Algorithm for Fair Bandwidth Allocation with QoS Constraints in DVB-S2/RCS

Gonzalo Seco-Granados, María Angeles Vázquez Castro, Antoni Morell, Fausto Vieira
Telecommunications and Systems Engineering Department
Universitat Autònoma de Barcelona (UAB)
Email: {gonzalo.seco, angeles.vazquez,antoni.morell}@uab.es, fvieira@sunaut.uab.es

***Abstract*— This paper presents a general framework and the corresponding solution for the problem of fair resource allocation among entities with absolute and relative QoS requirements. It is described how the framework can be applied to a variety of scenarios, in particular to the scheduling of the TDM transmission in DVB-S2 and to the dynamic bandwidth allocation needed in DVB-RCS systems. A usual need in this type of problems is that the allocation has to be computed in (almost) real-time even when the number of entities is very large. The paper proposes a low-complexity algorithm. The algorithm provides the exact solution and numerical simulations shows its low computation time.**

I. INTRODUCTION

The new version of the Digital Video Broadcasting over Satellite standard, named DVB-S2, has been recently completed. One of the main contributions of this new version is the introduction of Adaptive Coding and Modulation (ACM). Notwithstanding, the standard gives solely some indications about the architecture of MAC (Medium Access Control) layer, and its design is an open area of research. A return link is needed to implement ACM and interactive applications. The DVB-RCS (Return Channel via Satellite) system can be used for this purpose. Although the DVB-RCS standard details some particular procedures at MAC level, the interaction of these procedures with higher protocol layers and their appropriate use to provide QoS (Quality of Service) deserve more comprehensive studies.

We consider a satellite network with star architecture. The communication is established between a hub or gateway and several satellite terminals (ST). Both the hub and the terminals are located on the earth, and the communication is via a GEO (Geostationary Earth Orbit) satellite. We assume that the satellite is transparent (i.e. non-regenerative) only for the sake of simplicity, but the algorithm presented in the paper is also valid for the regenerative case. The coverage area is served by different beams and, apart from differences on the link budget, all beams are formally identical. Therefore, we can consider in the following only one of the beams without any loss in generality. The communication from the gateway to the terminals is normally called forward link. On the other hand, the communication from the terminals to the hub is referred to as return link. Although these two channels are very different from an information theoretic point of view, they have one key aspect in common, which is that a number of resources have to be shared by a number of entities. In this situation, an adequate design of the MAC layer of the protocol stack becomes of paramount importance.

In Section II, we will describe the forward and return links and see how the general concept of having resources shared by many entities can be particularized for these two cases. In Section III, the formulation and solution of a fair allocation problem with QoS is presented. A low-complexity exact algorithm is proposed in Section IV. Finally, Sections V and VI contain the simulation results and conclusions.

II. SYSTEM MODEL

A. Forward Link: DVB-S2

The DVB-S2 standard proposes the use of a number of TDM (Time Division Multiplexing) carriers per beam. The total available bandwidth is allocated to the different beams following a certain frequency reuse pattern and, hence the overall scheme used by the set of beams is actually an hybrid TDM/FDM (Time/Frequency Division Multiplexing). Each TDM stream is divided in time slots (also called packets since each time slot contains a DVB-S2 packet). Note that a DVB-S2 packet is rather long compared to other systems' encapsulations and it may contain several IP packets. As we are considering interactive (unicast) applications, the information in each IP packet is intended for only one terminal. Each DVB-S2 packet is transmitted using the ACM mode given by the SNIR (Signal-to-Noise-plus-Interference Ratio) at the end terminal. We assume transmission in the Ka band which implies that the SNIR depends basically on the presence of rain or clouds above the terminal. Each ACM mode is characterized by a spectral efficiency, η_m for $m = 1 \ldots M$, measured in information bits per channel symbol, where M is the number of ACM modes. Unlike other systems, the division of the TDM stream is not uniform because in general the duration of each slot, T_m, as well as the number of information bits per slot, b_m, are different for each ACM mode.

The standard does not specify how the TDM stream is formed; it is the task of the MAC scheduler to put the slots into sequence. This is done in a centralized (i.e. non-distributed) way at the hub. The design of the allocation algorithms offers a large number of possibilities depending on how the traffic is classified, which is the scheduling criterion, etc. For instance, the traffic to be transmitted by the hub can be classified according several criteria, such as, the ACM mode used by the destination terminal at that moment, the

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

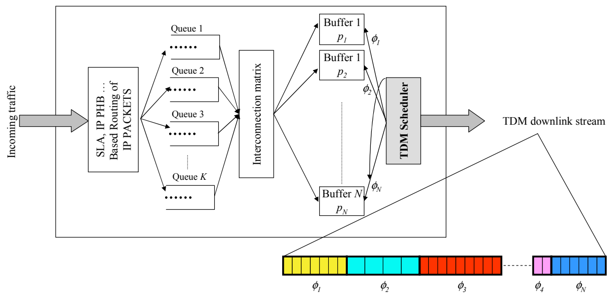

Fig. 1. Scheduling in the forward link.

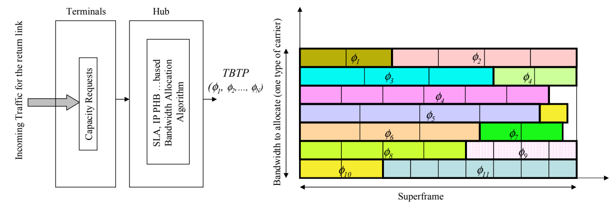

Fig. 2. Dynamic bandwidth allocation in the return link.

PHB (Per Hop Behavior) requested at IP level, the SLA (Service Level Agreement) with the corresponding user, etc. The scheduling criterion may be, for instance, to maximize the overall throughput of the system, to guarantee a maximum delay for some users, to provide to same throughput regardless of the ACM mode, etc.

As the architecture of the MAC scheduler is not specified in detail in the standard, some architectures and scheduling objectives have already been proposed in the literature [1], [2], and despite their differences, some common aspects can be found. In all cases, the incoming traffic is classified in several sets (e.g. queues and buffers) according to one or several of the criteria mentioned above (see Fig. 1). Next, an algorithm and a criterion for sharing the common resources, which is the TDM stream, has to be defined. The output of this algorithm should be the fraction of time that is allocated to each traffic set (or queue). Note that the fraction of time can be easily translated into the percentage of total slots by making used of the slot durations T_m [1]. A very general assumption is that the inputs of the scheduling algorithm are the minimum fraction of time to be allocated to each traffic set, the maximum fraction of time required by each traffic set (since it is useless to allocate resources that cannot be used) and an indication of the relative priority of each set (in order to be able to provide relative Quality of Service, QoS). Finally, some scheduling criteria have been proposed in [1], [3].

B. Return Link: DVB-RCS

Although the return link is quite different to the forward link in what concerns the physical and link layers, we will see that the resource allocation problem can be cast in similar terms. The access scheme implements a Bandwidth on Demand (BoD) solution, which is also referred to as Demand Assigned Multiple Access (DAMA). Therefore, the terminals make capacity requests to the NCC (Network Control Center) via the hub. The requests are in the form of a transmission rate and/or volume. Inherently to the BoD solution, there is a Dynamic Bandwidth Allocation (DBA) scheme, which is executed at the NCC and makes the capacity assignment to the terminals. Every superframe the assignment is sent via the forward link using the Terminal Burst Time Plan (TBTP).

The air interface of the return link is based on Multi Frequency Time Division Multiple Access (MF-TDMA). The bandwidth is divided in several carriers and each carrier is divided in time slots. Therefore, a grid is defined over a time-frequency area with total dimensions equal to the superframe duration times the total bandwidth (Fig. 2). The terminals can transmit inside each element of the grid (or slot), possibly using adaptive coding, according to the assignment made by the NCC. The time-frequency grid need not be regular; indeed DVB-RCS leaves the definition of the grid very open. An optimal design of the time slot duration in a generic scenario is proposed in [4]. Once the grid is defined, the slots in frequency-time plane (or in a region of the plane if we restrict the problem to certain carriers) are the resources to be shared. As in the forward link, the allocation is done in a centralized manner and an optimality criterion needs to be defined. The output of the DBA algorithm is of course the TBTP, and the inputs are the requests from the terminals together with possibly additional minimum capacity and relative QoS requirements.

III. Formulation of the Resource Allocation Problem with QoS

A. General problem and optimal solution

In this section, we present the mathematical formulation of a general allocation problem that encompasses at least the two presented above. We start with the notation definition. The total amount of resources is denoted as C. In the forward link scheduling problem, $C = 1$ since it represents the fraction (dimensionless) of time during which the TDM stream can be used. In the return link DBA problem, C is equal to the total number of slots in the time-frequency grid (see [4] for the computation of this number). The resource assignments are represented by the variables $\phi_1, \ldots, \phi_N$, which are fractions of time or amounts of slots in each of the two situations considered above, respectively. N represents the number of *entities* that share the resources, where *entities* is interpreted in a wide sense. In the forward link, an entity is each of the buffers that have to be scheduled, and each buffer may correspond to any combination of ACM mode, terminal, traffic type, flow, etc. depending on the architecture. In the return link, an entity may range from a set of terminals to a flow, in the highest and lowest degree of aggregation, respectively. Each entity ϕ_i demands $D_i > 0$ resources and has a guarantee minimum[1] of $d_i \geq 0$. The allocation is not trivial when $\sum_{i=1}^{N} D_i > C$ and, in this situation, a criterion for deciding

[1]We assume that the admission control ensures that $\sum_{i=1}^{N} d_i \leq C$

which demands are fulfilled and to which extent needs to be adopted. We choose *fairness* as criterion and hence we want to make a *fair* allocation, where *fair* is understood in the sense of the Nash bargaining solution (which is not the same as Nash equilibrium). Therefore, as a first idea, we should maximize a function of the following form:

$$\prod_{i=1}^{N} \phi_i \ , \tag{1}$$

since it is the way to satisfy the fairness axioms of efficiency, symmetry and independence of irrelevant alternative [5].

Function (1) or very similar ones have been used in the literature for bandwidth allocation or flow control problems [5]–[7]. In order to formulate the problem in the most general manner, we want to assign different priorities to the different entities, and this is one way how QoS requirements are introduced in our allocation. Note that this a way to impose relative levels of QoS among the entities, while the absolute levels of QoS are imposed by the d_i values. Moreover, the use of relative QoS at MAC level is in line with the DiffServ model at the IP layer, and this facilitates the translation of the IP PHB classes into priorities at MAC level. We will see in Section III-B that a minimum resource allocation can also be guaranteed using the priorities p_i.

We denote the priority of each entity as $p_i > 0$. Function (1) is modified in order to account for the priorities. The function, namely $\prod_{i=1}^{N} \phi_i^{p_i}$, corresponds to the so-called asymmetric Nash Bargaining solution, and it can be interpreted as if each entity is split in p_i sub-entities and function (1) is applied to the sub-entities, all of them having the same priority.

At this point, we can gather all the previous elements and define the solution of the generic allocation problem as the vector $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_N]$ that solves the following constrained optimization:

$$\max_{\boldsymbol{\phi}} \sum_{i=1}^{N} p_i \ln(\phi_i) \tag{2}$$

$$\phi_i \geq d_i \ \ \phi_i \leq D_i \ i = 1, \ldots, N \tag{3}$$

$$\sum_{i=1}^{N} \phi_i \leq C \ . \tag{4}$$

The objective function has been expressed in a logarithmic form in order to obtain a convex problem (with the appropriate change of signs). Each term $p_i \ln(\phi_i)$ can be interpreted as the utility of the allocation to the *i*-th entity, as in [8]. The Lagrangian is

$$L = \sum_{i=1}^{N} p_i \ln(\phi_i) + \lambda \left(C - \sum_{i=1}^{N} \phi_i \right) + \sum_{n=1}^{N} M_i (D_i - \phi_i) + \sum_{n=1}^{N} m_i (\phi_i - d_i) \ , \tag{5}$$

where λ, M_i and m_i are non-negative Lagrange multipliers. At the optimum, the partial derivative of L with respect to all ϕ_i's must be zero

$$\frac{\partial L}{\partial \phi_i} = \frac{p_i}{\phi_i} - \lambda - M_i + m_i = 0 \tag{6}$$

The value of ϕ_i that solves (6)

$$\phi_i = \frac{p_i}{\lambda + M_i - m_i} \tag{7}$$

is optimal when the following sufficient and necessary Karush-Kuhn-Tucker (KKT) conditions are satisfied

$$M_i (D_i - \phi_i) = 0 \tag{8}$$

$$m_i (d_i - \phi_i) = 0 \ . \tag{9}$$

Using (8)-(9) in (7) the solution of the allocation problem (2)-(4) can be expressed as

$$\phi_i = d_i + [p_i \nu - d_i]_0^{D_i - d_i} \ , \tag{10}$$

where $[x]_0^y = \min(\max(x, 0), y)$ and ν is chosen to satisfy the total capacity constraint with equality[2]. This solution recalls the so-called waterfilling solutions that usually appear when information theoretic criteria are applied to the distribution of power among different sub-channels ([9] contains a very good review of problems leading to waterfilling solutions). However, those solutions are usually simpler because there is only one threshold per variable in the final solution, that is to say, using the nomenclature introduced above, the solutions are of the type $[\cdot]_0^\infty$.

In the DVB-RCS scenario, the values ϕ_i must be integer since they represent a number of slots, but (10) results in general in real numbers. However, this is not a drawback because a simple conversion algorithm can be used, similar to those in [6] or [4]. It consists in first rounding down all non-integer ϕ_i's. Second, the total amount of freed resources are used to increment in one the allocation to some of the entities that have been just rounded down. Since not all of them can be increased, the increment is applied in the order given by the p_i's. It can be shown that this procedure is optimal if all the decremented entities have the same priority, otherwise the loss in optimality is usually negligible since the amount of allocated resources in much larger than one [4].

B. Particular cases and related solutions

The resource allocation represented by equation (10) generalizes other solutions that have appeared in the literature. In [7], the fair allocation of capacity to a large population of best-effort connections is addressed. A reduced version of problem (2)-(3) is proposed since neither absolute nor relative QoS constraints are considered. The framework presented above can be easily applied to such a problem by setting $p_i = 1$ and $d_i = 0$, $\forall i$. [6] deals with DBA in geostationary satellite networks with on-board processing. Priorities are considered by not explicit minimum allocations, which amounts to fixing $d_i = 0$. On the other hand, that work employs multiple

[2]As said before, we assume that $\sum_{i=1}^{N} D_i > C$ otherwise the solution is the trivial one: $\phi_i = D_i$. If $\sum_{i=1}^{N} D_i < C$, the proposed method could still be used to distribute the free resources.

total capacity constraints on different subsets of entities since both the uplink and downlink allocation problems are addressed jointly. In [5], each term in the objective function is $p_i \ln(\phi_i - d_i)$. Although this may seem a small modification, it leads to complete different results since, in that case, the utility to a certain entity depends on the additional resources allocated over the requested minimum. Despite being fair in mathematical terms, we think that in the DVB-S2 and DVB-RCS scenarios described above this solution favors excessively entities with high requested minima.

An intuitive graphical representation of the optimal solution (10) can be only obtained when all the priorities p_i are equal [4]. It resembles the conventional waterfilling solution but in a container with an irregular lid. Although less intuitive, a graphical representation in case p_i are not equal to one another but $d_i = 0$ is also possible, and it reveals that the solution (10) satisfies $\phi_i \geq \min\left(D_i, p_i/\sum_{n=1}^{N} p_n\right)$ in such case.

IV. Low-Complexity Algorithm

In a broadband satellite communication system, the number of entities N may be very large, on the order of several thousands or more, if the resource allocation is performed at the level of terminals, users or even flows (i.e. connections). The difficulty is that the allocation is dynamic and, hence, it must be done in almost real-time. For instance, in DVB-RCS there are typically thousands of time slots per superframe, and the TBTP must be updated every superframe, whose duration is typically 265ms [10]. Therefore, low-complexity algorithms for computing the optimal solution (10) are highly desirable. In [7], a hierarchical but suboptimal approach is proposed, in which the complete problem is divided into several smaller problems. In [6], several heuristic methods are presented. A general algorithm for solving waterfilling algorithms has been recently proposed in [9], but it is not directly applicable to our solution due to the differences pointed out at the end of Section III-A.

Each ϕ_i in the optimal solution (10) is a function of a single parameter ν and, depending on the value of ν, it can take three possible sets of values: $\phi_i = d_i$, $d_i < \phi_i < D_i$ or $\phi_i = D_i$. Therefore, 3^N hypotheses could be checked for finding the one to which the optimum is belongs. This number of hypotheses is prohibitively large, however an attentive look at the problem unveils that only $2N-1$ are possible, and they can be explored using a binary tree with logarithmic complexity.

The first step of the algorithm is to compute the hypotheses thresholds

$$\nu_i^l \triangleq \frac{d_i}{p_i} \qquad \nu_i^u \triangleq \frac{D_i}{p_i}. \tag{11}$$

Next, the first iteration ($k = 1$) of the algorithm is described in detail. Those thresholds are merged in a single vector, named $\boldsymbol{\nu}^{lu}(1)$, and arranged in increasing order. The range between each pair of consecutive elements of the vector corresponds to a valid hypothesis for ν. The value of ν is set equal to the element in the center of that vector[3], i.e. $\nu = \left[\boldsymbol{\nu}^{lu}\right]_{\lceil 2N/2 \rceil} = \left[\boldsymbol{\nu}^{lu}\right]_N$. At this point, all the entities are classified in three groups: S_1 contains the indices i of those entities for which $\nu \geq \nu_i^u$, S_3 corresponds to the entities that satisfy $\nu \leq \nu_i^l$, and S_2 contains the rest. The following quantities are computed as:

$$C_1 = \sum_{i \in S_1} D_i \qquad C_2 = \nu \sum_{i \in S_2} p_i \qquad C_3 = \sum_{i \in S_3} di \tag{12}$$

$$C_a = C_1 + C_2 + C_3\ . \tag{13}$$

If the fortuitous situation $C_a = C$ occurs, the present value of ν is the optimum one and the algorithm stops, otherwise the algorithm continues. If $C_a > C$, a new vector of thresholds $\boldsymbol{\nu}^{lu}(2)$ is formed by taking the first half of the existing vector of thresholds. If $C_a < C$, the new vector of thresholds $\boldsymbol{\nu}^{lu}(2)$ is the second half of the current vector $\boldsymbol{\nu}^{lu}(1)$.

The usual stopping criterion is as follows. If vector $\boldsymbol{\nu}^{lu}(2)$ contains only two elements, the correct hypothesis (i.e. the correct distribution of the entities in the sets $S1$, $S2$ and $S3$) has been found, and the optimal value of ν is computed as

$$\nu_{opt} = \frac{C - C1 - C3}{\sum_{i \in S_2} p_i}\ . \tag{14}$$

If $\boldsymbol{\nu}^{lu}(2)$ contains more than two elements, the algorithm continues with the selection a new value of ν equal to the element located in middle of that vector. In the first iteration, this is:

$$\nu = \left[\boldsymbol{\nu}^{lu}(2)\right]_{\lceil N/2 \rceil}\ . \tag{15}$$

The value of k is incremented in one and the procedure is repeated again starting from the computation of the new sets S_1, S_2, S_3 and the corresponding values C_1, C_2 and C_3.

It is worth noting that those sets and values need not be recomputed from scratch, which further reduces the complexity. If $C_a > C$ at the previous iteration, only the entities in $S_1 \cup S_2$ may be possibly reallocated using the new value of ν; whereas the entities in $S_2 \cup S_3$ have to be revised, if $C_a < C$ at the previous iteration. Table I summarizes the steps of the algorithm together with the associated complexity order.

The number of iterations is on the order of $\log(N)$, with N operations per iteration. This results in an overall complexity order of $N \log(N)$, which is of the same order as the complexity of the best sorting algorithm.

V. Simulation Results

In this section, we investigate the complexity of the algorithm by means of numerical simulations in order to corroborate the results in the previous section. Complexity is measured by the execution time of the algorithm coded in Matlab® and running on a conventional desktop computer, in particular a Pentium® Dual at 3GHz.

All parameters are selected randomly as follows. Each d_i is a uniform random variable between 0 and 100. D_i is computed as d_i plus a uniformly-distributed random increment between 0 and 100. The priorities d_i are set randomly between 1 and 3 using a uniform distribution. The absolute values of all these parameters are not really relevant, only their relative values

[3] $[\cdot]_n$ denotes the n-th element of a vector and $\lceil \cdot \rceil$ denotes the nearest bigger integer.

Step	Complexity order
Compute ν_i^l and ν_i^u, $\forall i$	N
Order vector $\boldsymbol{\nu}^{lu}(1)$	$N\log(N)$
For each iteration $k = 1, \ldots$	$\log(N)$ iterations
Select the value of ν	1
Form the sets S_1, S_2 and S_3	N
Compute the values C_1, C_2 and C_3	N
Check whether $C_a = C$ (and finish if condition is fulfilled)	1
Check whether $C_a > / < C$ and take the first/second half of $\boldsymbol{\nu}^{lu}(k)$ to build $\boldsymbol{\nu}^{lu}(k+1)$	1
If $\boldsymbol{\nu}^{lu}(k+1)$ has only two elements, compute the optimum ν and finish. Otherwise, select take as new value of ν the value in the middle of the vector and make a new iteration	1
Compute the optimum ϕ_i's	N

TABLE I

SUMMARY OF THE ALGORITHM FOR FAIR ALLOCATION WITH QOS.

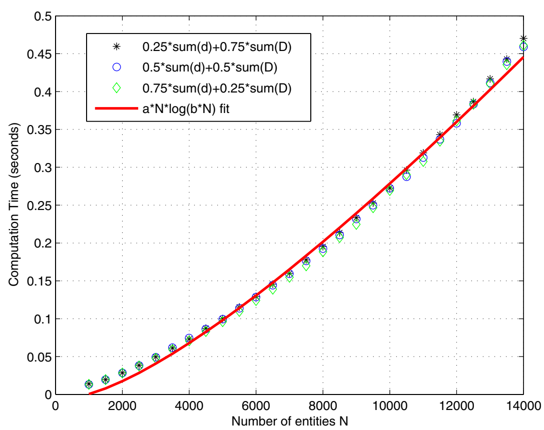

Fig. 3. Complexity of the proposed algorithm.

are important; and actually the fact that all of them have been chosen randomly corresponds to a worst-case situation since there is complete lack of structure in the problem.

Fig. 3 shows computation time required by the proposed algorithm as a function of the number of entities N. It is worth remarking that after this time an optimum fair allocation with QoS constraints is obtained. The comparison with the bisection method is not fair because the bisection does not provide exact results, and its complexity largely depends on the desired accuracy [11]. Three values for the total amount of resources are used: $C = \gamma \sum_{i=1}^{N} d_i + (1-\gamma)\sum_{i=1}^{N} D_i$, with $\gamma =$0.25, 0.5 and 0.75. Each point of the figure is computed by averaging 40 Monte Carlo realizations. The time spent by the sorting algorithm has not been included since it is probably required by any other resource allocation method, as in [9]. Moreover, its contribution is almost negligible.

Fig. 3 confirms that the computation time grows as $N\log(N)$, and it is rather insensitive to the value of C. The following expression fits accurately the simulation results:

$$T\,[\text{seconds}] = 2.732 \cdot 10^{-5} N \log_{10}(0.001045N) \ . \qquad (16)$$

In the case of DVB-RCS, if the allocation must be computed for instance every 20ms or 50ms, the algorithm can manage up to 2100 or 3250 entities, respectively. Such a large value of entities makes it possible to perform the dynamic bandwidth allocation at the level of user (needles to say, at the level of terminal as well) or even flow. With this technique the problem stated in [7, Sec. 2] is feasible.

VI. CONCLUSIONS

The paper has presented a general framework for resource allocation that can be applied to many scenarios of practical interest, such as forward link scheduling in DVB-S2 and dynamic bandwidth allocation DVB-RCS. The framework takes into account relative and absolute quality of service constraints in the form of priorities and required minimum service levels, respectively. The criterion used to make the resource assignment is the concept of *fair* treatment of the entities competing for the resources. The optimal solution of the problem has been presented, and a low-complexity algorithm for obtaining the exact solution has been derived. Simulation results have confirmed the relevance of this algorithm since it allows us to compute exactly the fair resource distribution to a large number of entities in an affordable time, without the need to resort to suboptimal methods.

REFERENCES

[1] M. Váquez-Castro and G. Seco-Granados, "Cross-layer Packet Scheduler Design of a Multibeam Broadband Satellite System with Adaptive Coding and Modulation," Accepted for publication at IEEE Trans. Wireless Communications, 2006.

[2] C. Párraga-Niebla and C. Kissling, "Design and Performance Evaluation of Efficient Scheduling Techniques for Second Generation DVB-S Systems," in *Joint Ka Band and AIAA Conference*, 2005.

[3] F. Vieira and M. A. Váquez-Castro, "Tunable Fairness-Based Capacity Allocation for Satellite Systems with Adaptive Physical Layer," *Joint Ka Band and AIAA Conference*, 2005.

[4] A. Morell, G. Seco-Granados, and M. Váquez-Castro, "Joint Time Slot Optimization and Fair Bandwidth Allocation for DVB-RCS Systems," in *IEEE Globecom*, Nov. 2006.

[5] H. Yaïche, R. Mazumdar, and C. Rosenberg, "A Game Theoretic Framework for Bandwidth Allocation and Pricing in Broadband Networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 667–678, 2000.

[6] G. Açar and C. Rosenberg, "Weighted Fair Bandwidth-on-Demand (WFBoD) for Geostationary Satellite Networks with On-Board Processing," *Computer Networks*, vol. 39, no. 1, pp. 5–20, 2002.

[7] A. Girard, C. Rosenberg, and M. Khemiri, "Fairness and aggregation: A primal decomposition study." in *NETWORKING, IFIP-TC6 European Commission International Conference*, ser. Lecture Notes in Computer Science, vol. 1815. Springer, 2000, pp. 667–678.

[8] F. Kelly, "Charging and Rate Control for Elastic Traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.

[9] D. P. Palomar and J. R. Fonollosa, "Practical Algorithms for a Family of Waterfilling Solutions," *IEEE Trans. Signal Processing*, vol. 53, no. 3, pp. 686–695, Feb 2005.

[10] "Digital Video Broadcasting (DVB); Interaction channel for Satellite Distribution Systems; Guidelines for the use of EN 301 790," ETSI TR 101 790 v1.2.1, 2005.

[11] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C, the Art of Scientific Computing*. Cambridge University Press, 1995.

1-4244-0357-X/06/$20.00 ©2006 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.