Dynamic bandwidth allocation in GEO satellite networks: a predictive control approach

L. Chisci a R. Fantacci b T. Pecorella c

a Dipartimento di Sistemi e Informatica, Università di Firenze, Via S. Marta 3, 50139 Firenze, Italy
b Dipartimento di Elettronica e Telecomunicazioni, Università di Firenze, Via S. Marta 3, 50139 Firenze, Italy
c CNIT, Unità di Firenze, Via S. Marta 3, 50139 Firenze, Italy

Abstract

The paper presents a novel approach for Dynamic Bandwidth Allocation (DBA) in GEO Satellite Networks based on adaptive predictive control. In the proposed DBA scheme each Satellite Gateway (SG) uses an adaptive predictor to forecast the future input traffic flow along with a bandwidth controller to generate a bandwidth request to the Bandwidth Manager (BM) according to a receding-horizon policy, via solution of an optimal control problem. A multi-service traffic scenario is considered with a careful eye to guarantee differentiated Quality of Service requirements. Simulation experiments demonstrate that the predictive DBA technique proposed in this work outperforms fixed allocation as well as existing reactive DBA methods.

Key words: Telecommunication networks, dynamic bandwidth allocation, predictive control.

1 Introduction

1.1 Current state of art

Over the last few years there has been a growing interest of the control engineering community towards communication networks, see e.g., the special
issues (Anantharam and Walrand eds., 1999; Bushnell ed., 2001; Delli Priscoli and Thomesse eds., 2003; Gong and Ba¸sar eds., 2002) and the references therein. Many networking issues such as congestion control (Imer et al., 2001; Low et al., 2002; Mascolo, 1999) routing (Altman et al., 2002; Baglietto et al., 1999), scheduling (Delli Priscoli and Isidori, 2002), admission control (Jagannathan, 2002), power control (Alpcan et al., 2002), dynamic bandwidth allocation (Delli Priscoli and Pietrabissa, 2002, 2004) have been successfully tackled with the powerful tools of dynamical system and control theories, for several types of networks, e.g. ATM or IP, wired or wireless, terrestrial or satellite. The present paper addresses Dynamic Bandwidth Allocation (DBA) of IP satellite networks by means of a Model Based Predictive Control (MBPC) approach.

IP satellite networks are gaining a considerable interest, mainly due to their ability to deliver high bandwidth services to nation-wide areas. The main trend in telecommunication market is to offer a wide variety of services to an increasing number of users. Internet2, i.e. the global Internet upgrade, will be able to satisfy this trend, but its deployment will probably occur gradually. Satellites are the proper candidates to provide high quality IP services in the development phase of fast terrestrial networks. It is also well known that satellites are proved to be cost-effective whenever the coverage area is large or the population is sparse.

However, some difficulties are connected with a full IP-based transport mechanism implementation on geostationary (GEO) satellite networks. TCP-based protocols are affected by the large delay-bandwidth product of GEO satellite networks. Moreover Quality of Service (QoS) provisioning has to take into account the peculiarities of satellite networks, where the resources (mainly bandwidth) are scarce and, hence, cannot be left unused. As far as the TCP adaptation to the satellite link is concerned, a number of solutions have been proposed. The solution of this problem is, however, out of the scope of this work. In the following, it will be assumed that a proper technique is used in order to solve the TCP limitations related to the bandwidth-delay problem.

As a matter of fact, a rough solution for terrestrial system QoS provisioning is to overestimate the requested capacity. Unfortunately, this method is not suited for satellite communication systems due to bandwidth limitations and excessive user costs, since usually the user has to pay for the allocated bandwidth, instead of the one effectively used. For this reason an efficient resource management technique has to be integrated with the QoS support to obtain the two main goals of resource fair sharing and user QoS satisfaction (Afek et al., 1996; Fulp and Reeves, 1997).

Although DBA techniques have been extensively applied to terrestrial networks (e.g., ATM), DBA is still an open issue in satellite communication systems. Recently some access techniques for LEO satellite systems have been
proposed in the literature, e.g., S-PRMA (Del Re et al., 1999), CFDAMA (Le-Ngoc and Krishnamurthy, 1995), etc., however, due to their use of contention-based signaling channels, their usefulness is limited to LEO systems. Our contribution will be focused on QoS related issues, assuming that the Medium Access Control (MAC) layer is able to provide a reliable and efficient signaling method.

An important drawback in applying DBA schemes to GEO satellite systems is the high response delay, e.g., 500 ms if the Network Control Center (NCC) is located on earth, or 250 ms if the NCC is integrated in the satellite on-board device.

Furthermore, as far as delay-sensitive or bursty traffic management is concerned, an on-line adaptive resource allocation scheme has to be developed in order to meet the required QoS in terms of needed bandwidth, while avoiding unnecessary delays. In particular, the following approaches can be used:

- fixed allocation proportional to the maximum source rate;
- fixed allocation at a given rate using DBA for peak bursts;
- full DBA techniques.

The fixed bandwidth approach raises several problems and limitations, as the maximum source rate is usually unknown. Further, this approach leads to a huge bandwidth waste, hence, it is not suitable for satellite systems. A full DBA technique implies that a user terminal does not hold any fixed channel during inactivity periods. This can save bandwidth, however the common signaling channel results to be overloaded during acquisition phases, hence implying an higher delay and congestion problems. As a consequence, a mixed approach seems to be the most flexible choice. In this way, a small fixed channel per user terminal is held and a pool of DBA channels are used during peak traffic periods. The proposed technique is based on the statistical properties of IP traffic to enhance the efficiency of bandwidth management. The goal is to obtain a better QoS for the time-sensitive traffic by a predictive capacity request.

A control-based DBA scheme for GEO satellite networks has been presented in (Delli Priscoli and Pietrabissa, 2002). In this work, the bandwidth request is generated by an LTI (Linear Time Invariant) controller consisting of a proportional feedback from the measured queue length, with incorporated Smith predictor for delay compensation. In (Delli Priscoli and Pietrabissa, 2004) the scheme has been improved by introducing a feedforward term from the measured input flow rate and a queue length setpoint which is tuned on the basis of the current network load.
1.2 Contribution

In this paper, we propose a novel DBA scheme based on adaptive predictive control. The main novelties of this DBA scheme with respect to the one of (Delli Priscoli and Pietrabissa, 2002, 2004) are summarized hereafter.

1. A model-based traffic predictor is used (in place of the Smith predictor) in order to forecast the future input flows. More specifically, the predictor relies on an autoregressive model whose parameters are estimated and continuously updated on-line so as to track possible time changes of the traffic characteristics.

2. A nonlinear model-based predictive controller is used to generate the bandwidth request taking into account the current queue length as well as the predicted future input flows. More specifically, the bandwidth controller selects the current request according to a receding-horizon strategy, i.e., via the solution of a constrained optimal control problem with a cost functional trading off performance (viz. low queueing delay) versus bandwidth consumption.

3. A multi-service traffic scenario is considered, i.e., it is assumed that the data packets arriving at the user terminal belong to multiple classes with different QoS requirements and, thus, different priority levels.

The rest of the paper is organized as follows. Section 2 provides an overview of a GEO IP satellite network and of the related DBA problem. Section 3 describes the proposed adaptive predictive DBA scheme. In section 4, this scheme is evaluated and compared with fixed bandwidth allocation and existing reactive DBA techniques via simulation experiments. Finally section 5 concludes the paper.

2 GEO satellite network and Dynamic Bandwidth Allocation

We consider a system where Internet connections to end-users are provided by a GEO satellite, two-way system, as depicted in fig. 1. In this scenario, a number of users (a residential zone, a corporate location, a small town, etc.) are connected to the Internet via a Satellite Gateway (SG), capable of bidirectional transmissions to/from a GEO satellite.

It is assumed that the downlink (i.e., the link from the Satellite to the SGs) is a broadcast channel, hence it does not need any resource management technique. On the contrary, the uplink (i.e., the link from SGs to the Satellite) is shared among all the SGs, so a suitable technique to manage the finite bandwidth capacity is needed to improve the system performance.
It is further assumed that the Medium Access Control (MAC) layer is able to dynamically request and release bandwidth with a proper signaling technique, e.g., embedding some signaling data on the main data stream or by resorting to dedicated signaling channels.

In order to manage the requests from different SGs, a Bandwidth Manager (BM), located in the Network Control Center (NCC), is used. When a SG needs a bandwidth increase or wants to release some bandwidth, it must send a request/release signal to the BM. The BM, depending on the system state (i.e., the available bandwidth) will send to the requesting SG a grant (or a deny) to use more bandwidth. Of course the bandwidth release signal does not require any authorization by the BM. The NCC, of which the BM is a component, may be either located on the earth or on the satellite, depending on the system design as well as on the computational capabilities of the satellite. As a matter of fact, the system characteristics are similar to the ones of a Digital Video Broadcasting-Return Channel System, i.e., DVB-RCS (ETSI-00). DVB-RCS standard defines a number of different allocation methods (i.e., CRA, RBDC, VBDC, FCA\(^1\)) that make it difficult to analyze the control algorithm performance. Hence, we will focus our attention here on the simple

---

\(^1\) CRA (Continuous Rate Assignment) is a fixed bandwidth, negotiated during setup phase. RBDC (Rate Based Dynamic Capacity) is a rate-based capacity request, e.g., \(x\) kbit/s, which may not be fulfilled. VBDC (Volume Based Dynamic Capacity) is a volume request, i.e., \(x\) kbit, which will always be fulfilled as it does not have any time constraint. FCA (Free Capacity Assignment) is an unsolicited bandwidth assignment.
case of SGs making only rate-based bandwidth requests (i.e., RBDC requests). This assumption has been made in order to evaluate the effectiveness of the proposed control algorithm for the management of interactive and multimedia Internet traffic. Real-time traffic may use an IntServ-based QoS management scheme Braden et al. (1994), however, such a scheme exhibits scalability issues. Hence, it is foreseeable that only DiffServ-based approaches (Blake et al., 1998; Durresi et al., 2001) will be allowed on satellite networks. The lack of flow information of the DiffServ approach makes it necessary to resort to RBDC-type requests only, since it is not easy to extract information about the amount of minimum and maximum bandwidth required for such streams.

3 Predictive Adaptive DBA

Our goal is to dynamically allocate the output bandwidth of the SGs. Towards this end, it is assumed that the SG is fed by the traffic input flows $w_i, i \in \mathbb{N} = \{1, 2, \ldots, n\}$, belonging to $n$ prioritized classes of traffic ($w_1$ has the highest priority, $w_n$ the lowest) and that for each class there is a separate FIFO (First-In-First-Out) buffer (queue). The overall DBA scheme adopted in this paper is depicted in the block-diagram of fig. 2. The dynamic bandwidth allocator consists of an Adaptive Traffic Predictor (ATP) and of a Bandwidth Controller (BC). At each sampling period $t$ the BC, using predictions of the future input traffic provided by the ATP and knowledge of the current queue lengths, issues a bandwidth request $u(t)$ to the BM which, in turn, communicates back the assigned bandwidth $v(t)$ after a delay of $\delta$ sampling periods (RTD = Round Trip Delay). The assigned output bandwidth $v(t)$ is then divided by
the scheduler among the output flows $v_i(t), i \in \mathbb{N}$, of the various queues. The functional blocks of the DBA scheme of fig. 2 as well as the adopted model for the input traffic, are described in detail hereafter.

**Queue dynamics**

The system to be controlled consists of $n$ queues, one for each type of traffic. Let $y_i(t), w_i(t), v_i(t)$ denote the length, the input flow and, respectively, the output flow of the $i^{th}$ queue at sampling time $t$. Then the queue dynamics is expressed by the following equations:

$$y_i(t+1) = y_i(t) + w_i(t) - v_i(t) - d_i(t)$$

i $\in \mathbb{N}$ (1)

where $d_i(t)$ represents the number of packets dropped between time instants $t$ and $t+1$. Hereafter, an infinite buffer capacity will be assumed and, consequently, $d_i(t) = 0$ will be considered in (1). The effects of finite buffer capacity and of AQM (Active Queue Management) strategies will be investigated in future work.

**Scheduler**

The output bandwidth of the SG is shared out among the $n$ prioritized services by an appropriate device, named scheduler, which operates independently with respect to the DBA process. Specifically, given the total bandwidth $v(t)$, the output flows from the queues are given by

$$v_i(t) = b_i v(t), b_i \in [0, 1], \sum_{i=1}^{n} b_i = 1.$$ (2)

The coefficients $b_i$ are chosen in a possibly time-varying fashion by the scheduler according to an appropriate scheduling policy.

**Input traffic flow model**

In order to model the input traffic flow, an Auto-Regressive (AR) process with offset

$$w(t) = \sum_{j=1}^{h} a_j w(t-j) + \bar{w}$$ (3)
will be assumed for the aggregated flow

$$w(t) \triangleq \sum_{i=1}^{n} w_i(t).$$

Furthermore, it will be assumed that the aggregated traffic $w(t)$ is shared out among the various services as follows

$$w_i(t) = c_i w(t), \ c_i \in [0, 1], \ \sum_{i=1}^{n} c_i = 1. \quad (4)$$

The coefficients $a_j, c_i, \overline{w}$ are estimated on-line by means of the RLS (Recursive Least Squares) algorithm so as to take into account the strong time-variability of multimedia traffic. Conversely, the order $h$ of the AR model is estimated off-line on the basis of a large record of experimental data so as to give an appropriate tradeoff between predictive capability and model complexity.

**Remark** - Notice that a single AR model for the aggregated flow $w$ is considered, instead of $n$ separate AR models for the flows $w_i$, just for the sake of simplicity.

### Adaptive Traffic Predictor

Using the model (3)-(4) and knowledge of the flows $w_i(k)$ up to the current time $t$, the ATP module performs predictions of the future traffic flows at the input of the SG. To this end, let us define the state vector

$$\mathbf{x}_w(t) \triangleq \begin{bmatrix} w(t) \\ w(t-1) \\ \vdots \\ w(t-h+1) \end{bmatrix} \quad (5)$$

Then the predictor based on the AR model (3) can be represented by the following state-space equations

$$\mathbf{x}_w(t+1) = \mathbf{A}_w \mathbf{x}_w(t) + \mathbf{B}_w \mathbf{w}$$

$$\mathbf{y}_w(t+1) = \mathbf{C}_w \mathbf{x}_w(t) \quad (6)$$
where

\[
\begin{bmatrix}
  a_1 & a_2 & \ldots & a_{h-1} & a_h \\
  1 & 0 & \ldots & 0 & 0 \\
  \vdots & \vdots & & \vdots & \vdots \\
  0 & 0 & \ldots & 1 & 0
\end{bmatrix}, \quad
\begin{bmatrix}
  1 \\
  0 \\
  \vdots \\
  0
\end{bmatrix}, \quad
\begin{bmatrix}
  c_1 & 0 & \ldots & 0 \\
  c_2 & 0 & \ldots & 0 \\
  \vdots & \vdots & & \vdots \\
  c_n & 0 & \ldots & 0
\end{bmatrix}
\]

Let \( w \triangleq [w_1, w_2, \ldots, w_n]' \). Initializing (6) with (5), the prediction \( \hat{w}(t+k|t) \) of the input flow vector \( w \) at time \( t+k \) based on the data up to time \( t \), is given by

\[
\hat{w}(t+k|t) = y_w(t+k) = C_w A^k_w x_w(t) + \left( \sum_{i=0}^{k-1} C_w A^i_w B_w \right) w
\]

(8)

In order to make the traffic predictor adaptive, i.e., capable of following time changes of the traffic characteristics, at sampling time \( t \) the parameters \( w, a_j, c_i \) are replaced in (7) by their LS estimates \( \hat{w}(t), \hat{a}_j(t), \hat{c}_i(t) \) based on data up to time \( t \). Let

\[
s(t) = [1, w(t-1), w(t-2), \ldots, w(t-h)]'
\]

Then the LS estimates are computed by the following RLS algorithm:

\[
R(t) = \lambda R(t-1) + s(t)s'(t) \\
z(t) = \lambda z(t-1) + s(t)w(t) \\
r(t) = \lambda r(t-1) + w^2(t) \\
z_i(t) = \lambda z_i(t-1) + w(t)w_i(t), \quad i \in n
\]

(9)

\[
[\hat{w}(t), \hat{a}_1(t), \hat{a}_2(t), \ldots, \hat{a}_h(t)]' = R^{-1}(t)z(t) \\
\hat{c}_i(t) = \frac{z_i(t)}{r(t)} , \quad i \in n
\]

where \( \lambda \in (0,1] \) is an exponential forgetting factor and the following initialization is used

\[
R(0) = \varepsilon I, \quad z(0) = 0, \quad r(0) = \varepsilon, \quad z_i(0) = 0, \quad 0 < \varepsilon \ll 1.
\]

Network Control Center

The BM receives bandwidth requests forwarded by the SGs and decides whether to satisfy or not such requests. On the basis of an appropriate policy of resource (bandwidth) sharing among all the SGs, which is out of the scope
of this paper, the BM assigns to each SG a bandwidth $\pi(t)$ less than or equal to the requested one $u(t)$. This can be modelled by introducing an additive disturbance $d(t) \geq 0$ such that

$$\pi(t) = u(t) + d(t). \quad (10)$$

Hence $d(t)$ represents the unassigned bandwidth. The delay element $z^{-\delta}$ in cascade to the BM block represents the overall delay (RTD), measured in sampling units, between the transmission of the bandwidth request (from the SG to the BM) and the reception of the reply (from the BM to the SG). Hence

$$v(t) = \pi(t - \delta). \quad (11)$$

For the sake of simplicity and also due to unpredictability of the unassigned bandwidth $d$, the model adopted for DBA will assume that $d(t) = 0$, i.e.

$$\pi(t) = u(t) \implies v(t) = u(t - \delta), \quad (12)$$

which means that the requested bandwidth is always assigned.

Bandwidth Controller

Using the model (1), the traffic predictions (8)) and the known scheduling coefficients $b_i$, the BC must decide at time $t$ the bandwidth $u(t)$ to request to the BM for a possible allocation at time $t + \delta$. To this end, let us assume that the requests be always satisfied and let us define

$$x(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \\ u(t-1) \\ u(t-2) \\ \vdots \\ u(t-\delta) \end{bmatrix}, \quad y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{bmatrix}. \quad (13)$$

Then, in order to predict the future evolution of the queues, the BC can use the following model, obtained by combining (1), (2), (4) and (12)

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) + Ew(t) \\ y(t) = Cx(t) \end{cases} \quad (14)$$
where

\[
A = \begin{bmatrix}
1 & 0 & \ldots & 0 & 0 & \ldots & 0 & -b_1 \\
0 & 1 & \ldots & 0 & 0 & \ldots & 0 & -b_2 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & \ldots & 1 & 0 & \ldots & 0 & -b_n \\
0 & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & \ldots & 0 & 0 & \ldots & 1 & 0
\end{bmatrix},
\]

\[
B = \begin{bmatrix}
0 \\
0 \\
\vdots \\
0 \\
1 \\
0 \\
\vdots \\
0
\end{bmatrix},
\]

\[
E = \begin{bmatrix}
I_n \\
0
\end{bmatrix},
\]

\[
C = [I_n, \mathbf{0}]
\]  

(15)

Remarks

- From a system-theoretic point of view, the pair \((A, B)\) is not stabilizable for \(n > 1\). In fact, \(A\) has an eigenvalue in \(z = 1\) with both algebraic and geometric multiplicities equal to \(n\), so that at least \(n\) inputs are required to make such an eigenvalue reachable. This means that, as could be easily argued, the control input \(u\) alone is not able to stabilize \(n > 1\) queues. Conversely, the combination of the bandwidth controller along with the scheduler provides the necessary degrees of freedom for controlling all the \(n\) queues. However, instead of considering a multi-output bandwidth controller which directly synthesizes the bandwidth requests \(u_i(t)\) for all services, we prefer to keep the two tasks of bandwidth request and bandwidth scheduling separated as in the scheme of fig. 2. This choice is motivated by the existence of well-established and efficient scheduling algorithms which can effectively share the allocated bandwidth among the various services according to their priority.

- It is worth pointing out that for the queue system it is not possible anyway to guarantee bounded input bounded state stability. In fact each queue is nothing but a (discrete-time) integrator with a bounded control input \(v_i\), which decreases the queue length, and a persistent disturbance \(w_i\) of unknown arbitrary magnitude, which increases the queue length. For such a system it is structurally impossible to ensure a finite gain from \(w\) to \(y\). Of course, state unboundedness is not experienced in practice as, whenever the buffer exceeds its capacity, incoming packets are dropped. Hence the objective of our bandwidth control will be to optimize performance in terms of a suitable tradeoff between minimizing packet loss and queueing delay on one hand and minimizing bandwidth waste on the other.

The controller’s objective is to allocate an amount of bandwidth \(u(t)\) that allows a good tradeoff among reduction of queue occupancy and of bandwidth
waste. To this end, the following quadratic performance index is introduced

$$J = \sum_{k=0}^{N-1} \left\{ \|y(t+\delta+k) - y^*\|_Q^2 + r [u(t+k) - w(t+\delta+k)]^2 \right\}$$

(16)

where: $\|y\|^2 = y^TQy$, $Q = Q^T \geq 0$, $r > 0$, $y^*$ is the vector of the desired queue lengths (typically selected as $y^* = 0$) and $N \geq 1$ is the prediction horizon. The first term in (16) penalizes the number of elements in the queues by the weight matrix $Q$, while the second term penalizes the difference between the input and the output bandwidth by the scalar weight $r$; note that in an ideal system (i.e., a system with bounded queues and no bandwidth waste) the difference should be zero. Hence, the QoS objective can be pursued by minimizing $J$ in (16) with respect to the input sequence $\{u(t), u(t+1), \ldots, u(t+N-1)\}$ subject to the following constraints, respectively on the queue lengths and on the output bandwidth,

$$\begin{cases}
y_{\min} \leq y(t+\delta+k) \leq y_{\max} \\
u_{\min} \leq u(t+k) \leq u_{\max}
\end{cases} \quad k = 0, 1, \ldots, N-1$$

(17)

where vector inequalities must be interpreted as component-wise. In order to reduce the dimension of the optimization problem and possibly slow down the rate of change of bandwidth, the following parameters are introduced:

- $M$: number of future inputs to be chosen;
- $\ell$: time interval between two consecutive inputs to be chosen;

and we impose a prediction horizon $N = M\ell$ thus constraining the input $u$ to remain constant over consecutive subintervals of length $\ell$ of the overall prediction horizon $[0, N) = [0, \ell M)$. More precisely, it is imposed that

$$u(t+(j-1)\ell) = u(t+(j-1)\ell+1) = \cdots = u(t+j\ell-1), \quad j = 1, 2, \ldots, M$$

(18)

and, therefore, the $M$ inputs $\{u(t), u(t+\ell), \ldots, u(t+(M-1)\ell)\}$ must be chosen so as to minimize the cost (16) which takes into account the evolution of the queues over the time window $[t+\delta, t+\delta+N)$. The main feature of the performance index (16) is the presence of the weights $Q$ and $r$ which allow to realize an appropriate tradeoff among the conflicting needs of low costs and high performance. In particular, the choice of the matrix $Q = \text{diag}\{q_1, q_2, \ldots, q_n\}$ also enables to penalize the specific queues in a different way and thus reflect the priority ranking of the corresponding classes of service. At the end of the constrained optimization, the BC issues to the BM the request of the bandwidth $u(t)$ while ignores the subsequent samples $\{u(t+\ell), u(t+2\ell), \ldots, u(t+(M-1)\ell)\}$ of the optimal sequence. At the next sampling period, the BC repeats the same procedure with the updated data according to the receding (moving)-horizon control strategy.
Summing up the above developments, at each sampling time $t$ the DBA algorithm consists of the following steps.

**Step 1.** Update the LS estimates $\hat{w}(t), \hat{a}_j(t), \hat{c}_j(t)$ via (9).

**Step 2.** Compute for $k = \delta, \delta + 1, \ldots, \delta + N - 1$ the prediction $\hat{w}(t+k|t)$ via (8) using the LS estimates $\hat{w}(t), \hat{a}_j(t), \hat{c}_j(t)$ for $\bar{w}, a_j, c_j$.

**Step 3.** Update the state $x(t)$ defined in (13) and get the current coefficients $b(t)$ from the scheduler.

**Step 4.** Assuming (12), determine the optimal control sequence $\{u(t+\ell_j)\}_{j=0}^{M-1}$ so as to minimize (16) subject to (17) and (18). In this optimization the scheduling coefficients $b_i(t)$ are replaced by $b_i(t)$, while the unknown future input flow vector $w(t + \delta + k)$ is replaced by its prediction $\hat{w}(t + \delta + k|t)$ via a certainty-equivalence approach.

**Step 5.** Forward to the BM the request of bandwidth $u(t)$, ignoring the subsequent optimal inputs in accordance to the receding-horizon strategy.

**Remarks**

- The adopted receding-horizon strategy seems especially well-suited in this DBA context wherein the planned control variable (output bandwidth) at a given time instant $t$ must be requested by the user terminal to the BM and is possibly confirmed (or denied) with some delay at time $t + RTD$. In fact it is clear that, in this situation, the possible denial of the requested bandwidth acts as a disturbance on the system state and that, to this end, control re-planning of the receding-horizon strategy seems very appropriate.

- From a computational point of view, the main burden of the proposed DBA algorithm is in the constrained optimal control problem of step 4, which consists of a QP (Quadratic Programming) problem of dimension $M$. There are mainly two types of iterative algorithms for QP solution: active-set algorithms which involve, in the worst case, a number of iterations exponentially increasing with $M$ and interior-point algorithms which involve only $O(M)$ iterations but, on the other hand, require a much higher cost per iteration. Active-set algorithms are by far more convenient for small values of $M$ while the converse holds true for large $M$. In the specific DBA context small values of $M$, e.g. $M = 2 \div 5$, have been found adequate and hence the Active Set Method turns out to be the preferable algorithmic option.

- It is worth pointing out that alternative forms of the cost (16) could be chosen. For instance, one could replace in (16) the term $u(t+k) - w(t+\delta+k)$ by $u(t+k)$ or $u(t+k)/w(t+\delta+k)$. Further, one could use linear (instead of quadratic) penalties thus obtaining an LP (instead of QP) optimization problem. A comparison among different choices of the performance index is, however, out of the scope of the present paper where only (16) will be
In order to evaluate the proposed control algorithm performance, a simulator based on the NePSing C++ framework (\textsuperscript{[?]}\textsuperscript{\textsuperscript{1}}) has been developed. The simulator follows the scheme depicted in fig. \textsuperscript{2}. Multiple SGs have been considered, with a simple round-robin bandwidth assignment policy implemented in the BM.

Due to the well-known Internet traffic self-similar properties (Park and Willinger, \textsuperscript{2002}), the input traffic flows $w_i$ used in the simulations have Long Range Dependent properties, with Hurst parameters $H_1 \simeq 0.852$ for the first class, $H_2 \simeq 0.851$ for the second class and $H_3 \simeq 0.85$ for the third one. Following DiffServ terminology, $w_1$ represents an Expedited Forward (EF) class flow, $w_2$ an Assured Forward (AF) class flow and $w_3$ a Best Effort (BE) class flow.

For comparison purposes, three fixed bandwidth management schemes have been considered. The first one aims at minimum delay at the price of a maximum bandwidth waste, the second one aims at the exact contrary while the third should represent a tradeoff between the two. The three schemes will hereafter be called ‘Fix 31’, ‘Fix 40’ an ‘Fix 54’. The numbers represent the amount of slots/frame granted to each SG. The slots per frame values have been chosen according to the mean and peak slots per frame values needed by each SG, measured offline. It should be noted that 31 slots/frame is about the 106.5\% of the average input bandwidth, 40 slots/frame is about the 137.4\% and 54 slots/frame is the peak input bandwidth. It must be remarked that the fixed bandwidth policies are presented here only for comparison purposes, as it is almost impossible for any practical purpose to know in advance the average and peak bandwidth requests of the SGs.

Further, a DBA scheme based on the Smith predictor (Delli Priscoli and Pietrabissa (\textsuperscript{2002, 2004})) has been considered; in the sequel this will be referred to as SPC (Smith Predictor Control) and will be used to perform cumulative bandwidth requests. In a sampled-data implementation, the SPC DBA scheme generates the bandwidth request at sampling time $t$ by

$$u(t) = w(t) + K \left[ y(t) - \sum_{k=t-\delta}^{t-1} u(k) \right] + K \left( 1 - \frac{\delta^*}{\delta} \right) \sum_{k=t-\delta}^{t-1} w(k)$$

(19)

where $\delta^* \geq \delta$ is a load-dependent parameter representing the desirable queueing delay while $K$ is a gain which, for stability, must belong to the interval $(0, 1]$.
Finally, the RHC DBA scheme presented in the previous section will be con-
sidered. All the compared DBA schemes adopt the same scheduling algorithm
in order to differentiate the flows of the prioritized classes. A very simple
scheduling scheme has been used in order to better evaluate the DBA effects
on the QoS of the various flows and to keep the simulation complexity low. The
implemented scheduler follows a simple priority-based rule, where the actual
priority of each queue is dynamically chosen depending on the queue length.
The EF queue has always a fixed priority (equal to 1), the AF and BE queues
follow two piecewise linear functions shown in fig. 3, where \( m = 50 \) kbit and
\( M = 100 \) kbit When two or more queues have the same priority, a random
one is selected. This scheduler is a version of the simple priority scheduler,
modified in order to avoid BE queue starvation.

The simulation time was set long enough to reach statistically stable results.
The parameters of the RHC and SPC algorithms used in the simulations are
reported in table 1.

The performance of the selected allocation schemes is illustrated in figs. 4-10.
The analysis is carried out in terms of bandwidth waste and packet delay.

Fig. 4 shows that RHC outperforms all the other schemes as far as bandwidth
waste is concerned. Average values are reported in Table 2; also note that
the Fix 54 scheme gives a much larger average waste and that while the SPC
scheme exhibits low average losses, the absolute number of wasted slots can
be very high.

On the other hand fig. 5 (please note the logarithmic scale) shows that RHC
is better than Fix 31 and SPC in terms of packet delay. The Fix 31 scheme,
indeed, is slightly better than RHC; however, the Fix 31 maximum delay is
way above the RHC one. Moreover note that \( Q \) parameters allow fine-tuning
of RHC performance.

In order to better highlights the multi-service capabilities of RHC, fig. 6 shows
the packed delay of each traffic class. Note that using the RHC scheme the
EF class delay is almost negligible and and that a differentiated treatment
Table 1
Simulation and Control parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round Trip Delay (RTD)</td>
<td>500 [ms]</td>
</tr>
<tr>
<td>Number of SGs</td>
<td>15</td>
</tr>
<tr>
<td>Number of input flows</td>
<td>3 (EF, AF, BE)</td>
</tr>
<tr>
<td>Average bandwidth ($w_i$)</td>
<td>$\approx 142.87$ [kbit/s]</td>
</tr>
<tr>
<td>Average bandwidth ($w$)</td>
<td>$\approx 29.105$ [slots/frame]</td>
</tr>
<tr>
<td>$q_1$</td>
<td>10.0</td>
</tr>
<tr>
<td>$q_2$</td>
<td>9.0</td>
</tr>
<tr>
<td>$q_3$</td>
<td>8.0 (default)</td>
</tr>
<tr>
<td>$r$</td>
<td>1.0</td>
</tr>
<tr>
<td>$y^*$</td>
<td>0</td>
</tr>
<tr>
<td>Sampling period</td>
<td>125.0 [ms]</td>
</tr>
<tr>
<td>$\delta = RTD$/sampling period</td>
<td>4</td>
</tr>
<tr>
<td>Prediction parameters $N, M, \ell$</td>
<td>4, 2, 2</td>
</tr>
<tr>
<td>Order of AR model $h$</td>
<td>4</td>
</tr>
<tr>
<td>$y_{min}, y_{max}$</td>
<td>0, $\infty$ [bit]</td>
</tr>
<tr>
<td>$u_{min}, u_{max}$</td>
<td>0, 8 [Mbit/s]</td>
</tr>
<tr>
<td>SPC parameters $K, \delta^*$</td>
<td>1, 4</td>
</tr>
</tbody>
</table>

Table 2
Average bandwidth loss in slots/frame using the various allocation schemes.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>RHC</th>
<th>SPC</th>
<th>Fix 31</th>
<th>Fix 40</th>
<th>Fix 54</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss</td>
<td>0.817448</td>
<td>5.0431</td>
<td>1.98435</td>
<td>10.1018</td>
<td>23.0843</td>
</tr>
</tbody>
</table>

of the various priority classes is guaranteed, ensuring that RHC is DiffServ-compliant. In order to stress the capabilities of the RHC scheme to carry delay-sensitive traffic as the EF-class traffic, the Packet delay Jitter\(^2\) for the EF class has been measured and it is shown in fig. 7. The figure shows that the Jitter performance of the proposed scheme is as good as the Fix 31 and SPC schemes and only slightly lower than the Fix 40 scheme.

Fig. 8 illustrates the RHC recovery after a congestion, simulated by large (500 kbit) initial values of queue lengths. In this simulation the queue length

\[ \delta_i = x_i - x_{i-1}, \text{ where } x_i \text{ is the } i\text{-th packet delay, the Jitter is} StDev\{\delta\} \]

\(^2\) The Packet delay Jitter or simply Jitter is measured as the average standard deviation of the delay variation between two consecutive packets. As is defined $\delta_i = x_i - x_{i-1}$, where $x_i$ is the $i$-th packet delay, the Jitter is $StDev\{\delta\}$.
has been used to better show the congestion recovery speed. Moreover RHC parameters have been tuned to \( q_1 = 10, q_2 = 5, q_3 = 3 \) in order to highlight the phenomena. The different EF, AF and BE queues have been feed with 500 kbit at 5, 15 and 25 seconds, simulation time. It can be observed that while the congestion recovery speed for the EF and AF queues is almost the same, the bandwidth wasted after the congestion is lower for the AF class, and as regards to the BE class, it is negligible. The BE class, as a matter of fact, have the longest congestion recovery time, as expected considering the low \( q_3 \) value. We can conclude that the proposed scheme exhibits a fast recovery even for the lowest priority class and that the recovery times and bandwidth waste of the EF, AF, BE queues decrease with increasing priority, as expected.

Fig. 9 illustrates the RHC performance varying the number of SGs. The simulation starts with only 5 SGs, at about 500s the number of active SGs becomes 10, at 800s becomes 15, at 1100s is 20 and at 1200s is back to 10. It should be noted that between 1100s and 1200s a congestion occurs, as the total bandwidth requested by the SGs is higher than the total system capacity. The packet delay for the EF and AF classes, however, does not change with the number of active SGs and the BE delay only changes during the congestion phase. In the congestion recovery phase the EF and AF delay is slightly lower due to the scheduler priority and the higher allocated bandwidth per SG.

Fig. 10 shows the delay for the various traffic classes varying the \( q_3 \) parameter. It is possible to note that only the BE class delay is affected, the Bandwidth loss is affected too by the \( q_3 \) value change, being about 0.82, 0.37 and 0.02 slot/frame for \( q_3 = 8, 5 \) and 2 respectively.

Finally, fig. 11 displays the CDF (Cumulative Density Function) of the relative and absolute 5-steps ahead prediction error; it can be seen that the probability that the error is within ±20 slots is approximately 80%.

5 Conclusion

The paper has addressed DBA in GEO Satellite Networks using adaptive predictive control methods. The key ideas have been: (1) to use an adaptive predictor for the input traffic flow, (2) to formulate the DBA problem as an optimal control problem with cost trading off queue occupancy and bandwidth waste, and (3) to adopt a receding-horizon strategy for bandwidth request generation. The resulting DBA scheme has highlighted appealing features in terms of (1) handling multiple services with differentiated QoS requirements; (2) low bandwidth waste; (3) high QoS (low delays and packet loss probability). Future investigations on the topic will concern: (1) adoption of more realistic “self-similar” models in place of the currently used “autoregressive model” for
traffic prediction; (2) distributed DBA schemes which take into account the presence of multiple SGs competing for the same bandwidth resource.

References


Fig. 6. Packet delay per class comparison between RHC, SPC and Fix policies.


Fig. 7. EF class packet Jitter comparison between RHC, SPC and Fix policies.

Fig. 8. RHC dynamic test


E. Del Re, R. Fantacci, G. Giambene, and W. Sergio. Performance analysis of


Fig. 11. Absolute and relative prediction error probability.


T. Le-Ngoc and S. Krishnamurthy. Performance of combined free/demand as-

