatNEx Network of Excellence, MEC projects ESP2005-03403 and ESP2006-26372-E, and by ESTEC Contract 19237/05/NL/AD.

The motivation is to provide better and more efficient usage of the scarce radio spectrum with good Radio Resource Management (RRM) schemes [1]. Concerning DBA, the problem is mathematically more interesting when the sum of the demands exceeds the available capacity, which forces to share the capacity. In some cases, however, the opposite case is also significative. A conceptually different but mathematically similar problem is that of distributing remaining resources or capacity to users in order to increase their satisfaction. This situation is realistic, for example, in the case of Digital Video Broadcasting - Return Channel Satellite (DVB-RCS) [2]. In this paper, we consider DVB-RCS as an application example. However, the approach is general and it is still valid for other systems. The goal is to allocate users fairly considering cross-layer information in order to sustain QoS defined at upper layers, such as TCP/IP with DiffServ in our application.

Among the works about DBA, with emphasis on satellite applications, consider [3], [4], [5] and [6]. In [3], a primal decomposition approach that uses approximated solutions for the subproblems is proposed to solve a DBA optimization problem. The goal is to provide a time-efficient algorithm at the same time that fairness among users is guaranteed. A similar and extended work appears in [5]. Fairness issues are analyzed from the perspective of game theory [7] and a dual decomposition approach is proposed to cope with a network DBA problem [8]. The authors in [4] contribute with traffic modelling in geostationary satellite networks operating in Ka band. As a consequence of the work, discrete optimization problems arise at two different time bases: static and dynamic. Finally, the contribution in [6] is devoted to providing QoS in networks with voice and data traffic using TCP-IP with DiffServ. The resulting scheme is also cross-layer.

The novelty of the paper is the proposal of a new method developed under the framework of convex optimization [9] and primal-dual decomposition techniques [8], [10], [11] to fulfill the previous requirements. Different to other approaches, where primal or dual decompositions are 'serially' concatenated, our method intertwines both decompositions. A detailed analysis of the technique will be presented and a stopping criterion that accelerates the convergence of the algorithm will be derived. In this way, a computationally efficient algorithm is derived. Efficiency is of great importance because such algorithms operate in real-time. The faster the solution is found, the higher the number of users potentially the system

1-4244-1251-X/07/$25.00 2007 IEEE.

can manage.

The rest of the paper is organized as follows. Section II models the resource allocation problem as a convex optimization problem and discusses fairness and QoS issues. Section III presents the proposed algorithm, and Section IV contains convergence analysis. Finally, Section V gives some results and Section VI concludes the paper.

II. Problem Formulation and Known Solutions

Consider the following generic resource allocation problem, where a certain quantity of resources P is to be allocated among N terminals or users (x_i is the amount of resources assigned to terminal i),

$$\begin{array}{ll} \max\limits_{\{x_i\}} & \sum_i^N p_i \cdot \log(x_i) \\ s.t. & \sum_i x_i \le P \\ & d_i \le x_i \le D_i \end{array} \quad (1)$$

where $\{d_i\}$ and $\{D_i\}$ define the minimum guaranteed allocation and the requests, respectively. The weights $\{p_i\}$ are used to prioritize users as a function of their QoS requirements.

Note that we can interpret (1) as the sum of weighted logarithmic utility functions. Utility models the user satisfaction as a function of the resources it gets. For logarithmic utility functions, new allocated resources highly increase satisfaction when the user has few resources, whereas it does not provide much benefit in the opposite case. The optimal solution of (1) forces to 'fairly' divide resources. A formal definition of fairness, termed *proportionally fairness*, and related to logarithmic utility functions can be found [12].

The solution to the proposed fair DBA optimization problem can be found semi-analytically. After imposing the Karush-Kuhn-Tucker (KKT) conditions [9], the solution is

$$x_i = \left.\frac{p_i}{\mu}\right]_{d_i}^{D_i} \triangleq \begin{cases} \frac{p_i}{\mu}, & d_i < \frac{p_i}{\mu} < D_i \\ d_i, & \frac{p_i}{\mu} \le d_i \\ D_i, & \frac{p_i}{\mu} \ge D_i \end{cases}, \quad (2)$$

where μ is such that $\sum_i x_i = P$. A classical way to find μ is using the bisection method as in [13]. Another possibility, the hypothesis testing method, can be found in [14]. The reader can find in Figure 1 a graphical interpretation of the solution. A set of communicated boxes (one box per terminal) of unit width and depth equal to p_i is filled with a quantity P of water. The resulting water level can be interpreted as $\frac{1}{\mu}$ and the amount of water in each box corresponds to the amount of resources the terminal gets.

III. Proposed Algorithm

It is a well-known issue in convex optimization theory [9] that problems can be solved both from their primal or dual representation. Moreover, it is also well-studied that, under certain conditions, a large problem can be divided into smaller subproblems thanks to decomposition techniques [11]. Traditionally, decompositions have been established from the dual or primal perspective. Some works discuss sequential mixtures of them, e.g. in [8]. At our best knowledge, these

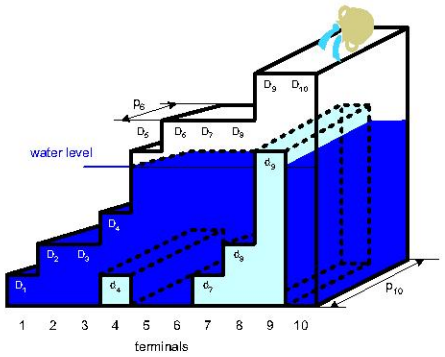

Fig. 1. Graphical interpretation of the solution.

are always serially concatenated, even in the so-called cross-decomposition [15], where primal and dual versions of the same problem are iteratively interleaved.

We now propose a novel method, where primal and dual versions of the same problem are coupled in a different way as it is done in cross-decomposition. We propose to interleave primal and dual decompositions (as defined in [8]) in the scheme.

Detailed convergence analysis will be then addressed and as a result, a criterion that prematurely stops the algorithm iterations without sacrificing the exact solution is obtained. This criterion highly improves computational efficiency.

Consider now the problem

$$\begin{array}{ll} \min\limits_{\{x_i,y_i\}} & -\sum_{i=1}^N p_i \cdot \log(x_i) \\ s.t. & \sum_i y_i \le P \\ & x_i \le y_i \\ & d_i \le x_i \le D_i \end{array} \quad (3)$$

which is equivalent to (1). Note that $\sum_i y_i \le P$ is the only coupling constraint.

Clearly, given the values of $\{y_i\}$, the problem can be divided into N independent and simple problems, named the subproblems, with solution

$$x_i = y_i\big]_{d_i}^{D_i}, \quad \lambda_i = \begin{cases} \frac{p_i}{x_i}, & y_i \le D_i \\ 0, & y_i > D_i \end{cases} \qquad i = 1 \ldots N \quad (4)$$

where the values $\{\lambda_i\}$ are the Lagrange multipliers associated to the constraints $x_i \le y_i$ ($i = 1 \ldots N$).

In a classical primal decomposition approach, the values of y_i are successively updated by the master problem in order to achieve global optimality while verifying $\sum_i y_i \le P$. Traditionally, gradient-type approaches are used. Among their disadvantages, one can mention that a user-defined adaptation step is required and convergence is generally 'slow'.

Dual decomposition is derived from the dual function of (3) when the *only* coupling constraints are taken into account. The master dual problem maximizes this dual function, which depends on the dual variable μ,

$$\begin{array}{ll} \max\limits_{\mu} & g(\mu) = \sum_{i=1}^N g_i(\mu) - \mu P \\ & \mu \ge 0 \end{array} \quad (5)$$

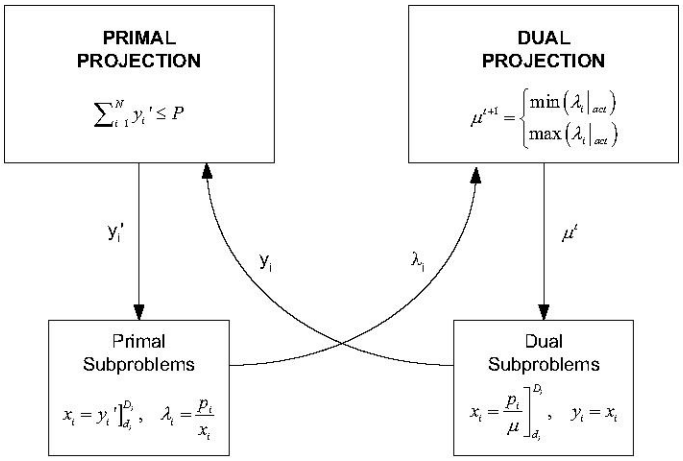

Fig. 2. Picture of the proposed algorithm.

where

$$g_i(\mu) = \min_{\substack{\{x_i, y_i\} \\ x_i \le y_i \\ d_i \le x_i \le D_i}} -p_i \cdot \log x_i + \mu \cdot y_i \qquad (6)$$

Note that with dual formulation, the problem can be decoupled into the functions $\{g_i(\mu)\}$. Incorporating *now* the individual (non-coupling) user constraints (recall that the dual function considers *only* the coupling constraint) to $g_i(\mu)$, we obtain the dual subproblems in (6). Fixing now a value for μ, the primal variables are readily found by

$$x_i = \frac{p_i}{\mu}\Big]_{d_i}^{D_i}, \qquad y_i = x_i, \qquad i = 1 \ldots N \qquad (7)$$

As in the primal counterpart, traditional dual decomposition approaches reach solutions based on gradient-type updates for the master dual problem. Therefore, the same disadvantages exist.

Our proposal combines both strategies in a unified algorithm. Starting from an initial guess of μ, say μ^t, we compute primal variables $\{x_i, y_i\}$ using dual subproblems (7). Now, and instead of updating μ (as in a traditional approach), we correct the potentially unfeasible y_i values to fulfill the coupling constraint $\sum_i y_i = P$. We refer to this step as Primal Projection, as it is obtained with the Euclidean projection [9]. We get in this way the values $\{y_i'\}$ and we assume that the coupling constraint is active (otherwise the problem is decoupled and the solution is readily found). Next, y_i' values are used by the primal subproblems in (4) to obtain the dual variables $\{\lambda_i\}$. The final step, which we call Dual Projection, updates μ assuring 'feasibility' on the dual variables (we comment on this in the sequel). Dual Projection is computed as either the minimum or the maximum of a subset of the previous $\{\lambda_i\}$ values (and remains unchanged for all iterations),

$$\mu^{t+1} = \begin{cases} \min\left(\{\lambda_i|_{act}\}\right) \\ \max\left(\{\lambda_i|_{act}\}\right) \end{cases} \qquad (8)$$

where $\{\lambda_i|_{act}\}$ defines the subset of the $\{\lambda_i\}$ values that are active. A λ_i is active if, for the associated primal y_i' value, $y_i' \in (d_i, D_i)$ holds.

Let us now briefly comment on 'dual feasibility'. Consider the Lagrangian function [9] of (3),

$$L(\{x_i, y_i, \lambda_i\}, \mu) = -\sum_{i=1}^{N} p_i \cdot \log(x_i) + \mu \cdot (\sum_{i=1}^{N} y_i - P) + \qquad (9)$$
$$+ \sum_{i=1}^{N} \lambda_i \cdot (x_i - y_i) + \sum_{i=1}^{N} \gamma_i \cdot (x_i - D_i) - \sum_{i=1}^{N} \delta_i \cdot (x_i - d_i)$$

and take the partial derivative with respect to y_i,

$$\frac{\partial L(\{x_i, y_i, \lambda_i\}, \mu)}{\partial y_i} = \mu - \lambda_i, \qquad i = 1, \ldots, N \qquad (10)$$

As KKT optimality conditions impose zero value to these derivatives, the optimal solution must verify $\mu = \lambda_1, \ldots, \lambda_N$. We say that a solution $\{\lambda_i\}$ is 'dual feasible' if and only if $\lambda_1 = \lambda_2, \ldots, \lambda_N$ and therefore, the Dual Projection must take a value within the candidates $\{\lambda_i\}$. For active users, λ_i is univocally determined by the dual subproblems in (3), whereas for non-active users, more values are valid since one of the two Lagrange multipliers associated to the constraints $d_i \le x_i \le D_i$ has non-zero value. Therefore, it makes sense to discard non-active users.

To end this section, Figure 2 contains a picture of the proposed algorithm, referred as Coupled Primal-Dual Decompositions algorithm. The difference with [15] is that optimization with respect to $\{x_i\}$ and $\{y_i\}$ is split in the algorithm, allowing us to update them without taking into account past decisions.

IV. Convergence Analysis

Consider the following expression

$$\left| \frac{1}{\mu^{t+1}} - \frac{1}{\mu^*} \right| \qquad (11)$$

which serves us to study the evolution of the absolute value of the difference between the optimal water-level, μ^*, and the successive algorithm updates. The objective is twofold:

i) proof that the algorithm effectively converges and ii) learn about the speed of convergence of the method.

Let us assume $\mu^t < \mu^*$ and $\mu^{t+1} = \min(\{\lambda_i|_{act}\})$. Next, we can write the optimal values of variables $\{y_i\}$ as

$$y_i^* = \begin{cases} \frac{p_i}{\mu^*}, & i \in \bar{\mathcal{S}}^* \\ d_i, & i \in \mathcal{M}^* \\ D_i, & i \in \mathcal{D}^* \end{cases}, \quad (12)$$

where $\bar{\mathcal{S}}^*$ is the subset of terminals with optimal solution $y_i \in (d_i, D_i)$, $\mathcal{M}^*$ defines the users with solution $y_i = d_i$ and $\mathcal{D}^*$ includes terminals with solution $y_i = D_i$. After imposing $\sum x_i = P$, we get the optimal water-level value

$$\frac{1}{\mu^*} = \frac{P - \sum_{i\in\mathcal{D}^*} D_i - \sum_{i\in\mathcal{M}^*} d_i}{\sum_{i\in\bar{\mathcal{S}}^*} p_i} \quad (13)$$

In the sequel, the derivation of $\frac{1}{\mu^{t+1}}$ from $\frac{1}{\mu^t}$ is reviewed. First, the dual subproblems use μ^t to propose their candidates for the primal variables

$$y_i = \begin{cases} \frac{p_i}{\mu^t}, & i \in \bar{\mathcal{S}}^t \\ d_i, & i \in \mathcal{M}^t \\ D_i, & i \in \mathcal{D}^t \end{cases} \quad (14)$$

where $\bar{\mathcal{S}}^t$, $\mathcal{M}^t$ and $\mathcal{D}^t$ are the counterpart of $\bar{\mathcal{S}}^*$, $\mathcal{M}^*$ and $\mathcal{D}^*$ at iteration t, respectively. The y_i values are corrected by the Primal Projection, resulting into

$$y_i' = y_i - \frac{\sum_i y_i - P}{N} = y_i - k \quad (15)$$

with $k > 0$ (the water-level in t is over the optimum). Primal subproblems propose now their candidates for the dual variables λ_i,

$$\frac{1}{\lambda_i} = \frac{y_i'\big]_{d_i}^{D_i}}{p_i} = \begin{cases} \frac{1}{\mu^t} - \frac{k}{p_i}, & i \in \bar{\mathcal{S}}^{t'} \\ \frac{d_i}{p_i}, & i \in \mathcal{M}^{t'} \\ \frac{D_i}{p_i}, & i \in \mathcal{D}^{t'} \end{cases} \quad (16)$$

Note that $\bar{\mathcal{S}}^{t'}$ must not necessarily coincide with $\bar{\mathcal{S}}^t$. The same is true for $\mathcal{M}^{t'}$ and $\mathcal{D}^{t'}$. Finally, the updated μ results from the Dual Projection

$$\frac{1}{\mu^{t+1}} = \frac{1}{\min\{\lambda_i|_{act}\}} = \frac{1}{\mu_t} - \frac{k}{p_{max}} \quad (17)$$

with $p_{max} = \max_{i\in\bar{\mathcal{S}}^{t'}}\{p_i\}$. This result reveals us that $\frac{1}{\mu^{t+1}} < \frac{1}{\mu^t}$.

Moreover, it also holds that $\frac{1}{\mu^{t+1}} > \frac{1}{\mu^*}$. This is proved as follows. The $\{y_i'\}$ variables exactly fit the total resource constraint, so that

$$\sum y_i' = P = \sum_{i\in\mathcal{D}^{t'}} D_i + \sum_{i\in\mathcal{M}^{t'}} d_i + \sum_{i\in\bar{\mathcal{S}}^{t'}} \left(\frac{p_i}{\mu^t} - k\right) \quad (18)$$

As $\sum_{i=1}^N y_i' = \sum_{i=1}^N y_i^*$, otherwise the y_i' are the optimal ones, there will be some values where $y_i' \geq y_i^*$ and some others where $y_i' \leq y_i^*$. Accordingly, the same reasoning (in the inverse form) is valid for the associated λ_i values, which are obtained from (16). Finally, choosing $\frac{1}{\mu^{t+1}}$ to be the maximum value among $\{\frac{1}{\lambda_i|_{act}}\}$ assures $\frac{1}{\mu^{t+1}} > \frac{1}{\mu^*}$.

Grouping results, we can state that

$$\frac{1}{\mu^*} < \frac{1}{\mu^{t+1}} < \frac{1}{\mu^t} \quad , \quad (19)$$

which proves convergence.

Reconsider now (11) and include the expression for μ^{t+1},

$$\left|\frac{1}{\mu^{t+1}} - \frac{1}{\mu^*}\right| = \left|\frac{1}{\mu^t} - \frac{1}{\mu^*} - \frac{k}{p_{max}}\right| \quad (20)$$

Using (14) and (15) with $\sum y_i' = P$, k is

$$k = \frac{\sum_{i\in\bar{\mathcal{S}}^t} \frac{p_i}{\mu^t} + \sum_{i\in\mathcal{M}^t} d_i + \sum_{i\in\mathcal{D}^t} D_i - P}{N} \quad (21)$$

and as $k > 0$, the solutions computed by the dual subproblems in (14) exceed the optimal ones if they do not saturate. If $\mu^t < \mu^*$, the following statements hold at the t^{th} iteration

$$\begin{aligned} \mathcal{D}^t &= \mathcal{D}^* \cup \mathcal{D}^{extra} \\ \bar{\mathcal{S}}^* &= \bar{\mathcal{S}}^t \cup \bar{\mathcal{S}}^{extra} \\ \mathcal{M}^* &= \mathcal{M}^t \cup \mathcal{M}^{extra} \end{aligned} \quad (22)$$

Introducing (22) in (21) and identifying (13) in the resulting expression, we obtain

$$\begin{aligned} k &= \frac{\sum_{i\in\bar{\mathcal{S}}^*} p_i}{N}\left[\frac{1}{\mu^t} - \frac{1}{\mu^*}\right] + \frac{1}{N}\Bigg[\sum_{i\in\mathcal{D}^{extra}} D_i \\ &\quad - \sum_{i\in\bar{\mathcal{S}}^{extra}} \frac{p_i}{\mu^t} - \sum_{i\in\mathcal{M}^{extra}} d_i\Bigg] \end{aligned} \quad (23)$$

As the algorithm converges, it exists an iteration $t^{\triangleright}$ where $\bar{\mathcal{S}}^{extra} = \mathcal{M}^{extra} = \mathcal{D}^{extra} = \{\emptyset\}$, so that

$$k = \frac{\sum_{i\in\bar{\mathcal{S}}^*} p_i}{N}\left[\frac{1}{\mu^t} - \frac{1}{\mu^*}\right]. \quad (24)$$

Combination of (24) and (20) shows the speed of convergence of the algorithm when the optimal zone ($t \geq t^{\triangleright}$) is reached,

$$\begin{aligned} \left|\frac{1}{\mu^{t+1}} - \frac{1}{\mu^*}\right| &= \left|\frac{1}{\mu^t} - \frac{1}{\mu^*}\right| - \frac{\sum_{i\in\bar{\mathcal{S}}^*} p_i}{p_{max}\cdot N}\left|\frac{1}{\mu^t} - \frac{1}{\mu^*}\right| \\ &= \left|\frac{1}{\mu^t} - \frac{1}{\mu^*}\right| \cdot \left(1 - \frac{\sum_{i\in\bar{\mathcal{S}}^*} p_i}{p_{max}\cdot N}\right) \end{aligned} \quad (25)$$

In the case that $\mu^t > \mu^*$, a similar reasoning conducts to the same convergence results by choosing $\mu^{t+1} = \max(\{\lambda_i|_{act}\})$. The situation with $\mu^t > \mu^*$ and $\mu^{t+1} = \min(\{\lambda_i|_{act}\})$ is also guaranteed to converge as it is easy to verify that $\mu^{t+1} < \mu^*$. Similar reasoning is valid for the opposite situation, i.e. $\mu^t < \mu^*$ and $\mu^{t+1} = \max(\{\lambda_i|_{act}\})$.

Let us assume for instance the particular case with $p_i = 1$ for all terminals. In that situation, (25) can be rewritten as

$$\left|\frac{1}{\mu^{t+1}} - \frac{1}{\mu^*}\right| = \frac{n_s}{N}\left|\frac{1}{\mu^t} - \frac{1}{\mu^*}\right| \quad (26)$$

where n_s is the number of terminals that have a saturated solution, i.e. $x_i = D_i$ or $x_i = d_i$. Note that when no terminals saturate, the optimum is found in one iteration. This is true because the Primal Projection exactly computes that optimum. On the contrary, when nearly all users saturate, the convergence of the algorithm is much slower.

In order to improve this feature, consider the following quantity, which is obtained from three consecutive μ updates,

$$B^t = \frac{\frac{1}{\mu^{t+1}} - \frac{1}{\mu^t}}{\frac{1}{\mu^t} - \frac{1}{\mu^{t-1}}} \tag{27}$$

We propose to calculate B^t at each iteration until $B^t = B^{t+1}$. This happens when B^t and B^{t+1} take the following B_c value

$$B_c = 1 - \frac{\sum\limits_{i \in \bar{\mathcal{S}}^*} p_i}{p_{max} \cdot N}. \tag{28}$$

When the condition holds we are in the optimal zone ($t \geq t^{\triangleright}$) and we find the exact solution in one step as

$$\mu^* = \frac{\sum\limits_{i \in \bar{\mathcal{S}}^*} p_i}{P'} \Rightarrow y_i = \frac{p_i}{\mu^*}, \tag{29}$$

where $P' = P - \sum_{i \in \mathcal{D}^*} D_i - \sum_{i \in \mathcal{M}^*} d_i$.

In the next section, numerical results are presented together with an application example.

V. Numerical Results

Imagine an scenario with P resources to be fairly distributed among N terminals, e.g. DVB-RCS. The simulated values of minimum guaranteed resources and users' demands is performed as follows. Guaranteed resources are randomly calculated as $d_i \sim \mathcal{U}[0, 10]$, where $\mathcal{U}[a, b]$ is the representation of a uniformly distributed random variable with values in the interval $[a, b]$. Demands are obtained as $D_i \sim d_i + \mathcal{U}[0, 100]$. P depends on these values as $P = \alpha \sum d_i + (1 - \alpha) \sum D_i$, where $\alpha \in [0, 1]$.

The first simulation result appears in Figure 3 and shows the time required to compute the solution using three different methods: the bisection method (with precision set to $0.5 \cdot 10^{-12}$ with respect to the quantity $\sum_i x_i - P$), the hypothesis testing method [14] and the proposed method. All methods run in a Pentium®-Mobile processor running at $1.73GHz$. N is evaluated from 1000 to 20000 terminals (in steps of 500 terminals), α is set to 0.25 and 40 Monte-Carlo runs are averaged. We notice that the proposed method is in general more efficient than the other two and that the hypothesis testing method is a good election when the number of users is low. Both the hypothesis testing and the proposed methods have the advantage of having much less time variance, with a more predictable computational time.

The second group of simulations includes Figure 4 and Figure 5. Figure 4 shows the speed of convergence of the previous three algorithms when $N = 10000$ terminals and α equals 0.25 and 0.75, respectively. Our algorithm has almost linear convergence in both cases and reaches the exact solution

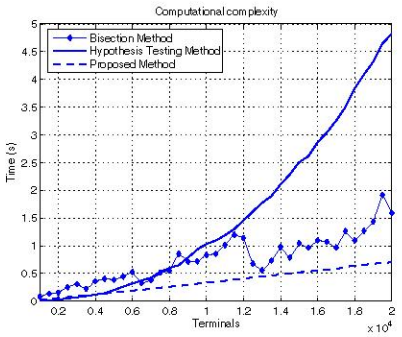

Fig. 3. Computational time of the algorithms ($P = 0.25 \sum d_i + 0.75 \sum D_i$).

at a certain iteration. This explains the abrupt convergence at the end. Both the proposed method and the hypothesis testing one require less iterations than the bisection method and depending on the scenario (driven by α), one of them obtains first the solution. With respect to the proposed method, note the different convergence slopes. In the first case ($\alpha = 0.25$), more users reach their requests or saturate (as P is higher) and convergence is slower, as is verified from (25) and the interpretation in (26).

Next simulation studies the behavior of the algorithm when Dual Projection uses the min or the max function. In Figure 5, the reader can find the evolution of the successive updates of the water-level, i.e. $\frac{1}{\mu^t}$, using both functions and a certain initial value for the water-level. Note that when the minimum is used and the initial water-level is over the optimal one, the successive iterates remain always over the optimum value. On the contrary, if we use the maximum in the Dual Projection with $\frac{1}{\mu^0} > \frac{1}{\mu^*}$, we verify that the first update leads to $\frac{1}{\mu^1} < \frac{1}{\mu^*}$ and that the successive updates remain under the optimum water-level, as seen in the previous section.

The last simulation in Figure 6 examines a possible cross-layer application example, DBA in DVB-RCS. For the sake of brevity, we present here a simplified vision of the system. The interested reader can find a complete description of the operational framework we have considered in [16]. Assume that at MAC layer, 20 users request transmitting 100 ATM

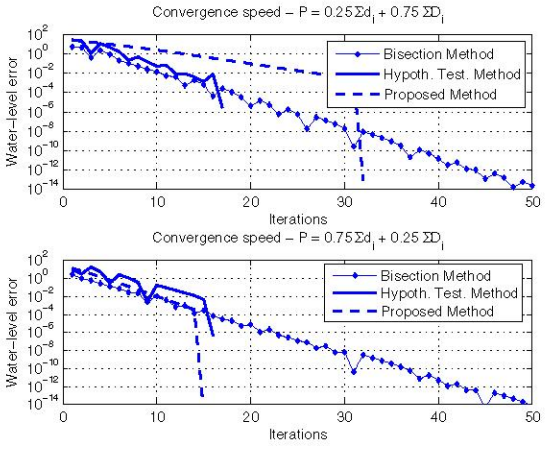

Fig. 4. Convergence speed of the algorithms.

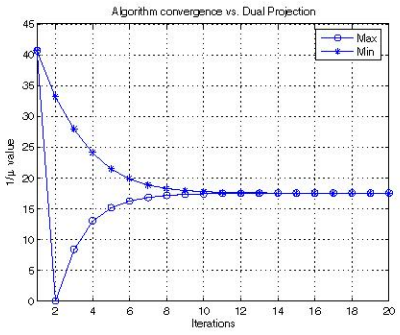

Fig. 5. Choice of Dual Projection.

cells. We assume three different types of IP traffic, namely QoS 1, QoS 2 and QoS 3. In order to effectively take into account their different characteristics, we facilitate cross-layer information from the TCP to the MAC layers and this information is used in configuring the $\{p_i\}$ values. Assume 5 users of QoS 1 ($p_i = 2$), 5 more users of QoS 2 ($p_i = 1.5$) and 10 users of QoS 3 ($p_i = 1$). At each allocation cycle, the system has 1000 ATM cells to be assigned and each user requests the number of ATM cells that are in queue. In Figure 6 (top) we plot the aggregated number of ATM cells transmitted by all users in each QoS group at each allocation cycle, whereas in Figure 6 (down) we plot the number of ATM cells transmitted by a single user of each QoS. Note that by this mechanism, QoS can be effectively sustained at IP level. Observe that users with higher priority finish first their transmission as they get more resources, whereas lowest priority users will have higher latencies and will access the whole system capacity only when high priority users have nothing to send . Finally, realize that a proper design of $\{p_i\}$ values determines any desired balance among the potential variety of services.

VI. Conclusions

This paper has contributed with a novel and time-efficient algorithm for solving the problem of DBA in systems that operate in a BoD basis. The solution maximizes fairness as defined according to the *proportionally fairness* sense and allows us to include cross-layer information in the parameters. It is derived under the framework of convex optimization and uses the ideas of primal/dual decomposition and cross-decomposition to derive an algorithm that requires neither a user-defined adaptation step, as in primal/dual decomposition, nor the solutions of past iterations, as in cross-decomposition.

We have shown through analysis and simulations the performance of our algorithm when compared to others, such as the bisection method and the hypothesis testing method. Note that the bisection method is widely used in the literature (e.g. in water-filling) and is considered to be rather efficient. Time efficiency is of great importance as it limits the size of the allocation problem (number of users, different connections per user, etc.) that can be solved in the available time in systems operating in real-time. Finally, we have analyzed a possible application example, extracted from the DVB-RCS scenario. It has been shown that cross-layer can be effectively introduced in the algorithm and QoS policies defined at upper layers sustained.

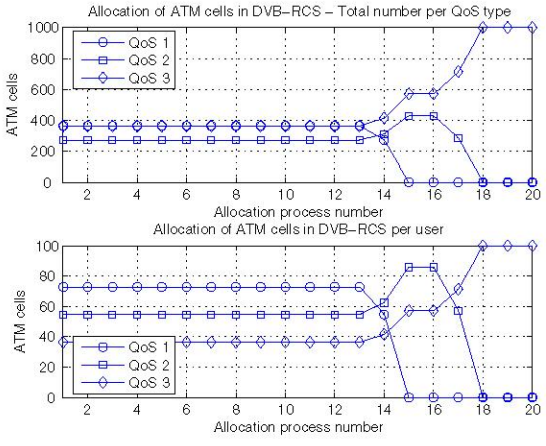

Fig. 6. Allocation example.

References

[1] H. Chen, L. Huang and S. Kumar, *Radio Resource Management for Multimedia QoS Support in Wireless Networks*, Kluwer Academic Publishers, 2004.

[2] ETSI, "Digital Video Broadcasting (DVB); Interaction Channel for Satellite Distribution Systems," *ETSI EN 301 790*, Apr 2005.

[3] A. Girard, C. Rosenberg and M. Khemiri, "Fairness and Aggregation: A Primal Decomposition Study," *Networking 2000, Lecture Notes in Computer Science 1815, Springer-Verlag, pp. 667-678*, May 2000.

[4] N. Celandroni, F. Davoli and E. Ferro, "Static and Dynamic Resource Allocation in a Multiservice Satellite Network with Fading," *Int. J. Satell. Commun. Network., Vol. 21, No. 4-5, pp. 469-487*, July-Oct 2003.

[5] H. Yaïche, R.R. Mazumdar and C. Rosenberg, "A game theoretic framework for bandwidth allocation and pricing in broadband networks," *IEEE/ACM Trans. on Networking, Vol. 8, No. 5, pp. 667-678*, Oct 2000.

[6] H. Jiang and W. Zhuang, "Cross-Layer Resource Allocation for Integrated Voice/Data Traffic in Wireless Cellular Networks," *IEEE Trans. on Wireless Comm., Vol. 5, No. 2, pp. 457-468*, Feb 2006.

[7] A. Muthoo, *Bargaining Theory with Applications*, Cambridge University Press, 1999.

[8] D.P. Palomar and M. Chiang, "Alternative Decompositions for Distributed Maximization of Network Utility: Framework and Applications," *in Proc. IEEE Infocom, Barcelona, Spain*, Apr 2006.

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

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

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

[12] F. Kelly, "Charging and Rate Control for Elastic Traffic," *Eur. Trans. on Telecomm., Vol. 8, No. 1, pp. 33-37*, Jan 1997.

[13] G.V. Reklaitis, A. Ravindran and K.M Ragsdell, *Engineering Optimization: Methods and Applications*, Wiley-Interscience, 1983.

[14] G. Seco-Granados, M.A. Vázquez-Castro, A. Morell and F. Vieira, "Algorithm for Fair Bandwidth Allocation with QoS Constraints in DVB-S2/RCS," *in proc. IEEE Global Telecomm. Conf. (GLOBECOM), San Francisco (USA)*, Nov 2006.

[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.

[16] A. Morell, G. Seco-Granados and M.A. Vázquez-Castro, "Joint Time Slot Optimization and Fair Bandwidth Allocation for DVB-RCS Systems," *in proc. IEEE Global Telecomm. Conf. (GLOBECOM), San Francisco (USA)*, Nov 2006.