

# Video Smoothing of Aggregates of Streams with Bandwidth Constraints

Pietro Camarda, Cataldo Guaragnella, and Domenico Striccoli

**\*Abstract** - Compressed variable bit rate (VBR) video transmission is acquiring a growing importance in the telecommunication world. High data rate variability of compressed video over multiple time scales makes an efficient bandwidth resource utilization difficult to obtain. One of the approaches developed to face this problem are smoothing techniques. Various smoothing algorithms that exploit client buffers have been proposed, thus reducing the peak rate and high rate variability by efficiently scheduling the video data to be transmitted over the network.

The novel smoothing algorithm proposed in this paper, which represents a significant improvements over the existing methods, performs data scheduling both for a single stream and for stream aggregations, by taking into account available bandwidth constraints. It modifies, whenever possible, the smoothing schedule in such a way as to eliminate frame losses due to available bandwidth limitations. This technique can be applied to any smoothing algorithm already present in literature and can be usefully exploited to minimize losses in multiplexed stream scenarios, like Terrestrial Digital Video Broadcasting (DVB-T), where a specific known available bandwidth must be shared by several multimedia flows.

The developed algorithm has been exploited for smoothing stored video, although it can also be quite easily adapted for real time smoothing. The obtained numerical results, compared with the MVBA, another smoothing algorithm that is already presented and discussed in literature, show the effectiveness of the proposed algorithm, in terms of lost video frames, for different multiplexed scenarios.

**Index terms:** Video Smoothing, Bandwidth Constraints, Digital Video Transmission, Stream Aggregation.

## I. INTRODUCTION

MULTIMEDIA applications like Video on Demand (VoD), distance learning, internet video broadcast, entertainment services, etc, are assuming a growing importance in the telecommunication world. Such applications are now feasible thanks to the increasing computational capacity of modern computers together with the sustained growth of telecommunication networks bandwidth. The common aspect of all these applications is the transmission of

multimedia streams that require a sustained and relatively high Variable Bit Rate (VBR) bandwidth with stringent Quality of Service (QoS) requirements. Such high bit rate variability is characterized by a burstiness of the transmitted data over multiple time scales, that makes the optimization of network resource utilization more difficult while providing at the same time QoS guarantees, i.e., low delays and jitters, low data losses, and so on. An example of a VBR video stream is provided in Fig. 1, where a piece of the “Simpson’s” cartoon, codified with the MPEG-1 video compression algorithm, is represented.

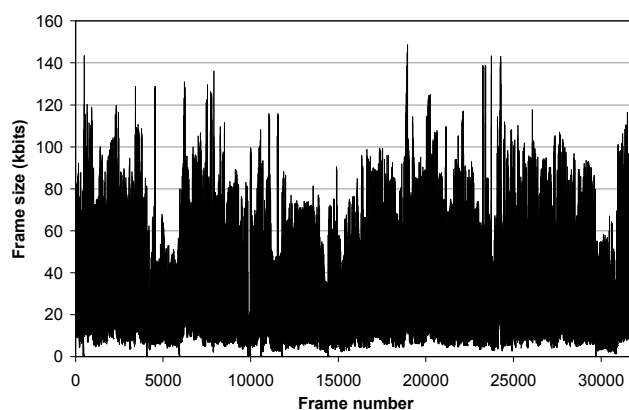


Fig. 1. 32,000 video frames of the “The Simpson’s” cartoon, codified with the MPEG-1 algorithm.

From Fig. 1 it can be noted that to guarantee a lossless transmission of a VBR video stream, a bandwidth corresponding to the stream peak rate should be assigned; nevertheless, due to the high bit rate variability characteristic of the VBR traffic, such a bandwidth assignment would result in a waste of the total available bandwidth in the almost totality of the transmission time.

To reduce the total amount of bandwidth assigned to video streams, work-ahead smoothing techniques can be exploited. They are all based on the peak rate and bit rate variability reduction of video streams, by transmitting, ahead of playback time, pieces of the same film with a constant bit rate that varies from piece to piece according to a scheduling algorithm that smoothes the video streams bursty behaviour. On the transmission side a buffer regularizes the transmission, while on the receiving side the frames are temporarily stored in a client buffer and extracted during the decoding process.

Different types of smoothing algorithms can be implemented [5]-[7]. The purpose of some algorithms is to minimize specific bandwidth parameters of VBR video

Manuscript received July 30, 2004; revised May 09, 2005 and September 05, 2005. The paper was presented in part at the Conference on Software, Telecommunications and Computer Networks (*SoftCOM*) 2004.

Authors are with Politecnico di Bari, Dip. di Elettrotecnica ed Elettronica, Via E. Orabona 4, 70125 Bari, Italy (e-mail: {camarda, guaragnella, d.striccoli} @poliba.it).

streams, e.g., the number of bandwidth increases and decreases, the number of bandwidth changes, the variability of bandwidth requirements, etc. Other types of smoothing algorithms define periodic bandwidth changes. The scheduling algorithm regulates each of the CBR bit rate values in such a way to avoid buffer overflow and underflow, ensuring a continuous playback at the client side, as will be more clear in the sequel.

Most smoothing algorithms already known, have been designed to smooth the single video stream by taking into account client smoothing buffer size, unsmoothed data and in some cases also bandwidth constraints. An example of smoothing algorithm for multiplexed traffic is reported in [8], but it is developed in the particular cases in which the clients have different buffer capacities without bandwidth constraints, or clients have no buffer capacities but some bandwidth constraints. Furthermore, in [8] only a scheduling on the data aggregate is performed, without redistributing bit rates among the single video streams.

The novel smoothing algorithm proposed in this paper substantially extends and generalizes previous approaches. Specifically, it performs data scheduling both for a single stream and stream aggregations, by taking into account available bandwidth constraints, modifying, whenever possible, the smoothing schedule in such a way to eliminate or minimizes frame losses due to available bandwidth limitations. In case of multiplexed streams the obtained cumulative schedule is subdivided among the single video streams, respecting the client buffer constraints.

The reference scenario of this paper is the Terrestrial Digital Video Broadcasting (DVB-T) system [9]-[11], where different multimedia streams can be multiplexed at server side into a single channel of constant capacity, as illustrated in Fig. 2, while a multi link scenario is out of the scope of this paper.

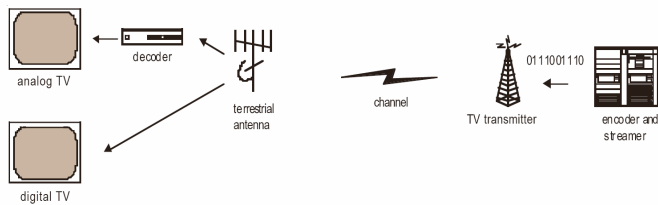


Fig. 2. DVB-T transmission scheme

The rest of the paper is structured as follows. In the next section, the relevant work on video smoothing that has been introduced in the literature and the new proposed algorithm main features are reported. In Section III the Available Bandwidth Smoothing Algorithm (ABSA) and its extended version for video stream aggregations (Extended-ABSA, E-ABSA) are formally presented and analyzed in detail. In Section IV ABSA and E-ABSA performance are analyzed in both cases of variable and constant available bandwidth, pointing out the main advantages of the proposed algorithms. Finally, in Section V some conclusions and future work are provided.

## II. BACKGROUND

### A. Smoothing Principles

Let us suppose that a VBR video stream is composed by  $N$  video frames, each of them of size  $d_i$  bytes ( $1 \leq i \leq N$ ). On the server side the smoothed video stream is transmitted, while at the client side, the smoothed video data enter a buffer whose capacity is  $b$  bytes, and the original unsmoothed video frame sequence leaves the buffer for playback. Let us now consider the client buffer model in the  $k^{th}$  time slot, with  $1 \leq k \leq N$ . To guarantee a feasible transmission, the cumulative input data to the buffer should arrive quickly enough to avoid buffer underflow, whose temporal evolution is given by the total amount of data consumed by the client buffer at discrete time  $k$ , given by:

$$D(k) = \sum_{i=1}^k d_i. \quad (1)$$

At the same time, to avoid buffer overflow, at time  $k$  the buffer client should not receive more data than:

$$B(k) = b + \sum_{i=1}^k d_i = b + D(k). \quad (2)$$

So, the cumulative smoothed data have to respect the following bounds:

$$D(k) \leq S(k) = \sum_{i=1}^k s(i) \leq B(k) \quad (3)$$

where  $s(i)$  represents the smoothed stream bit rate in the discrete time slot  $i$ , while  $S(k) = \sum_{i=1}^k s(i)$  are the cumulative smoothed data arrived to the client buffer until time slot  $k$ . The smoothed stream transmission plan will result in a number of CBR segments, and the correspondent stream evolution is given by a monotonically increasing and piecewise-linear path that lies between the  $D(k)$  and  $B(k)$  curves, as shown in Fig. 3.

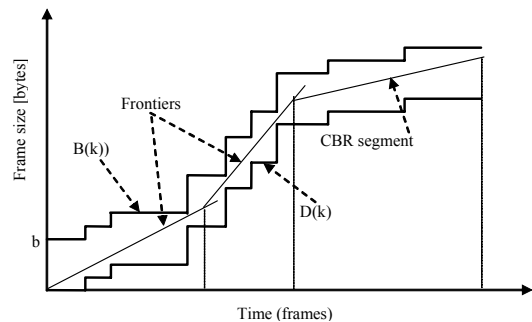


Fig. 3. An example of smoothed video stream transmission plan.

In this figure, the smoothed video stream transmission plan consists of three CBR pieces, at constant bit rate (testified by the correspondent constant segment slope). According to the definition given in [5], each CBR segment defines a run; moreover, each run can be considered as a frontier of possible starting points for the next run. A transmission plan example can be observed in Fig. 3. A “bandwidth increase” occurs when a CBR segment slope is higher than the previous segment slope, while a “bandwidth decrease” occurs when CBR segment slope is lower than the previous segment slope.

### B. The proposed algorithm

Most smoothing algorithms already known, are able to smooth the single video stream by taking into account client smoothing buffer size, unsmoothed data and in some cases also bandwidth constraint. In this paper we develop an Available Bandwidth Smoothing Algorithm (ABSA), that smoothes a single video stream, and its enhanced version (Enhanced-ABSA, E-ABSA) that smoothes stream aggregations. The smoothing algorithm proposed in this paper is thus able to smooth directly stream aggregations, at the same time taking into account the available bandwidth constraint information, in a single-link multiplexed scenario in which several video streams are aggregated in a network link with a variable or constant bandwidth constraint.

The novelty of the proposed algorithm is that it smoothes both single video streams or stream aggregations by adapting an already existing smoothing algorithm (i.e., MVBA, CBA, etc., see Section II-C for further details) to available bandwidth conditions. ABSA and E-ABSA thus preserve an existing algorithm main features at the same time taking into account bandwidth bounds. In the case of stream aggregation smoothing, the algorithm is also able to re-distribute the aggregate schedule among the single video streams in such a way that the sum of the single schedules is always under the available bandwidth constraint. Available bandwidth can be any positive-valued function of time, without any restriction. Video data are smoothed in a lossless manner, without any frame dropping or video playback pause. This kind of approach greatly helps avoiding video quality degradation or delays due to pause times.

In the application scenario considered for this paper, i.e., the DVB-T, an issue like the queueing delay is negligible; the ABSA algorithm thus only worries about adapting in a lossless way, whenever possible, an existing smoothing algorithm to available bandwidth fluctuations.

We have experimented ABSA performance for a single stream and for stream aggregations both in case of constant available bandwidth and in case of time variable available bandwidth, supposing a scenario consisting of a single network link. ABSA and E-ABSA have been applied modifying the MVBA algorithm schedule. In this way, the main MVBA features i.e., minimum smoothed peak rate and rate changes, are preserved, but thanks to ABSA and E-ABSA the obtained schedule is always under available bandwidth limit. Algorithm transmission plan is then compared with the MVBA algorithm as proposed in [4]. Let us remember that MVBA schedule itself is not able to take into account available bandwidth bounds, nor smoothing of stream

aggregations. ABSA and E-ABSA are able to bridge this gap.

As previously explained, although ABSA in this paper modifies MVBA schedule to adapt it to bandwidth availability, it is also able to act the same with transmission plans of each of the smoothing algorithms already present in literature (e.g., CBA, MCBA, etc.) by taking into account available bandwidth information for single video streams or stream aggregations.

### C. Related Work

Several smoothing algorithms, each one with its own peculiarities, have been reported in literature. To be more explicit, the Critical Bandwidth Allocation (CBA) algorithm minimizes the number of bandwidth increases at the same time obtaining the smallest peak bandwidth requirement [12].

The Minimum Changes Bandwidth Allocation (MCBA) algorithm minimizes instead both the number of bandwidth increases and decreases, together with the peak bandwidth requirement [13].

The Minimum Variability Bandwidth Allocation (MVBA) algorithm reduces the bandwidth change variability by searching, for each CBR piece, the earliest point in time in which a bandwidth increase or decrease can happen, respecting at the same time the lower and upper bounds constraints. The corresponding transmission plan gradually performs rate changes, assuring in this way the smallest variability of rate changes at the expense of a greater number of CBR pieces if compared with CBA and MCBA algorithms [4][5]. Moreover, as shown in [4], MVBA produces the lowest smoothed peak rate.

The Piecewise Constant Rate Transmission and Transport (PCRTT) algorithm divides the video flow into fixed time intervals; the transmission plan is obtained by creating a CBR piece in each time interval in such a way to avoid buffer overflow and underflow [5][14]. An enhanced version of PCRTT, called e-PCRTT, can be found in [15]. It behaves like PCRTT, but it is capable of reaching the same transmission plans of the original PCRTT algorithm with smaller smoothing buffers, or alternatively, with the same buffer sizes, it reduces the number of bandwidth changes. Furthermore, it reduces also the playback delay. Another PCRTT variant considers Dynamic Programming (DP) to build a minimum cost transmission plan [16], where the cost metric is given by the smoothing buffer size  $b$ . It can be seen that PCRTT-DP algorithm reduces peak rate requirements if compared with the original PCRTT algorithm, even if its computational complexity is higher [5][16].

Smoothing algorithms depending from available bandwidth conditions can also be found in literature. An available bandwidth dependent smoothing algorithm deriving from CBA, called Adaptable CBA algorithm, is described in [17]. It is able to adapt the CBA transmission plan to the target available bandwidth by opportunely reducing transmission frame rate through frame dropping when available bandwidth is low, and increasing it again when available bandwidth raises again. This algorithm dynamically re-evaluates CBA runs to maximize the frame rate delivered to the final user according to bandwidth availability, but frame dropping is inevitable when available bandwidth is low.

A Network Constrained Smoothing (NCS) algorithm, is considered in [18]. It takes into account available bandwidth constraints and schedules a single video stream over a server-side transmission. This simple technique considers future network traffic knowledge to derive available bandwidth. Multimedia data are then divided into equal-sized intervals in which a CBR segment is scheduled. The CBR segment values are then regulated in such a way that all schedule is always maintained under bandwidth constraints, at the same time guaranteeing the total amount of video data transmitted. This kind of approach is quite similar to the one adopted in the e-PCRTT algorithm developed in [15], whose schedule is made of CBR pieces of the same length.

Another example of bandwidth dependent smoothing algorithm can be found in [19]. The schedule derived from an MPEG-coded video stream is maintained under the network bandwidth constraint simply by introducing a pause playback or dropping less significant frames (i.e., B-frames) at the client side in a specified short time period.

A bandwidth dependent smoothing algorithm is presented in [20]. It exploits network calculus to optimize the client buffer size, playback delay and look-ahead delay in such a way to generate a lossless video stream schedule respecting particular traffic envelopes, i.e., curves representing the maximum traffic that can be sent to the network. This algorithm operates by defining specific traffic bound curves  $\sigma(t)$ , called “traffic envelopes” or “arrival curves”, representing the maximum amount of traffic that can be sent to the network. This data, together with the original unsmoothed traffic  $d(t)$ , are utilized by the smoother to minimize the smoothing client buffer size  $b$ , the playback delay  $D$  and the total delay  $T$ , expressed as the sum of  $D$  and the look-ahead delay in such a way to obtain a lossless schedule  $s(t)$  (see Section II-1). The look-ahead delay is computed as the time necessary to prefetch all data to be scheduled [7][20]. Proof of optimality of the obtained schedule  $s(t)$  are given when the traffic envelope is a constant rate curve, that is, a constant bandwidth constraint  $\sigma(t) = C \cdot t$ , where  $C$  is the maximum available bandwidth. New upper and lower bounds,  $D(k)$  and  $B(k)$  respectively, are calculated taking into account the traffic envelope  $\sigma(t)$ . This smoothing technique can thus be applied in combination with other existing smoothing algorithms like the one illustrated in [4] to further minimize other metric, like number of bandwidth changes or rate variability.

A Rate Constrained Bandwidth Smoothing (RCBS) is also presented in [21]. It is explicitly thought for delivery of interactive video streams. There are situations in which rate consumption of the client can be altered because of interactivity with the server, like in the case of classical VCR commands (i.e., rewind or playback pause). The RCBS algorithm proposed in [21] minimizes the amount of buffering needed by smoothing when a maximum constant rate constraint is given, simply by prefetching video data only when the rate constraint is violated, and leaving the original unsmoothed data unchanged when they maintain under the bandwidth constraint. In this way, if compared with the classical smoothing techniques previously illustrated (CBA, MCBA, MVBA, etc.), the buffering needed for smoothing is

greatly reduced and client buffers can store much more data for VCR functionalities like stop, pause, rewind and examine operations. The so called VCR-windows size, that is, the time interval in which the VCR operations can be made on video data, is thus increased.

Some kind of smoothing is also taken into account in [22], where multimedia data are previously packetized into data units and then stored in a file or in a video server for transmission. Various scenarios are presented where data units are transmitted across a lossy network with delay constraints, in such a way to minimize an error-cost function that takes into account packet losses and delays. The authors introduce a kind of rate control to smooth the instantaneous transmission rate. The mechanism proposed is very similar to the rate control mechanisms adopted in the standard video decoders. It consists on reducing data rate leaving the transmission buffer when it is almost empty and increasing it when the buffer is almost full, to avoid buffer underflow and overflow respectively. The data rate leaving the buffer can so be regulated to not exceed a bandwidth threshold.

Another example of smoothing algorithm taking into account also multiplexed network traffic together with bandwidth limitations can be found in [8]. It is developed in the particular cases in which the clients have different buffer capacities but there is no bandwidth constraints, or clients have no buffer capacities but some bandwidth constraints.

The choice among the mentioned algorithms depends on what aspect of data transmission has to be optimized among the peak rate, the number, the variability and the periodicity of bandwidth changes, and if to consider available network bandwidth as further constraint for schedule. In particular the MVBA smoothing technique proposed in [4] because of its peculiarity can significantly improve network performance, that mainly depends on peak rate and burstiness of the admitted flows [4].

As an example of the application of the smoothing algorithm, in Fig. 4 the bit rate of the “Simpson’s” film is reported smoothed with a client buffer of 1 Mbyte, utilizing MVBA.

All the smoothing techniques mainly apply to stored video traffic, where all source video data are a priori known and can be optimally scheduled “off-line” applying one of algorithms described above. The optimality of the off-line algorithms derive from the a priori knowledge of the entire video data to be scheduled. Nevertheless, there is a growing number of VBR live interactive video applications (videoconferences, video news, etc.) that could require smoothing algorithms to reduce bit rate variability, at the cost of a short playback delay. Smoothing techniques have then to be applied on-the-fly, in an “on-line” manner, taking into account a variety of situations, like client interactions and heterogeneous client buffer sizes [23]-[25]. Online smoothing algorithms perform generally worse than the corresponding off-line algorithms: the latter have knowledge of the entire video stream data structure, thus performing an optimal transmission plan. The algorithm proposed in this paper has been developed and analyzed in this context of off-line video smoothing but it can be quite easily adapted to online smoothing scenarios.

ABSA and E-ABSA substantially differ from all the smoothing algorithms already cited in this Section. ABSA acts

directly on the video data without pre-processing them; for this reason the approach is very different to what presented in [22], that smoothes already packetized multimedia data, thus acting at a lower protocol layer. Since the ABSA purpose is to smooth video data without frame losses, it can be considered a more efficient approach if compared with the one illustrated in [19]. It takes also into account both a finite smoothing buffer capacity and bandwidth constraints, so it generalizes in some way the cases proposed in [8]. Since ABSA can be applied to any available bandwidth condition, it is more general than the approach developed in [18], where both the stream schedule and the available bandwidth are constant in time intervals of fixed constant size. It is also more general than the approaches followed in [21],[22] and [8], where a constant bandwidth constraint is considered. ABSA and E-ABSA follow an approach quite similar to the one considered in [20], where any available bandwidth curve  $\sigma(t)$  is considered. ABSA can be considered as an alternative approach to [20], since it not necessarily considers minimum values of client smoothing buffers and playback delays, but it tries the same to produce a lossless smoothing schedule that respects available bandwidth limitation, thus taking into account a wide variety of different scenarios. No limitations on smoothing buffer size and/or playback delays occur to let ABSA and E-ABSA work. As an example, an end user that receives a smoothed video schedule could tolerate a relatively large playback delay with a very little buffer size, or very stringent playback delays but with a relatively large buffer size.

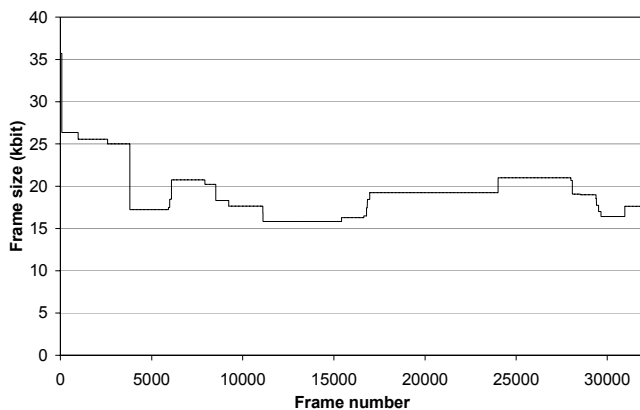


Fig. 4. 32,000 video frames of the “Simpson’s” cartoon, smoothed with the MVBA smoothing algorithm (buffer size 1,024 kbytes).

### III. ABSA SMOOTHING ALGORITHM

In this section ABSA smoothing algorithm is proposed and analyzed, taking into account not only the parameters of the smoothing algorithms usually proposed in literature, i.e., buffer size and unsmoothed data, but also available bandwidth information.

ABSA algorithm has been tested both when the available bandwidth is variable in time and when it assumes a constant value a priori established. A possible application of the first proposed scenario is when a stream aggregation, whose occupied bandwidth varies in time, is already running into the network link. Since the channel capacity is constant, a variable

residual available bandwidth remains for other smoothed stream transmissions. An application of the second scenario considers a single video stream or a stream aggregation smoothing when a constant available bandwidth is reserved. An example of such a scenario arises in Terrestrial Digital Video Broadcasting (DVB-T) systems, where specific bandwidth resources are shared among several video streams to efficiently provide multimedia contents to final users. A smoothing technique taking into account available bandwidth can thus be used for optimizing video schedules without wasting bandwidth.

ABSA can be fruitfully exploited for stored video applications, both for a single video stream or an aggregation of streams. The smoothing algorithm is based on the a priori knowledge of all unsmoothed data, on available bandwidth information, and on the client smoothing buffer size.

The a priori knowledge of bandwidth resources is an important requirement. In this paper the algorithm has been applied to stored video and thus a knowledge of available bandwidth resources for all stream duration is needed. Such information is obvious in case of constant available bandwidth, while for variable available bandwidth, more complex situations can arise. In fact, in this last case, this hypothesis can be verified only in specific situations; for example, when available bandwidth fluctuates in time due the presence in the channel of other independent stored video streams, residual available bandwidth depends on video streams temporal evolution but can a priori be evaluated, situation not so far from reality in practice; examples of knowledge of future network loading for multimedia streams can be found in [18]. In general, the residual bandwidth resources could be derived using bandwidth estimation techniques, but the way in which this can be done is out of the scope of this paper.

#### A. ABSA for single video stream

In this section ABSA is applied to a single video stream when the available bandwidth is supposed to be known a priori. Let us suppose to analyze the data transmission of a single video stream of length  $N$  frames. Given a client buffer whose capacity is  $b$  bytes, smoothing algorithms calculate the cumulative data transmission plan  $S(k)$  in time slot  $k$  ( $1 \leq k \leq N$ ), satisfying the two bounds  $D(k)$  and  $B(k)$  expressed by (1) and (2). Let us now suppose to know the smoothing buffer size  $b$ , the temporal evolution of the unsmoothed data and the temporal evolution of the available bandwidth  $w(k)$  ( $1 \leq k \leq N$ ). The smoothed schedule  $s(k)$  has to respect the available bandwidth bound  $w(k)$  for each ( $1 \leq k \leq N$ ):

$$s(k) \leq w(k) \quad \forall 1 \leq k \leq N.$$

That is:

$$S(k) - S(k-1) \leq w(k) \Rightarrow S(k) \leq w(k) + S(k-1) \quad (4)$$

The condition (4) modifies the upper bound defined in (2), since the cumulative schedule has to verify both the

limitations:

$$S(k) \leq w(k) + S(k-1) \text{ and } S(k) \leq B(k).$$

Thus, posing:

$$U_s(k) = \min[B(k), w(k) + S(k-1)]$$

it follows:

$$S(k) \leq U_s(k), \forall 1 \leq k \leq N$$

The bound  $U_s(k)$  depends on the available bandwidth  $w(k)$ , the transmission plan  $S(k)$  and the  $B(k)$  curve.

A feasible transmission plan  $S$  thus has to verify the condition:

$$D(k) \leq S(k) \leq U_s(k) \quad \forall 1 \leq k \leq N \quad (5)$$

The main problem of the upper bound  $U_s(k)$  is that it depends on the scheduled data  $S(k-1)$  calculated at the time step  $k-1$ , that can not be a priori known since it in turn depends on  $U_s(k-1)$ . The two constraints (5) can thus be verified only after the calculation of  $S(k)$  in each time slot  $k$ , but this can be done if we eliminate the dependence from  $S$  in the upper bound  $U_s$ .

So let us define the new upper bound:

$$U'(k) = \min[B(k), U'(k-1) + w(k)]$$

with the initial condition  $U'(1) = B(1)$ . Let us observe that the  $U'(k)$  curve depends only on the available bandwidth  $w$  in the time slot  $k$  and the  $B(k)$  curve. Nevertheless, it has to be proved that the function  $U'(k)$  is effectively an upper bound for  $S(k)$ . At this aim, we demonstrate the following proposition.

*Proposition:*

If  $S(k)$  is a feasible transmission, then the curves  $U'$  and  $U_s$  satisfy the relation:

$$U'(k) \geq U_s(k) \quad \forall 1 \leq k \leq N \quad (6)$$

*Proof:*

It is valid that

$$U_s(1) = U'(1) = B(1)$$

Let us proceed by induction and suppose that (6) is valid in  $k$ . We have to proof that (6) is valid in  $k+1$ . It will be:

$$\begin{aligned} U'(k+1) &= \min[B(k+1), U'(k) + w(k+1)] \geq \\ &\min[B(k+1), U_s(k) + w(k+1)] \geq \\ &\geq \min[B(k+1), S(k) + w(k+1)] = U_s(k+1) \end{aligned}$$

exploiting the hypothesis  $U'(k) \geq U_s(k)$  for the first inequality and the hypothesis  $U_s(k) \geq S(k)$  for the second inequality. This demonstrates (6). ■

Given (6) and calling  $S$  the global transmission plan as obtained by the smoothing algorithm, we can assert that:

$$D \leq S \leq U_s \leq U'. \quad (7)$$

In other words, the function  $U'$  is effectively an upper

bound for  $S$ , but with the advantage that it is independent from  $S$  itself. So we can apply the MVBA smoothing algorithm as described in [4] with the bounds  $D$  (lower bound) and  $U'$  (upper bound):

$$D \leq S \leq U'. \quad (8)$$

As explained in Section II, in this paper MVBA has been chosen to take into account peak rate and rate variability minimization, even if ABSA technique illustrated in this section can be applied with each of the smoothing algorithms already present in literature.

If a time slot is found in which  $D > U'$ , the corresponding transmission plan  $S$  will not be feasible and the smoothing algorithm can not be applied due to the strong limitation in available bandwidth. Otherwise, the transmission plan  $S$  is calculated. Then the real upper bound  $U_s(k)$  can be derived and (5) can be verified, that is the purpose of ABSA.

In particular, let us suppose to have calculated  $S$  through (8) and also that (5) is satisfied. That is,  $D(k) \leq S(k) \leq U'(k) \quad \forall 1 \leq k \leq N$  where the transmission plan  $S(k)$  has been obtained through the MVBA smoothing algorithm [4]. In this case, ABSA behaves exactly like MVBA. Let us suppose, instead, that after the calculation of  $U_s(k)$  (5) is not effectively verified for each  $k$ . This can happen since the transmission curve  $S$  is obtained by applying MVBA with the  $U'$  upper bound, and not  $U_s$ . ABSA is then able to adjust CBR segment slopes of  $S$  (and consequently the constraint  $U_s$ ) in such a way that (5) is verified for each  $k$ , at the same time maintaining the main MVBA features.

To better explain this concept, let us suppose to have verified (5) for  $1 \leq k \leq k_1 - 1$ , and that in time slot  $k_1$  (5) is not verified. We can demonstrate that a feasible transmission plan  $S(k_1)$  exists that verifies (5) in the time slot  $k_1$ . In particular, it can be demonstrated the proposition:

*Proposition:*

The schedule  $S(k_1) = U'(k_1)$  surely verifies the constraints:

$$D(k_1) \leq S(k_1) \leq U_s(k_1).$$

*Proof:*

Since (8) is verified in each time slot  $k$ ,  $S(k_1) = U'(k_1)$  is a feasible transmission plan with the bounds  $D(k_1)$  and  $U'(k_1)$ . Given this assumption, it is immediately verified that:

$$S(k_1) = U'(k_1) \geq D(k_1) \quad (9)$$

Furthermore:

$$U_s(k_1) \Big|_{S(k_1)=U'(k_1)} = \min[B(k_1), U'(k_1-1) + w(k_1)] = U'(k_1)$$

exploiting the definition of  $U'(k_1)$ . It derives that  $S(k_1) = U_s(k_1)$ . So (5) is fully verified in  $k_1$ . ■

Summarizing, surely a transmission plan  $S(k_1)$  verifying (5) in  $k_1$  can be found by choosing:

$$S(k_1) = U'(k_1). \quad (10)$$

Thus starting from the initial value of  $S(k_1)$  obtained through MVBA (not verifying (5)), to maintain the ABSA schedule the closest possible to MVBA schedule the value of  $S(k_1)$  is iteratively increased since it verifies (5) in  $k = k_1$ . We know that this last condition can be surely reached at worst exploiting (10). After have performed the calculation of  $S(k_1)$ , ABSA applies again MVBA algorithm in the time interval  $[k_1+1, N]$  exploiting (8) and then verifying (5) through the same procedure previously illustrated.

In Fig. 5 the ABSA algorithm is formally presented. In line 1, the main smoothing parameters are defined; then, the bounds  $D(k)$ ,  $B(k)$  and  $U'(k)$  are built along the entire time window duration ( $1 \leq k \leq N$ ). The two flags *flag1* and *flag2* have specific functions: *flag1* is set to control the condition  $D \leq S \leq U'$ , while *flag2* controls if the smoothing scheduling, from line 12 to line 36, has to be performed in the next step or not. In the latter case, the transmission plan will be calculated according to the MVBA smoothing algorithm, in line 35. Proceeding with the algorithm analysis, if the condition  $D \leq U'$  is not verified, the smoothing algorithm is not feasible and cannot be performed, due to heavy bandwidth limitations. In such a case, other procedures have to be taken into account, like unsmoothed stream rate reduction, discarding of less important data packets, etc. If, on the other hand, the condition  $D \leq U'$  is verified, the algorithm calculates the transmission plan  $S$ , by exploiting the MVBA smoothing algorithm, in line 13, while in line 15 verifies the condition  $D \leq S \leq U_s$  in each  $k$ . If a value of  $k = k_1$  is found in which (5) is not verified, the smoothing algorithm enters the loop from line 19 to line 33, that increases the bit rate  $S(k)$  in such a way so as to verify (5) in  $k = k_1$ , at the same time preserving the MVBA transmission plan schedule in  $k = k_1$ , as previously explained. The choice of increasing  $S(k)$  in  $k = k_1$  is justified because it has been demonstrated that if the transmission plan is feasible (that is,  $D \leq U'$ ), the transmission plan  $S(k_1) = U'(k_1)$  surely verifies (5) in  $k_1$ .

The increment of  $S(k_1)$  is performed by progressively modifying the bound  $B(k_1)$ . The two bound curves  $D$  and  $U'$  are calculated again, starting from  $k = k_1$ , and  $B(k_1)$  is calculated as the arithmetic mean of  $U_s(k_1)$  and  $S(k_1)$ . If the condition in line 21 is not satisfied,  $S(k_1)$  is increased through line 24 to make the transmission plan feasible by approximating  $S(k_1)$  to the upper bound  $U'(k_1)$ . Elsewhere, if the above condition is satisfied,  $U_s(k_1)$  is decreased through line 28. This step is very important, since it is necessary to verify the condition of the best approximation of the smoothed transmission plan to the MVBA smoothing curve in  $k = k_1$ . This procedure is repeated until  $S(k_1)$  and  $U_s(k_1)$  coincide; after  $S(k_1)$  is found, the MVBA smoothing algorithm is performed again in line 13, starting from  $k = k_1$  (since for  $k < k_1$  the ABSA transmission plan has been already found). The loop from line 19 to line 33 is then

iteratively verified incrementing step by step the time slot  $k$ , until the end of the observation window is reached.

```

1  Assign  $b, D(k), w(k)$  in each  $1 \leq k \leq N$ 
2   $D(k) = \sum_{i=1}^k d(i); B(k) = \min[D(k-1)+b, D(N)];$ 
3  Calculate bound  $U'(k)$  in each  $1 \leq k \leq N$ 
4   $flag1 = 0; flag2 = 0; k = 1;$ 
5  IF exists  $k = \bar{k}$  such that  $U'(k) \leq D(k)$ 
6      ABSA algorithm NOT FEASIBLE
7      EXIT
8  ELSE
9      IF ( $flag1 \neq 1$  AND  $k < N$ )
10          $U^{(1)} = U'$ 
11          $k = k + 1;$ 
12         IF  $flag2 \neq 2$ 
13              $S = OPTIMAL\_MVBA[D, U^{(1)}]$  starting from  $k$ 
14             Calculate bound  $U_s^{(1)} = U_s$ 
15             IF  $\min(U_s^{(1)} - S) \geq 0$  AND  $\min(S - D) \geq 0$ 
16                  $flag1 = 1$ 
17             END
18              $flag2 = 0$ 
19             IF  $\left| \frac{S(k) - U_s^{(2)}(k)}{U_s^{(2)}(k)} \right| < \varepsilon$  AND  $flag1 \neq 1$  AND  $flag2 \neq 2$ 
20                 Calculate  $U^{(3)} = U'$ 
21                 IF  $\min(U^{(3)} - D) < 0$ 
22                      $flag2 = 1$ 
23                     Assign  $B(k) = \frac{U_s^{(2)}(k) + S(k)}{2}$ 
24                     Increase  $S(k) = B(k)$ 
25                 ELSE
26                     IF  $flag2 = 1$ 
27                         Assign  $B(k) = \frac{U_s^{(2)}(k) + S(k)}{2}$ 
28                         Decrease  $U_s^{(2)}(k) = B(k)$ 
29                     ELSE
30                          $flag1 = 2$ 
31                     END
32                 END
33             END
34         ELSE
35             Exists  $S = OPTIMAL\_MVBA[D, U^{(1)}]$ 
36         END
37     END
38 END
```

Fig. 5. The ABSA smoothing algorithm, in its formal presentation.

### B. E-ABSA for stream aggregation

The ABSA algorithm can also be fruitfully applied to smooth a stream aggregation flowing into the network link by exploiting the same basic concepts of a single video stream smoothing. The generalization of the ABSA algorithm referred to a flow aggregation is called for simplicity Extended-ABSA algorithm (E-ABSA). The E-ABSA schedules the aggregate bit rate like the single stream case, given a smoothing buffer of size  $b$  and the original unsmoothed aggregate data, at the same time preserving the MVBA features and considering available bandwidth information. Then it redistributes the so obtained transmission plan among each of the streams composing the aggregate. This is a crucial point, since each video stream cumulative transmission plan has to be monotonically increasing, at the same time respecting the buffer overflow and underflow constraints (as can be noted in Fig. 3). It thus can not assume all possible values. E-ABSA redistributes the aggregate transmission plan strictly verifying this condition for each video stream.

Let us suppose to have an aggregation of  $N_u$  multiplexed video streams, of different types. Each of them has total length  $N$  video frames. The same definitions given for the

$j^{th}$  video stream hold also in this case, i.e., the cumulative data transmission plan is  $S_j(t)$ , while the total amount of data consumed by the client buffer at discrete time  $k$  is  $D_j(k)$  and the maximum amount of data that the client buffer can receive is  $B_j(k)$ , according to (1) and (2) respectively, for each  $1 \leq j \leq N_u$ . Also (3) must be verified for each  $j$ . Let us define in each discrete frame time  $k$  the following quantities:

$$D(k) = \sum_{j=1}^{N_u} D_j(k), \quad S(k) = \sum_{j=1}^{N_u} S_j(k), \quad B(k) = \sum_{j=1}^{N_u} B_j(k). \quad (11)$$

It is immediately verified that:

$$D(k) \leq S(k) \leq B(k). \quad (12)$$

Given the available bandwidth information  $w(k)$ , the E-ABSA algorithm tries to calculate the transmission plan  $S_{opt}(k)$  exploiting the same technique illustrated in Section III-1. In other words, E-ABSA tries to smooth the flow aggregation taking into account the available bandwidth, while at the same time trying to maintain minimum variability among bandwidth changes like the MVBA algorithm. Then, E-ABSA is able to derive, from the schedule  $S_{opt}(k)$ , the single cumulative transmission plan  $S_j(t)$  at the same time verifying that:

$$S(k) = \sum_{j=1}^{N_u} S_j(k) = S_{opt}(k). \quad (13)$$

and:

$$S_j(k) \leq S_j(k+1) \quad \forall 1 \leq j \leq N_u. \quad (14)$$

To reach this purpose let us suppose to have applied ABSA illustrated in section III-1, exploiting the bounds  $D(k)$  and  $B(k)$  as defined in (11) and obtaining  $S_{opt}(k)$ . E-ABSA subdivides  $S_{opt}(k)$  among all the streams in such a way so as to verify the conditions (3), (13) and (14), which is of paramount importance to guarantee that all bit rates  $s_j(k)$  are positive in each frame time and for each  $j$ . To guarantee (14), E-ABSA starts from the end of the observation window, where:

$$S_j(N) = D_j(N) \quad \forall 1 \leq j \leq N_u \quad (15)$$

This last condition is obvious, since for each stream, at the end of the observation window the total amount of scheduled data has to be consumed by the client buffer. The algorithm then proceeds backward in time for decreasing values of  $k$ , by posing:

$$S_j^{\max}(k) = \min(B_j(k), S_j(k+1)) \quad (16)$$

It can be easily noted that the (16) verifies (14). Then the aggregate cumulative transmission plan  $S^{\max}(k) = \sum_{j=1}^{N_u} S_j^{\max}(k)$  and the difference  $diff(k) = S^{\max}(k) - S_{opt}(k)$  is calculated. The purpose of E-ABSA is to adapt the cumulative transmission plans of all video streams in order to obtain  $diff(k) = 0$ , thus verifying (13). In the very particular case in which  $diff(k) = 0$  is immediately verified, nothing has to be done, since:

$$S_j(k) = S_j^{\max}(k)$$

and consequently:

$$S_j^{\max}(k) \leq B_j(k); \quad S_j^{\max}(k) \leq S_j(k+1)$$

(see (16)).

Elsewhere, two cases can happen.

*Case 1:  $diff(k) > 0$ .*

Then we have:

$$\sum_{j=1}^{N_u} S_j^{\max}(k) \geq S_{opt}(k).$$

Each  $S_j^{\max}(k)$  of the single video stream is thus decreased according to the relation:

$$S_j(k) = S_j^{\max}(k) - diff(k) \frac{S_j^{\max}(k) - D_j(k)}{S^{\max}(k) - D(k)}. \quad (17)$$

▪ *Constraint  $S_j^{\max}(k) \geq D_j(k)$ .*

If:

$$S_j^{\max}(k) = B_j(k),$$

then:

$$S_j^{\max}(k) = B_j(k) \geq D_j(k)$$

(otherwise the smoothing algorithm for the  $j^{th}$  flow would not be feasible).

If instead:

$$S_j^{\max}(k) = S_j(k+1),$$

it will be:

$$S_j^{\max}(k) = S_j(k+1) \geq D_j(k+1) \geq D_j(k)$$

given that  $D_j(k) \leq D_j(k+1)$ .

▪ *Constraint  $S_j(k) \leq B_j(k)$ .*

Given:

$$S_j^{\max}(k) - D_j(k) \geq 0$$

as previously stated, and also:

$$S^{\max}(k) - D(k) \geq 0,$$

through (17) each  $S_j^{\max}(k)$  is effectively decreased. Thus surely:

$$S_j(k) \leq B_j(k)$$

and

$$S_j(k) \leq S_j(k+1).$$

Let us now calculate  $S(k) = \sum_{j=1}^{N_u} S_j(k)$  exploiting (17). It



follows:

$$\begin{aligned} \sum_{j=1}^{N_u} S_j(k) &= \sum_{j=1}^{N_u} S_j^{\max}(k) - \text{diff}(k) \frac{\sum_{j=1}^{N_u} S_j^{\max}(k) - \sum_{j=1}^{N_u} D_j(k)}{S^{\max}(k) - D(k)} = \\ &= \sum_{j=1}^{N_u} S_j^{\max}(k) - \text{diff}(k) = S_{\text{opt}}(k) \end{aligned} \quad (18)$$

It can be proved that in such situation  $S_j(k) \geq D_j(k)$ . In fact, let us consider again (17), that can be rewritten as follows:

$$\frac{S_j^{\max}(k) - S_j(k)}{S_j^{\max}(k) - D_j(k)} = \frac{\text{diff}(k)}{S^{\max}(k) - D(k)} = \frac{S^{\max}(k) - S_{\text{opt}}(k)}{S^{\max}(k) - D(k)}.$$

The condition  $S_j(k) \geq D_j(k)$  is verified if and only if:

$$\frac{S_j^{\max}(k) - S_j(k)}{S_j^{\max}(k) - D_j(k)} \leq 1$$

that means:

$$S^{\max}(k) - S_{\text{opt}}(k) \leq S^{\max}(k) - D(k) \Rightarrow S_{\text{opt}}(k) \geq D(k).$$

This last inequality is obvious, since the schedule  $S_{\text{opt}}(k)$  derives from the application of ABSA, that verifies (3). Thus, the conditions (3) and (14) hold if  $\text{diff}(k) > 0$ . ■

Case 2:  $\text{diff}(k) < 0$ .

We have:

$$\sum_{j=1}^{N_u} S_j^{\max}(k) < S_{\text{opt}}(k).$$

The only solution is to increase the sum  $\sum_{j=1}^{N_u} S_j^{\max}(k)$  until  $\text{diff}(k) = 0$ . This step can be obtained by imposing:

$$S_j^{\max}(k) = B_j(k) \quad \forall 1 \leq j \leq N_u \quad (19)$$

In this way it is surely verified that:

$$\sum_{j=1}^{N_u} S_j^{\max}(k) = \sum_{j=1}^{N_u} B_j(k) = B(k) > S_{\text{opt}}(k).$$

Thus, E-ABSA can be applied for  $\text{diff}(k) > 0$  as explained in Case 1, redistributing all the cumulative data  $S_j(k)$  through (17). The (19) guarantees the conditions (3) and (13), but not (14).

To verify (14), let us suppose:

-  $\text{diff}(k) \geq 0$ ,  $k = \bar{k} + 1$ ; this condition is surely verified for  $k = N$  where  $\text{diff}(N) = 0$ . Moreover,  $S_j(k)$  is derived by (17).

-  $\text{diff}(\bar{k}) < 0$ .

■ Step 1

E-ABSA calculates:

$$S'_j(\bar{k}) = B_j(\bar{k}) - (B_j(\bar{k}) - S_{\text{opt}}(\bar{k})) \frac{B_j(\bar{k}) - D_j(\bar{k})}{B(\bar{k}) - D(\bar{k})}$$

by (17) and (19).

■ Step 2

E-ABSA assigns:

$$S_j^{\text{new}}(\bar{k} + 1) = \max(S_j(\bar{k} + 1), S'_j(\bar{k})) \quad (20)$$

and calculates the new difference:

$$\text{diff}^{\text{new}} = S^{\text{new}}(\bar{k} + 1) - S_{\text{opt}}(\bar{k} + 1) \quad (21)$$

where  $S^{\text{new}}(\bar{k} + 1) = \sum_{j=1}^{N_u} S_j^{\text{new}}(\bar{k} + 1)$ .

■ Step 3

E-ABSA redistributes the new cumulative smoothed data  $S_j(\bar{k} + 1)$  as follows:

$$S_j(\bar{k} + 1) = S_j^{\text{new}}(\bar{k} + 1) - \text{diff}^{\text{new}} \frac{S_j^{\text{new}}(\bar{k} + 1) - S_j(\bar{k})}{S^{\text{new}}(\bar{k} + 1) - S_{\text{opt}}(\bar{k})} \quad (22)$$

This procedure starting from (19) and ending with (22) is repeated for increasing values of  $k$  ( $k = \bar{k} + 2$ ,  $k = \bar{k} + 3$ , etc.) until a frame time  $k$  for which  $\text{diff}(k) = 0$ . As previously said, this frame time surely exists ( $k = N$ ), so that the method previously explained surely comes to an end.

The formal presentation of E-ABSA is illustrated in Fig. 6.

```

1  Assign  $b, D(k), w(k)$  in each  $1 \leq k \leq N$ 
2   $D_j(k) = \sum_{i=1}^k d_j(i)$ ;  $B_j(k) = \min[D_j(k-1) + b_j, D(N)]$  for  $1 \leq j \leq N_u$ 
3   $D(k) = \sum_{i=1}^{N_u} D_j(k)$ ;  $B(k) = \sum_{i=1}^{N_u} B_j(k)$ 
4   $S_{\text{opt}}(k) = \text{ABSA\_algorithm}(B, D, w)$ 
5  FOR  $k = N-1:1$ 
6     $S_j^{\max}(k) = \min(B_j(k), S_j(k+1))$  for  $1 \leq j \leq N_u$ 
7     $S^{\max}(k) = \sum_{i=1}^{N_u} S_j^{\max}(k)$ ;  $\text{diff}(k) = S^{\max}(k) - S_{\text{opt}}(k)$ 
8    IF  $\text{diff}(k) \geq 0$ 
9       $S_j(k) = S_j^{\max}(k) - \text{diff}(k) \frac{S_j^{\max}(k) - D_j(k)}{S^{\max}(k) - D(k)}$ 
10     ELSE IF  $\text{diff}(k) < 0$ 
11       REPEAT
12          $S_j^{\max}(k) = B_j(k)$   $1 \leq j \leq N_u$ 
13          $S^{\max}(k) = \sum_{i=1}^{N_u} S_j^{\max}(k)$ 
14          $S'_j(k) = B_j(k) - (B_j(k) - S_{\text{opt}}(k)) \frac{B_j(k) - D_j(k)}{B(k) - D(k)}$   $1 \leq j \leq N_u$ 
15          $S_j^{\text{new}}(k+1) = \max(S_j(k+1), S'_j(k))$   $1 \leq j \leq N_u$ 
16          $S^{\text{new}}(k+1) = \sum_{i=1}^{N_u} S_j^{\text{new}}(k+1)$ ;  $\text{diff}^{\text{new}} = S^{\text{new}}(k+1) - S_{\text{opt}}(k+1)$ 
17          $S_j(k+1) = S_j^{\text{new}}(k+1) - \text{diff}^{\text{new}} \frac{S_j^{\text{new}}(k+1) - S_j(k)}{S^{\text{new}}(k+1) - S_{\text{opt}}(k)}$ 
18       UNTIL  $\text{diff}(k) = 0$ 
19     END
20   END
21 END
```

Fig. 6. The E-ABSA smoothing algorithm for stream aggregation.

The ABSA algorithm presented in Fig. 5 and applied to the stream aggregation is performed in line 4.

#### IV. ABSA PERFORMANCE EVALUATION

In this section some numerical results derived from the ABSA application are provided, to test the effectiveness of the proposed algorithm in both cases of variable and constant

available bandwidth. Algorithm performance has been tested for a single video stream and stream aggregations, using different buffer sizes. Let us remember that the smoothing buffer size influences the transmission plan temporal behaviour and peak rate: the smaller the buffer size, the higher is the peak rate. Nevertheless, ABSA and E-ABSA are always able to maintain the scheduled transmission plan under available bandwidth constraint whatever is the chosen smoothing buffer size.

The variable available bandwidth information has been derived as follows. Twelve video streams of different types, chosen from a larger set of MPEG-1 coded video sources have been aggregated. The chosen films are: the “Asterix”, “James Bond: Goldfinger”, “Jurassic Park”, “Mr. Bean”, “Simpson’s”, “Terminator 2” and “Star Wars” movies, a MTV video clip, a car race and a soccer final. Flow aggregation has been performed randomly choosing the video stream starting points and deriving the total bandwidth occupied by stream aggregation simply by adding the number of bits contained in each video streams frame, in each discrete time unit given by a frame transmission time. In this case, the obtained bandwidth is expressed in bit/frame. To guarantee the total superimposition of all the chosen smoothed streams it has been supposed that when a video stream of a given type finishes, it starts again immediately. Supposed a constant channel capacity  $C$ , the available bandwidth has been derived simply by subtracting to  $C$  the bandwidth exploited by stream aggregation previously calculated. In this case, the available bandwidth information has been a priori derived, supposing to know in advance all the flow aggregation information in each frame time. This aggregation scenario is useful to analyze in detail the ABSA and E-ABSA behaviour.

Let us suppose to first apply the ABSA algorithm for a single video stream smoothing. Let us recall that the main purpose of the proposed smoothing algorithm is to fully exploit the available bandwidth information, at the same time preserving as much as possible the MVBA smoothing characteristics, that is, the maximum reduction possible in bandwidth change variability and peak rate. A first comparison among the ABSA and MVBA smoothing techniques is illustrated in Fig. 7.

A piece of the “Terminator 2” film, MPEG-1 codified and of total length  $N = 40,000$  video frames, has been smoothed with both the MVBA and the ABSA algorithms, both employing a smoothing buffer of 256 kbytes. Fig. 7 highlights the main differences between them in a time window of 32,000 video frames. In particular, there is a first strong available bandwidth reduction, beginning from the 11,267<sup>th</sup> frame time until the 13,100<sup>th</sup> frame time, due to high bandwidth requirements by flow aggregation already present in the network link. As it can be seen from Fig. 7, during this period the ABSA smoothing algorithm follows perfectly the available bandwidth curve while the MVBA algorithm crosses it, thus testifying a frame loss that occurs until the available bandwidth curve raises again. Furthermore, after the ABSA application no losses have been experienced throughout the video stream duration. When available bandwidth is high, ABSA and MVBA behave in the same way, according to ABSA capacity to preserve MVBA main features. Algorithm comparison can be better appreciated by observing Fig. 8, in

which the critical time interval, in which the available bandwidth falls down, has been zoomed.

In Fig. 8 the frame time interval ranging from the 10,000<sup>th</sup> until 15,000<sup>th</sup> video frame has been better highlighted. It can be noted that ABSA accommodates the smoothed stream CBR pieces both before and after the bandwidth reduction zone to completely avoid frame losses. During the bandwidth reduction time interval ABSA perfectly follows the available bandwidth curve. When the available bandwidth raises again ABSA increases the CBR segment values, recovering from low bandwidth schedule performed in the previous time interval.

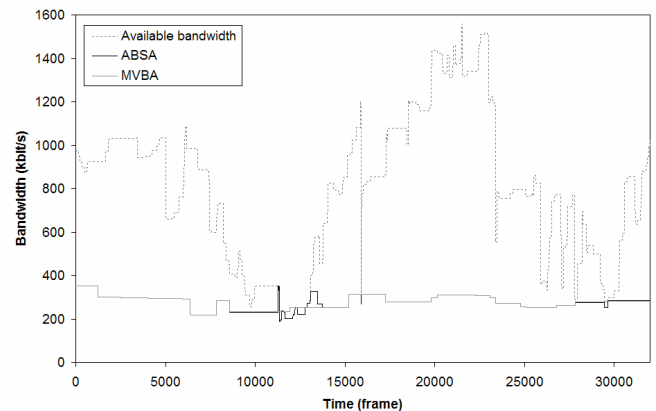


Fig. 7. A comparison between the ABSA and MVBA smoothing algorithms.

The ABSA behaviour can be synthesized as follows. ABSA decreases the smoothed stream bit rate when available bandwidth is very low, by decreasing the corresponding CBR segment slopes (see Fig. 3) in such a way that they are always lower than the corresponding available bandwidth slope. When the available bandwidth raises again, the CBR segment slopes can be increased again in such a way that the smoothed stream transmission plan remains between the lower and upper bounds, experimenting no frame losses (see again Fig. 3). In other words, the ABSA algorithm is able to compensate in some way the lower bit rates scheduled during the strong bandwidth reduction zone simply by increasing the transmission plan bit rate when available bandwidth is available again.

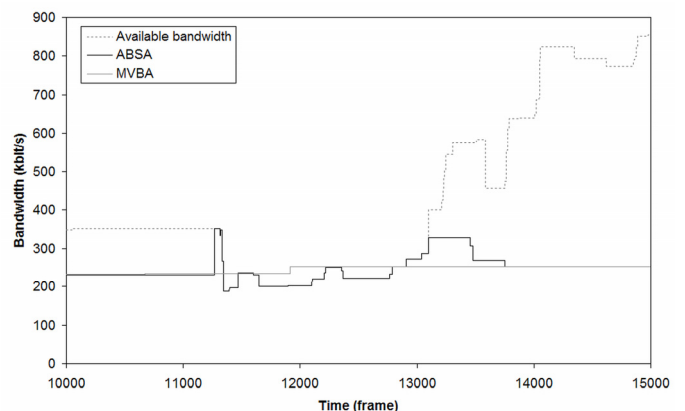


Fig. 8. ABSA and MVBA behaviour in the critical time zone in which available bandwidth is missing.

To test the E-ABSA performance, three video streams of different types, a soccer final, a movie and a MTV video clip, all lasting for  $N = 40,000$  video frames and codified with the MPEG-1 standard, have been aggregated randomly choosing their starting points. The E-ABSA algorithm has then been applied considering a total smoothing buffer of 3,072 kbytes corresponding to three client smoothing buffers each of 1,024 kbytes. To guarantee the total stream superimposition it has been supposed that each of the three video streams starts immediately again after its end. The available bandwidth information has been obtained exactly in the same manner as has previously been explained.

In Fig. 9 there is a comparison between the bit rate evolution of the unsmoothed stream aggregate and that of the E-ABSA smoothed aggregate. Let us remember that the E-ABSA smoothes directly the stream aggregation taking into account bandwidth limitation. Fig. 9 points out the great difference in peak rate and rate variability between the unsmoothed and smoothed aggregate.

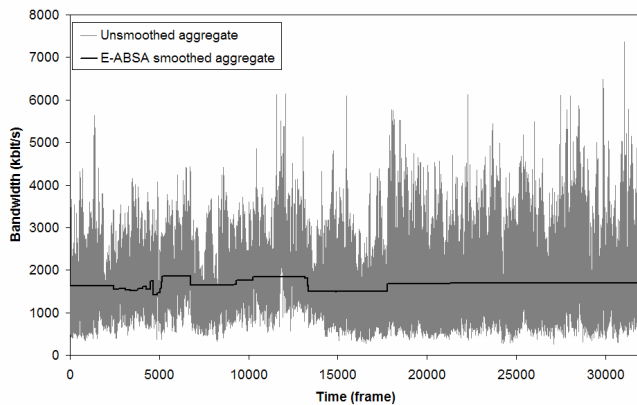


Fig. 9. A comparison between unsmoothed and E-ABSA smoothed stream aggregate.

In Fig. 10 there is an example of the E-ABSA smoothing algorithm application for a stream aggregation. In this figure E-ABSA results have been compared with MVBA applied on the same aggregate data in a time window of length 32,000 video frames.

E-ABSA follows very well the available bandwidth behaviour, again no frame losses have been experimented. This behaviour is more evident in the time interval ranging from the 2,453<sup>th</sup> to the 5,159<sup>th</sup> frame. Again, it is also evident from Fig. 10 that the MVBA curve crosses the available bandwidth curve in the same time interval, testifying frame losses. Like the single video stream, immediately after the low available bandwidth time interval, the bit rate calculated by E-ABSA becomes higher than the corresponding MVBA bit rate. This happens because the former has to compensate in some way the lower bit rates assigned to the stream aggregation during the time intervals of stronger available bandwidth reduction. Finally, in the last time interval the E-ABSA curve resolves in a single, very long CBR piece, while the MVBA smoothed stream aggregation assumes different bit rates. This behaviour testifies to the E-ABSA capacity to schedule stream aggregation in such a way so as to have long CBR pieces with minimum variability among them,

like MVBA algorithm. In this case we note that E-ABSA performs better than MVBA in the last time interval. This happens because of the difference among E-ABSA and MVBA schedules when available bandwidth is missing, that influences future scheduling. In fact, MVBA schedule presents minimum variability among CBR pieces all the time. E-ABSA schedule presents higher bit rate variability when available bandwidth is missing. Nevertheless, this behaviour makes possible a smoother transmission plan in the next frame times, also according to E-ABSA characteristic to preserve MVBA main features.

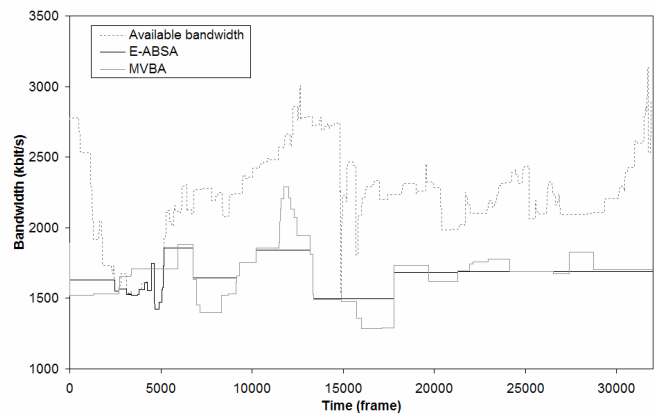


Fig. 10. A comparison between the E-ABSA and MVBA smoothing algorithms, for a stream aggregation smoothing.

In Fig. 11 the temporal evolution of each of the three chosen video streams is represented, composing the smoothed aggregate (each represented with a dotted line) in the chosen temporal observation window of 32,000 video frames (the total receiving smoothing buffer has been chosen of 3,072 kbytes corresponding to three client smoothing buffers each of 1,024 kbytes). The three schedules have also been compared with the E-ABSA smoothed aggregate, depicted as a continuous black line.

It is worth noting that in each frame time the sum of the three video streams gives exactly the E-ABSA smoothed stream aggregate. Furthermore, in some discrete time instants one of the video streams can assume a 0 bit rate value. The null bit rate value sometimes assumed by the single video stream is justified by considering that E-ABSA assigns bandwidth with a higher priority to video streams that transmit frames with a relatively high bit rate, to avoid frame losses. Contemporarily, streams transmitting with a lower bit rate can be sometimes scheduled with a null bit rate without losing continuous data flowing downstream the buffer. In fact, at the single stream receiving side continuous video playback is still possible if the receiving smoothing buffer is full enough to guarantee data flowing downstream the buffer even if there is no data coming upstream. This is confirmed by observing again Fig. 11: around the 10,000<sup>th</sup> video frame, the soccer final and the MTV clip assume higher bit rates if compared with the movie. Thus E-ABSA tends to assign higher bit rates to the first two mentioned streams at the same time guaranteeing the aggregate smoothing.

The ABSA algorithm has been tested also when available bandwidth assumes a fixed constant value. The comparison

between ABSA and MVBA algorithms has been performed on a piece of 40,000 video frames of the “Terminator 2” film. It has been smoothed with both MVBA and ABSA algorithms using a smoothing buffer of 512 kbytes. In Fig. 12 this comparison is represented for the first 32,000 video frames. The constant available bandwidth value has been fixed to 300 kbit/s. As expected, while the ABSA schedule does not always exceed available bandwidth, the MVBA schedule exceeds it during the first 1,240 video frames.

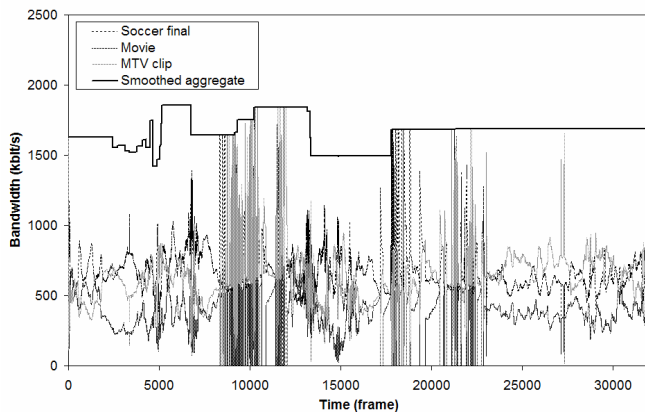


Fig. 11. Temporal evolution of the three video streams composing the aggregate and comparison with the smoothed aggregate.

As discussed in [4], MVBA scheduling minimizes also the smoothed stream peak rate. From [4] it also emerges that each MVBA rate increase is due to buffer underflow. In fact, when there is a buffer underflow, the MVBA algorithm increases smoothed bit rate in such a way to avoid it. This holds also for the smoothed peak rate. Nevertheless, if the constant available bandwidth is lower than the MVBA smoothed stream peak rate, some frame losses necessarily will occur both with MVBA and ABSA, since frame losses are due to buffer underflows. In fact, because of the stringent bandwidth limit, the cumulative amount of bytes arriving to the receiving buffer is not sufficient to extract video frames at the same time guaranteeing continuous video stream playback downstream the receiving buffer. As a consequence, the cumulative schedule curve as depicted in Fig. 3 will tend to be very close to the lower bound limit whenever low bandwidth occurs, and crosses it in case of frame losses. Nevertheless, the ABSA algorithm also tries to minimize frame losses by increasing the smoothed stream bit rate whenever possible, at the same time trying to follow the MVBA algorithm. The only way to increase the CBR segment bit rates is to increase the cumulative transmission plan slopes whenever possible (see Fig. 3). In the case of variable available bandwidth this task is performed when available bandwidth raises again, so that segment slopes can also be increased. Unfortunately, for constant available bandwidth ABSA can never increase segment slopes (bit rates) more than the constant available bandwidth. For this reason, ABSA can not compensate with higher bit rates the lower bit rates forced in the critical time zone in which available bandwidth is missing (e.g., the earliest time period in Fig. 12). This will surely cause loss of frames. Nevertheless, they will be presumably less than the frame losses due to MVBA algorithm application, since ABSA is

able to redistribute bit rates in such a way so as to maintain the cumulative transmission plan between the two bounds. This aspect can be noted by observing Fig. 12 again. Here it can be seen that ABSA increases the first CBR segment duration until the 6,803<sup>th</sup> video frame. Also the second and the third CBR segment bit rates are higher if compared with the single CBR segment obtained through MVBA that extends until the 11,917<sup>th</sup> frame.

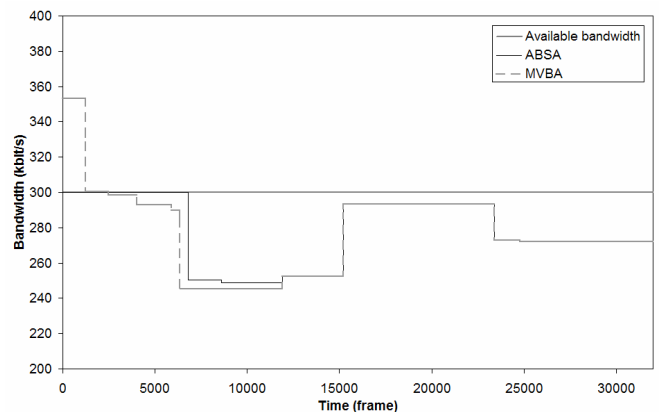


Fig. 12. Comparison between ABSA and MVBA for a single video stream and constant available bandwidth.

By running ABSA a frame loss evaluation is also possible. In particular, with ABSA only 265 frames have experimented to be lost. With MVBA, all the first 1,240 video frames are lost, as it is clear from Fig. 12.

The extended version of ABSA for video stream aggregation smoothing, E-ABSA, behaves in the same way as for constant available bandwidth. In Fig. 13 a comparison between the E-ABSA and MVBA algorithms is displayed. Three films of different types and all of length 40,000 video frames have been aggregated: a soccer final, a movie and a MTV video clip, randomly choosing their starting points. The bit rate smoothing has been performed directly on the stream aggregation in a time window lasting for the three films duration (40,000 video frames), even if only the first 32,000 video frames are displayed in Fig. 13. As the case of variable available bandwidth, it has also been supposed that when a video stream finishes, another of the same type begins immediately. This guarantees the total superimposition of the three considered streams. The constant available bandwidth value has been fixed to 1.75 Mbit/s.

From Fig. 13 the different behaviour of E-ABSA and MVBA algorithms can be noted. The E-ABSA perfectly follows available bandwidth constraint for the first 5,428 video frames, on the left side of the figure, at the same time trying to smooth the aggregate schedule while respecting bandwidth limitation when lower bit rates occur. This behaviour is evident after the 5,428<sup>th</sup> video frame: E-ABSA tries to keep the MVBA characteristics of minimum variability among CBR pieces and maximum CBR segments length. From a comparison with MVBA schedule, it is also evident that E-ABSA minimizes frame losses by starting with a long CBR segment at the maximum permitted value. The initial part of MVBA schedule instead presents more CBR segments, one of them exceeding the constant capacity and thus



provoking several continuous frame losses. As it can be noted from Fig. 13, the MVBA schedule causes frame losses in three different points: from frame 3,527 to frame 5,404, from frame 8,231 to frame 11,525, and from frame 24,407 to frame 26,104. Through E-ABSA application, a similar situation arises only for 14 video frames in non contiguous positions, bringing to an enormous advantage in terms of transmission efficiency in critical available bandwidth conditions.

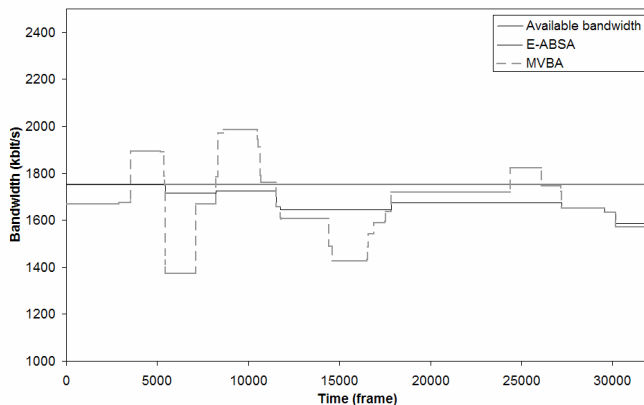


Fig. 13. E-ABSA and MVBA algorithms for stream aggregation and constant available bandwidth.

In Fig. 14 the bit rate evolution of the three video streams composing the aggregate have been visualized; they have also been compared with the smoothed aggregate. Like the case discussed in Fig. 11, even if the single stream bit rate is quite highly variable, the stream aggregate is smoothed while respecting available bandwidth bounds.

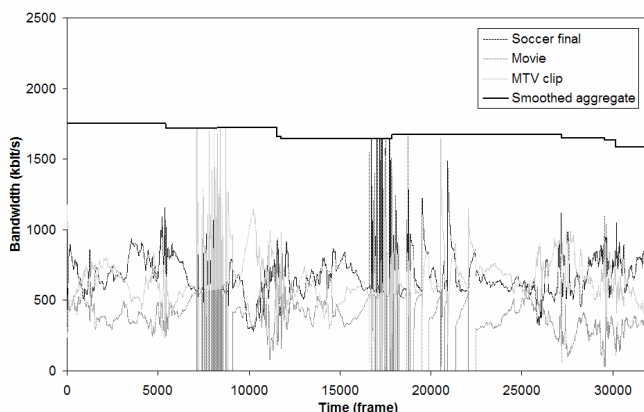


Fig. 14. Bit rate evolution of the three video streams and comparison with the smoothed aggregate.

## V. CONCLUSIONS AND FUTURE WORK

In this paper, a smoothing algorithm, called ABSA algorithm, together with its enhanced variant that smoothes stream aggregates (E-ABSA algorithm), has been developed and analyzed. It is able to take into account available bandwidth bounds, adapting the smoothing transmission plan to available bandwidth resources. Furthermore, ABSA is able to minimize frame losses in all experimented cases of variable and constant available bandwidth, if compared with the

original MVBA algorithm not taking into account bandwidth bounds. This makes ABSA suitable for a more efficient packet transmission planning. This comes at a cost of a grown computational complexity of ABSA if compared with MVBA. Other aspects of ABSA are currently under study, like an integration with Admission Control schemes that refuse video admission in case of excessive frame losses.

## REFERENCES

- [1] J. Kurose, K. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet", Addison Wesley (third edition), 2005.
- [2] Z.-L. Zhang, J. Kurose, J. D. Salehi, D. Towsley, "Smoothing, Statistical Multiplexing and Call Admission Control for Stored Video", IEEE Journal on Selected Areas in Communications, vol.15, no.6, pp. 1148-1166, August 1997.
- [3] M. Grossglauser, J.-C. Bolot, "On the Relevance of Long-Range Dependence in Network Traffic", IEEE/ACM Transaction on Networking, vol.7, n.5, pp.629-640, October 1999.
- [4] J.D. Salehi, Z.-L. Zhang, J. Kurose, D. Towsley, "Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements Through Optimal Smoothing," IEEE/ACM Transactions On Networking, vol.6, n.4, pp. 397-410, August 1998.
- [5] W.-C. Feng, J. Rexford, "Performance Evaluation of Smoothing Algorithms for Transmitting Pre-recorded Variable-Bit-Rate Video", IEEE Transactions on Multimedia, Vol. 1, No.3, September 1999, pp. 302-313.
- [6] J.Zhang, J.Hui, "Applying Traffic Smoothing Techniques for Quality of Service Control in VBR Video Transmissions", Computer Communications, vol.21, pp.375-389, April 1998.
- [7] P. Thiran, J.-Y. Le Boudec, F. Worm, "Network Calculus Applied to Optimal Multimedia Smoothing", IEEE INFOCOM 2001, pp.1474-1483.
- [8] S.Anastasiadis, P.Varman, J.S. Vitter, K.Yi, "Lexicographically Optimal Smoothing for Broadband Traffic Multiplexing", 21<sup>st</sup> Annual ACM Symposium on Principles of Distributed Computing, pp.68-77, Monterey, CA, July 2002.
- [9] U. Reimers, "Digital Video Broadcasting", IEEE Communication Magazine, vol.36, pp.104-110, June 1998.
- [10] DVB project homepage (<http://www.dvb.org>).
- [11] ETSI, "Digital video broadcasting (DVB); framing structure, channel coding and modulation for digital terrestrial television", EN 300 744, V1.5.1, 2004-2006.
- [12] W. Feng, S. Sechrest, "Critical Bandwidth Allocation for Delivery of Compressed Video", Computer Communication, vol.18, pp.709-717, October 1995.
- [13] W. Feng, F. Jahanian, S. Sechrest, "An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Pre-recorded Video", ACM/Springer-Verlag Multimedia Systems Journal, September 1997.
- [14] J.M. McManus, K.W. Ross, "Video on Demand over ATM: Constant-rate Transmission and Transport", IEEE Journal on Selected Areas in Communications, vol.4, n.6, pp.1087-1098, August 1996.
- [15] O. Hadar, R. Cohen, "PCRTT Enhancement for Off-Line Video Smoothing", Journal of Real-Time Imaging, vol.7, n.3, pp. 301-314, June 2001.
- [16] J.M. McManus, K.W. Ross, "A Dynamic Programming Methodology for Managing Pre-recorded VBR Sources in Packet-Switched Networks", Telecommunication Systems,

vol.9, 1998.

- [17] W. Feng, M. Liu, "Critical Bandwidth Allocation Techniques for Stored Video Delivery Across Best Effort Networks", Proceedings of 20th International Conference on Distributed Computing Systems, pp.56-63, April 2000.
- [18] C. Bewi, R. Pereira, M. Merabti, "Network Constrained Smoothing: Enhanced Multiplexing of MPEG-4 Video", Proc. of 7<sup>th</sup> International Symposium on Computers and Communications (ISCC'02), pp. 114-119, July 2002.
- [19] J. K.-Y. Ng, S. Song, "A Video Smoothing Algorithm for Transmitting MPEG Video over Limited Bandwidth", Proc. 4<sup>th</sup> International Workshop on Real-Time Computing Systems and Applications (RTCSA'97), Taipei, pp. 229-236, October 1997.
- [20] J.-Y. Le Boudec, P. Thiran, "Network Calculus: A Theory of Deterministic Queueing Systems for the Internet", Book Springer Verlag, version May 10, 2004.
- [21] W.-C. Feng, "Rate-constrained bandwidth smoothing for the delivery of stored video", Proc. SPIE Multimedia Computing and Networking, San Jose, CA, February 1997.
- [22] P. A. Chou, Z. Miao, "Rate-distortion optimized streaming of packetized media," accepted for publication on IEEE Transactions on Multimedia.
- [23] S. G. Rao, S. V. Raghaven, "Fast Techniques for the Optimal Smoothing of Stored Video", ACM/SpringerVerlag Multimedia Systems Journal, vol.7, n.3, pp. 222-233, May 1999.
- [24] J. Rexford, S. Sen, J. Dey, J. Kurose, D. Towsley, "Online Smoothing of Variable-Bit-Rate Streaming Video", IEEE Transactions on Multimedia, vol.2, n.1, pp.37-48, March 2000.
- [25] G. Cao, W.-C. Feng and W. Singhal, "Online VBR video traffic smoothing", in Proceedings of the Eighth International Conference on Computer Communications and Networks (IC3N'99), 1999.



**Pietro Camarda** received a University Degree in Computer Science from the University of Bari, Italy, in 1978. Since 1986, he has been with the department of "Elettrotecnica ed Elettronica" at the "Politecnico di Bari", Italy, where he is currently a Full Professor of Telecommunications. He was visiting scholar from October 1986 to December 1987 at Computer Science Department of the University of California, Los Angeles, with the supervision of Prof. Mario Gerla.

He has been involved in various research areas, mainly: LAN/MAN Architectures and Protocols, Optical Networks, Network Reliability, Cellular Radio Networks, Telecommunication Services, etc. He is the author of more than 80 research papers on these topics. In the past few years he has focused mainly on wireless networks (cellular networks, wireless LANs, Ad Hoc Networks), developing topics that include Resource Allocation based on Neural Networks, Header Compression Schemes for Wireless Networks, Authentication Protocols in Wireless Networks, Bandwidth Allocation and Quality of Service in WLANs, MAC protocols for Ad Hoc networks, etc.



**Cataldo Guaragnella** graduated in Electronic Engineering in 1990 at Università degli Studi di Bari with honors, and received the PhD in Telecommunications in 1994 by the Politecnico di Bari.

In 1996 he joined the Electrics and Electronics Department of Politecnico di Bari as an Assistant Professor in Telecommunications.

His main research interest are in the fields of multidimensional signal processing, pattern recognition video and image processing/coding.



**Domenico Striccoli** received, with honors, the Dr. Eng. Degree in Electronic Engineering in April 2000, and the Ph.D. degree in April 2004, both from the "Politecnico di Bari", Italy. In 2005 he joined the DIASS Department of Politecnico di Bari as Assistant Professor in Telecommunications. He is actually involved in the study of Video Distribution Systems and Quality of Service issues. His main research

interests span the fields of Video on Demand, MPEG coding techniques, Smoothing algorithms, Quality of Service, Bandwidth Estimation, etc.