# Optimization of Multiservice Scheduling for Variable Bit Rate Video Transmission in DVB-H Systems

Domenico Striccoli
Politecnico di Bari/DIASS, Viale del Turismo 8, 74100, Taranto, Italy
Email: d.striccoli@poliba.it

Pietro Camarda and Francesco Capozzi
Politecnico di Bari/DEE, Via E. Orabona 4, 70125, Bari, Italy
Email: camarda@poliba.it, fracap84@gmail.com

***Abstract*—Digital Video Broadcasting for Handheld terminals (DVB-H) is assuming an ever growing importance for digital video transmission over wireless terminals. In such a context, Time Slicing has been implemented to achieve a better power saving and manage handover. Specifically, a generic user transmits bursts of data, interspaced by time periods in which no data are transmitted.**

**In this paper, to improve time sliced multiservice transmission effectiveness, the Variable Burst Time (VBT) algorithm is presented and discussed. It dynamically varies the whole set of stream Burst Durations according to input stream data, available channel bandwidth, receiving buffer size and eventually stream priority. Burst Durations are derived by the minimization of a Total Loss Function (TLF) representing the amount of losses of the whole service set.**

**Numerical results show the VBT effectiveness if compared with the time sliced transmission recommended in the DVB guidelines, for different numbers, types and quality degrees of VBR streams, receiving buffer sizes and stream priorities. This suggests that VBT could be efficiently exploited for transmission of VBR streams in DVB-H systems.**

***Keywords*—DVB-H, Time Slicing, Multiservice, Available Bandwidth.**

## I. INTRODUCTION

Digital Video Broadcasting (DVB) is actually one of the most significant technological challenges of our time. Its evolution for hand-held terminals allows multimedia content to be received on several wireless terminals like smartphones, Personal Digital Assistants (PDAs), notebooks, etc. Data are transmitted through the IP protocol, exploiting the same DVB Terrestrial (DVB-T) network infrastructure. Nevertheless some specific problems typical of mobile stream delivery require additional specifications for DVB-H at physical and data-link layers of the ISO-OSI protocol stack. Specifically, there are deep similarities between DVB-H and cellular networks transmission [1][2][3]. Error correction techniques are indispensable to compensate the multipath and Doppler effects typical of wireless terminals reception. Furthermore, due to the limited battery capacity, handheld terminals power consumption must be minimized.

To improve terminal performance and energy saving Multi Protocol Encapsulated data-Forward Error Correction (MPE-FEC) and Time Slicing have been introduced, both at data link layer. They are implemented at the Transport Stream layer of the MPEG-2 video flow, independently from the DVB physical layer, that is almost the same of DVB-T [4].

Time slicing is introduced to improve both terminal power saving and handover. MPE-FEC instead improves system robustness and tolerance towards noise [2]. At network layer, IP protocol is adopted. The DVB-H payload consists of IP datagrams encapsulated into MPE sections and multiplexed together with classical MPEG-2 services [5]. An example of the DVB-H use for IP service transmission is illustrated in Figure 1, where both classical DVB-T and new DVB-H services are multiplexed. Mobile terminals decode only DVB-H services.

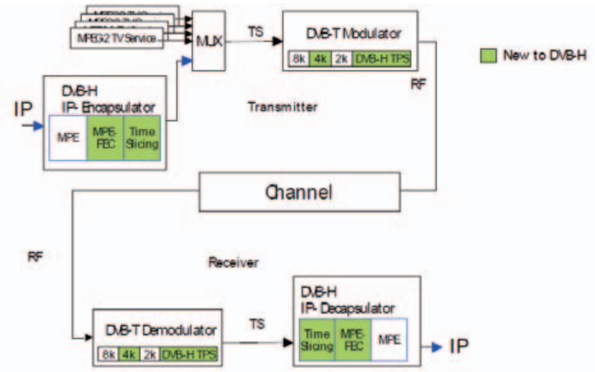


Figure 1. Structure of a DVB-H system.

Manuscript received March 6, 2009; revised June 3, 2009; accepted August 19, 2009.


doi:10.4304/jcm.4.9.681-690

Time slicing is very important for mobile terminals stream delivery. Service data are transmitted in "packets" or "bursts" periodically repeating in time. For each service, bursts are interspaced by "off-time" periods, in which no data are transmitted. To guarantee a lossless decoding at receiving side, the bit rate of data transmitted in a burst is consistently higher than the average bit rate required for the continuous transmission in DVB-T systems. Figure 2 illustrates the comparison between DVB-T and DVB-H transmission of four services.

Detailed time slicing parameters for the single service are represented in Figure 3. The Burst Size (BS) represents the total Network Layer bits in the burst, included the overhead due to MPE-FEC and Cyclic Redundancy Check code (CRC-32) header. There is also an adding overhead due to the transport packets header depending on the MPE section length. Generally, a total 4% overhead is assumed, according to [3], and in this work we will keep this assumption. The Burst Bitrate (BB) is the bit rate of the time-sliced stream. The Constant Bitrate (CB) is instead the average bit rate required by the stream not time sliced. The Burst Duration (BD) is the time interval from the beginning to the end of the burst. Together with this parameter, the Maximum Burst Duration (MBD) shall be defined and signaled for a time sliced elementary stream. The duration of each burst can never exceed the MBD. The MBD information can be exploited by the receiver to identify the end of a burst, in conditions of weak signal reception. The Off-time (Ot) is the time interval between two consecutive bursts of the same service. During the Off-time bursts relative to other services can be transmitted in a classical TDM transmission.

In the sequel, we will define the Burst Cycle (BC) as the time interval between the beginning of two burst of the same service. Obviously it holds that $BC = BD + Ot$.

Time slicing allows a consistent power saving at receiving side since the receiver remains inactive during off-time and the power consumption is reduced. Furthermore, off-time can be exploited to perform handover without any service interruptions.

To indicate the beginning of the next burst, the "delta-t" method is adopted.

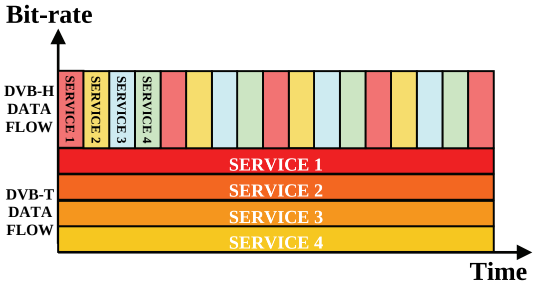


Figure 2. Comparison between DVB-T and DVB-H transmission.

It is a "relative" information contained into each MPE section and indicates the time interval between two consecutive bursts. The delta-t method is useful to know the beginning of the next burst also in conditions of weak signal reception.

Data contained in a burst are buffered in the client memory during burst times, and are consumed also during off-times. So, sufficient buffering is needed at receiving side for continuous and lossless decoding. Guidelines recommend that the Burst Size must always be less than the memory available in the receiver and the Burst Bandwidth and Burst Duration can be set so that the buffer can store enough data to guarantee continuous playback also during off-time [3]. Let us note that it is quite easy to statically set the burst parameters for Constant Bit Rate (CBR) streams; it is instead more difficult for Variable Bit Rate (VBR) videos, coded with MPEG-4 or H-264 standards. In these cases, the video bit rate is highly variable in time. Considering the same burst size during video transmission could easily bring to losses at receiving side because of the insufficient amount of data to be played during off-times and/or a relatively small buffer size.

Another aspect is that service data are stored in bursts regardless any bandwidth information. They could be more effectively scheduled at transmission side if available bandwidth assigned for the single service is a known, for example, when different bandwidth levels are assigned to services with different priorities. In this work we consider the available bandwidth parameter as the result of specific resource allocation policies adopted by the service provider, derived by the application of QoS algorithms at transmission side, that are not considered in this study. We suppose that this information is known a priori and available to calculate the burst bitrate.

It is clear in fact that a reduced available bandwidth limits the burst bitrate and consequently reduces the burst size, under the same burst duration. This aspect is critical for a lossless transmission. In fact, when the available bandwidth is relatively small, burst size could not be enough to guarantee the continuous playback at receiving side also during off-time. This problem is emphasized for VBR streams, that present a high bit rate variability in time. In this case available bandwidth during the transmission of a burst could also be not entirely exploited, with consequent bandwidth waste.

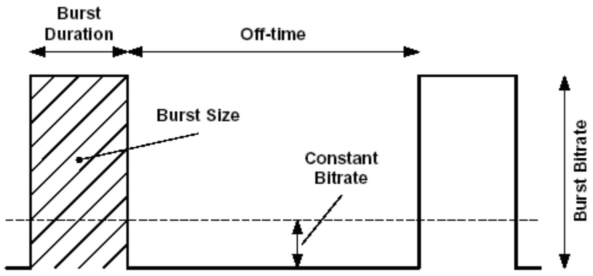


Figure 3. Time Slicing parameters for the single service.

In this work we propose and analyze a multiservice scheduling algorithm, the Variable Burst Time (VBT) algorithm, suitable for transmission of time-sliced high quality VBR services, that takes also into account available bandwidth. Transmission optimization is performed by dynamically and simultaneously adjusting the burst durations of the whole set of services transmitted in a Temporal Observation Window (TOW) of fixed size, sliding in time. The additional information on varying burst durations and off-times could be properly signaled in each burst, using the MPE headers and the Delta-t method, without any significant change in DVB-H implementation. Optimization is performed by taking into account available bandwidth, burst and receiving buffer sizes, and the different service priorities. The goal is the on-the-fly loss minimization of the whole set of multiplexed services. The scheduling algorithm proposed in this work is more flexible than the classical DVB-H transmission that considers a fixed service Burst Duration, since it adjusts service Burst Durations during stream running for a more efficient resource allocation.

VBT schedule, or transmission plan, is generated at server side with the aim to prevent buffer overflows and underflows, the only two conditions supposed to generate losses. In this scenario, a buffer underflow occurs if the burst size is relatively small and cannot “cover” the whole Burst Cycle. A small burst size is due to a small burst duration and/or a low available bandwidth. A buffer overflow occurs instead if the receiving buffer is not large enough to fully store the incoming burst.

The rest of the paper is organized as follows. Section II describes VBT. In Section III VBT performance is compared with the classical DVB-H scheduling that statically assigns burst durations and does not take into account available bandwidth. In Section IV some conclusions on the effectiveness of the proposed method will be provided.

## II. THE VARIABLE BURST TIME (VBT) ALGORITHM

VBT algorithm exploits some basic scheduling principles developed for smoothed transmission of VBR streams in video distribution systems. Several “off-line” smoothing algorithms have been developed and widely analyzed [6]-[9]. They regularize the transmission of VBR streams by a schedule that drastically reduces their burstiness. At receiving side, the schedule enters a buffer and the original VBR stream leaves it for decoding and playing. On-line algorithms have also been considered in literature; they generate a transmission plan by knowing only a stream portion in a limited time window [10][11]. Other algorithms consider also the influence of available bandwidth in generating the transmission plan [12]-[14]. The key feature for all algorithms is to generate a transmission plan for the single video stream that should always avoid buffer overflow and/or underflow conditions. VBT performs much more. It takes into account simultaneously the transmission plan of all time sliced services, by dynamically varying their burst durations in a TOW. The main complication is that the transmission plans of all services are strongly correlated, since a service burst duration variation influences the other service off-times and vice versa (see Figure 2). VBT calculates the optimal transmission plan for all services that minimizes bit losses, by taking into account the video data, receiving buffer size, available bandwidth and service priorities. The basic idea is to schedule enough data in each burst to prevent buffer overflows and underflows. This is performed by dynamically varying all service burst durations. To fully understand the VBT operation, let us explain in detail the single service transmission plan, before reporting the multiservice scenario.

### A. The Single Service Scenario

Let us now concentrate on the single service schedule. We suppose the frame time (1/25 s for PAL) as the basic time unit for our purposes, so that any time interval can be treated as a positive integer. VBT schedules as many data as possible in each burst in advance respect to their playback time, avoiding both buffer overflows and underflows. To this aim, given the receiving buffer size of $b$ bits and $f_i$ the $i^{th}$ frame size (in bits), two curves are built:

$$F_{under}(k) = \sum_{i=0}^{k} f_i \tag{1}$$

$$F_{over}(k) = b + \sum_{i=0}^{k} f_i \tag{2}$$

They represent, respectively, the cumulative amount of data leaving the client buffer for playback at the $k^{th}$ frame time and the maximum cumulative amount of data to be received by client at the $k^{th}$ frame time without overflowing the client buffer. Similarly, the cumulative transmission plan at the $k^{th}$ frame time will be given by:

$$F_{under}(k) \le S(k) = \sum_{i=0}^{k} s_i \le F_{over}(k) \tag{3}$$

where $s_i$ represents the scheduled bit rate at $i^{th}$ frame time [6]. By definition, (1), (2) and (3) are non decreasing curves. They are represented in Figure 4 in a generic burst cycle, where it is supposed that the service burst duration starts in $T_{bi}$ and ends in $T_{bs}$, and that the service off-time starts in $T_{bs}$ and ends in $T_{cycle}$. The burst duration is given by $T_{on} = T_{bs} - T_{bi}$ and the off-time by $T_{off} = T_{cycle} - T_{bs}$.

In this case $s_i$ coincides with the burst bitrate ($s_i = BB$) for $T_{bi} \le i \le T_{bs}$, and $s_i = 0$ for $T_{bs} \le i \le T_{cycle}$.

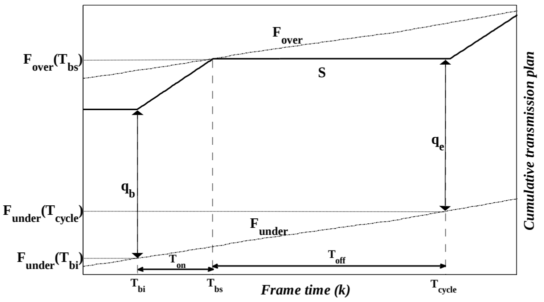


Figure 4. Cumulative transmission plan S(k) in a burst cycle.

The cumulative schedule $S(k)$ can increase only in $[T_{bi}, T_{bs}]$ and remains constant in $[T_{bs}, T_{cycle}]$, and the slope of $S(k)$ in $[T_{bi}, T_{bs}]$ coincides with the burst bitrate. $q_b$ represents the buffer fill level at the beginning of the burst cycle, that is the amount of data stored in the buffer and not yet decoded by client at the end of the previous burst cycle. Similarly, $q_e$ is the receiving buffer fill level at the end of the burst cycle.

It holds:

$$q_b = S(T_{bi}) - F_{under}(T_{bi}) \tag{4}$$

$$q_e = S(T_{cycle}) - F_{under}(T_{cycle}) \tag{5}$$

The schedule will be feasible without losses if and only if:

$$F_{under}(k) \le S(k) \le F_{over}(k) \ \forall \mathrm{k} \tag{6}$$

Figure 5 depicts the two events supposed to generate losses at receiving side.

If there is a $k$ exists where $S(k) > F_{over}(k)$ (see the $S_{ov}$ curve in Figure 5), cumulative data sent in $k$ exceed the maximum amount of data that can be stored in the client buffer and a buffer overflow occurs in $k$. Vice versa, if a $k$ exists where $S(k) < F_{under}(k)$ (see the $S_{und}$ curve in Figure 5), data received by client in $k$ are less than data consumed by the client and a buffer underflow occurs in $k$.

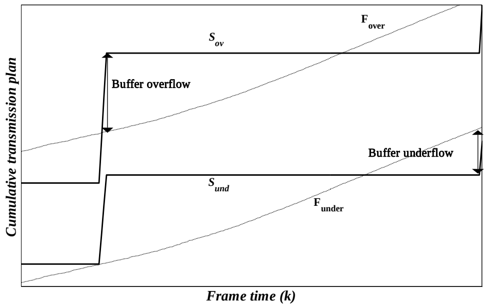


Figure 5. Buffer overflow and underflow events in a burst cycle.

Let us point out that a buffer overflow can easily be avoided by properly regulating the burst bitrate at transmission side, where the receiving buffer size is supposed to be known, so that $S(k)$ cannot cross $F_{over}(k)$ in $[T_{bi}, T_{bs}]$. A buffer underflow occurs because there are not enough data in the burst to guarantee lossless decoding also during off-time, when receiver is off. This could happen both because available bandwidth limits the burst bitrate (the slope of $S(k)$ in $[T_{bi}, T_{bs}]$ is reduced), or because the off-time is relatively long if compared with the burst duration (even without any bandwidth limitation). Both these events cannot be controlled at transmission side because they do not depend on the single service parameters. As will be clearer in the next section, VBT tries to reduce all service losses for buffer underflow. To this aim, all service burst durations are simultaneously varied in a sliding time window taking into account available bandwidth, until the minimum for losses is reached.

### *B. The Multiservice Scenario*

As highlighted in Section II, burst durations cannot be calculated independently from each other. VBT calculates simultaneously them all in a Temporal Observation Window (TOW) whose length is chosen as a integer number of burst cycles. Let us suppose $N_S$ services multiplexed in a burst cycle, and that in a TOW there are $W_S$ burst cycles as highlighted in Figure 6.

We define as *configuration* the n-uple of burst durations :

$$\bar{T}_{on} = \left(T_{on}^{(1,1)}, .., T_{on}^{(N_S,1)} .., T_{on}^{(1,W_S)}, ..., T_{on}^{(N_S,W_S)}\right) \tag{7}$$

where $n = N_s \cdot W_s$, and each $T_{on}^{(i,j)}$ is the burst duration of the $i^{th}$ service in the $j^{th}$ burst cycle. $T_{on}^{(i,j)}$ is a positive integer multiple of the frame time unit. VBT finds the optimal configuration $\bar{T}_{on,opt} = \left(T_{on,opt}^{(1,1)}, ..., T_{on,opt}^{(N_S,W_S)}\right)$ that minimizes all service losses. A *Total Loss Function* (TLF) is introduced, that considers all the service losses for buffer underflow in the TOW. In this study we will not consider other kind of losses due to transmission and/or decoding errors. The TLF must be properly set, like the TOW length and the optimization method. These three key aspects will be developed in detail in the following subsections.

#### *1) The Total Loss Function*

To define the TLF the first step is to calculate losses for the generic $i^{th}$ service by considering its buffer state $q_e^{(i,j)}$ at the end of the $j^{th}$ burst cycle. By looking at Figure 4 we know that cumulative data filling the buffer are:

$$D_{in}^{(i,j)}(T_{on}^{(i,j)}, W_{av}^{(i,j)}) = BB^{(i,j)} \cdot T_{on}^{(i,j)} \tag{8}$$

where $BB^{(i,j)}$ is the $i^{th}$ service burst bitrate in $T_{on}^{(i,j)}$.

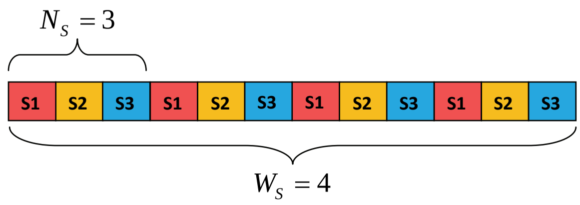


Figure 6. Graphical representation of a Temporal Observation Window.

$BB^{(i,j)}$ must be properly calculated to avoid buffer overflow and to be less than the available bandwidth $W_{av}^{(i,j)}$ assigned to the $i^{th}$ service , supposed constant in $T_{on}^{(i,j)}$. That is:

$$BB^{(i,j)} = \min\left\{W_{av}^{(i,j)}, \frac{F_{over}(T_{bs}^{(i,j)}) - \left(F_{under}(T_{bi}^{(i,j)}) + q_b^{(i,j)}\right)}{T_{on}^{(i,j)}}\right\} \quad (9)$$

where $q_b^{(i,j)}$ is the buffer fill level in $T_{bi}^{(i,j)}$. Let us note that $BB^{(i,j)}$, and consequently $D_{in}^{(i,j)}$, depend both on $T_{on}^{(i,j)}$ and $W_{av}^{(i,j)}$. Cumulative data leaving the buffer at the end of the Burst Cycle are:

$$D_{out}^{(i,j)}(T_{on}^{(i,j)}, T_{off}^{(i,j)}) = F_{under}(T_{cycle}^{(i,j)}) - F_{under}(T_{bi}^{(i,j)}) \quad (10)$$

as clearly visible in Figure 4.

The buffer fill level in $T_{cycle}^{(i,j)}$ is thus:

$$\begin{aligned} &q_e^{(i,j)}(T_{on}^{(i,j)}, T_{off}^{(i,j)}, W_{av}^{(i,j)}) = \\ &= q_b^{(i,j)}(T_{bi}^{(i,j)}) + D_{in}^{(i,j)}(T_{on}^{(i,j)}, W_{av}^{(i,j)}) - F_{under}(T_{cycle}^{(i,j)}) \end{aligned} \quad (11)$$

Losses will occur if and only if $q_e^{(i,j)}(T_{on}^{(i,j)}, T_{off}^{(i,j)}, W_{av}^{(i,j)}) < 0$, that is, $S$ crosses $F_{under}$ in $T_{cycle}^{(i,j)}$. That is:

$$\begin{aligned} L^{(i,j)}(T_{on}^{(i,j)}, T_{off}^{(i,j)}, W_{av}^{(i,j)}) = \max\left\{-q_e^{(i,j)}(T_{on}^{(i,j)}, T_{off}^{(i,j)}, W_{av}^{(i,j)}), 0\right\} \\ 1 \le i \le N_S,\ 1 \le j \le W_S - 1 \end{aligned} \quad (12)$$

Let us point out that losses computation for the $N_S$ streams can be performed until the $(W_S - 1)^{th}$ burst cycle. In fact, the $N_S$ burst durations of the $W_S{}^{th}$ burst cycle are only needed to evaluate the $N_S$ streams off-times in the $(W_S - 1)^{th}$ burst cycle, without any data allocation.

$L^{(i,j)}(T_{on}^{(i,j)}, T_{off}^{(i,j)}, W_{av}^{(i,j)})$ can be derived only after the available bandwidth $W_{av}^{(i,j)}$ and the $\overline{T}_{on}$ vector (7) have been set. In fact the off-time of the $i^{th}$ service in the $j^{th}$ burst cycle $\left(T_{off}^{(i,j)}\right)$ is given by:

$$\begin{cases} T_{off}^{(i,j)} = \sum_{k=2}^{N_S} T_{on}^{(k,j)} \text{ if } i = 1 \\ T_{off}^{(i,j)} = \sum_{k=i+1}^{N_S} T_{on}^{(k,j)} + \sum_{k=1}^{i-1} T_{on}^{(k,j+1)} \text{ if } i > 1 \end{cases}, 1 \le j \le W_S - 1 \quad (13)$$

and depends on the other services burst durations $T_{on}^{(k,j)}$.

A TLF that simply minimizes the sum of the service losses in a TOW disregards the fairness principle. There could be in fact multiplexed video streams with different quality degrees. Simply minimizing the same amount of lost bits for all services would penalize services with lower mean bit rates. The TLF guarantees *fairness* among services with respect to losses. Fairness is quantified in this work by taking into account two factors: the service quality degree and the service priority. Regarding the first aspect, TLF should minimize losses for each service relatively to its transmitted data, so that lower quality streams should have a smaller weight in the TLF computation. To take into account this aspect, losses for each service are normalized to data transmitted in the TOW:

$$L^{(i)}(\overline{T}_{on}, \overline{W}_{av}) = \left(\sum_{j=1}^{W_S-1} L^{(i,j)} \Big/ \sum_{j=1}^{W_S-1} D_{in}^{(i,j)}\right),\ 1 \le i \le N_S, \quad (14)$$

with $L^{(i,j)}$ given by (12), $D_{in}^{(i,j)}$ by (8), $\overline{T}_{on}$ by (7) and $\overline{W}_{av}$ is:

$$\overline{W}_{av} = \left(W_{av}^{(1,1)}, .., W_{av}^{(N_S,1)} .., W_{av}^{(1,W_S)}, ..., W_{av}^{(N_S,W_S)}\right) \quad (15)$$

that is the available bandwidth vector of the $N_S$ services in a TOW, supposed to be derived by the application of specific resource allocation policies as previously explained.

The second aspect is introduced to take into account the possibility of different priorities among streams. Losses for each service are further weighted by a priority factor $p_i$ that quantifies the importance of the $i^{th}$ service so that losses for higher priority services assume a higher weight in the TLF computation. The resulting loss factor is then:

$$L_{fac}^{(i)}(\overline{T}_{on}, \overline{W}_{av}) = L^{(i)}(\overline{T}_{on}, \overline{W}_{av}) \cdot \mathrm{p}_i,\ 1 \le i \le N_S \quad (16)$$

Losses calculated in (16) are then averaged over the services:

$$M(\overline{T}_{on}, \overline{W}_{av}) = \sum_{i=1}^{N_S} L_{fac}^{(i)} \Big/ N_S \quad (17)$$

Formula (17) indicates the total amount of losses averaged over the services in the TOW. Since (17) considers the total amount of losses in a TOW, there could be different $\overline{T}_{on}$ configurations bringing to the same value of $M(\overline{T}_{on}, \overline{W}_{av})$. The chosen configuration should be the fairest one, that is, it should guarantee, as much as possible, the equality of all the loss factors calculated in (16). A fairness index is thus introduced in order to measure the fairness of resource distributions in shared systems. This aspect is illustrated in [15] by the so called “Jain index”. It is widely adopted in computer networks and network engineering to determine whether users/applications are receiving a fair share of system resources. The Jain’s fairness metric has the desirable feature that it is minimized when one flow receives all the system capacity, and maximized when all flows receive the same capacity. In this work the “system resources” are represented by the loss factors, whose distribution among services in the chosen $\overline{T}_{on}$ configuration should be the fairest possible.

We define the Jain index as follows:

$$J(\overline{T}_{on},\overline{W}_{av}) = \left(\sum_{i=1}^{N_S} L_{fac}^{(i)}\right)^2 \Big/ N_S \sum_{i=1}^{N_S}\left(L_{fac}^{(i)}\right)^2 ; \qquad (18)$$

it is always comprised between $1/N_S$ and 1. The smaller value $(1/N_S)$ is obtained in correspondence of the worst case (the most unfair distribution of losses among services). The TLF is then defined as:

$$TLF(\overline{T}_{on},\overline{W}_{av}) = \frac{M(\overline{T}_{on},\overline{W}_{av})}{J(\overline{T}_{on},\overline{W}_{av})} ; \qquad (19)$$

so that the most unfair configurations produce higher TLF values.

VBT finds the optimal configuration $\overline{T}_{on,opt} = \left(T_{on,opt}^{(1,1)},\ldots,T_{on,opt}^{(N_S,W_S)}\right)$ that verifies the condition:

$$TLF(\overline{T}_{on,opt},\overline{W}_{av}) = \min\left\{TLF(\overline{T}_{on,},\overline{W}_{av}),\overline{T}_{on} \in \mathbb{N}_+^n\right\} \quad (20)$$

Where $\mathbb{N}_+^n$ is the subset of $\mathbb{N}^n$ including all the n-uples of strictly positive natural numbers.

*2) The Temporal Observation Window*

VBT is an algorithm that can be applied in both online or offline contexts, where respectively the video server has limited or full knowledge of the stream data to be sent. The first could be especially the case of online and interactive video applications, where the server must dynamically perform service scheduling as new data are available. The second is especially the case of stored video applications, where the server can compute a priori the optimal burst durations for the entire service length.

For all the scenarios of interest, the *Temporal Observation Window* (TOW) should be properly set, where VBT can be applied. In this work we consider the TOW length of $W_S$ burst cycles, sliding by $N_B$ burst cycles. $\overline{T}_{on,opt}$ calculation is repeated each step until the end of all services. The first $N_B$ burst cycles optimized in the previous step are transmitted and $N_B$ new burst cycles are introduced in the optimization process of the following step. This sliding procedure is illustrated in Figure 7.

The parameters $W_S$ and $N_B$ influence the VBT performance. In fact, the choice of the TOW length depends on the delay constraints allowed by video playback. The scheduler must know all service frames to be stored in bursts before setting the optimal burst durations. The playback delay is then equal to the TOW length. For stored video applications the TOW length could also be set as the entire video length and the optimization process will require only one step. On the other side, a larger TOW allows VBT to span a higher number of $\overline{T}_{on}$ configurations, increasing the probability to find a lower minimum for TLF.

The optimal TOW length should be chosen as a compromise between the startup delay and the optimal configuration $\overline{T}_{on,opt}$ with minimum losses.

Also the slide length $N_B$ should be chosen as a compromise between the computational overhead and the optimal solution $\overline{T}_{on,opt}$. A smaller $N_B$ allows to more efficiently optimize burst durations as new service data are scheduled in bursts. In fact, service data to be scheduled in the last $N_B$ new burst cycles included in the $(n+1)^{th}$ step of Figure 7 refine also the burst durations of the previous $W_S - N_B$ burst cycles already calculated in the $n^{th}$ step. On the other side, a smaller $N_B$ means a higher computational overhead since the optimization process must be repeated each $N_B$ burst cycles. VBT computational complexity mainly depends on the number $n$ of services in the TOW.; it decreases for smaller TOW lengths and multiplexed services, and for higher slide lengths.

*3) The Optimization Method*

The solution to (20) in a generic TOW has to be found iteratively by numerical methods that find the minimum of a nonlinear bounded multivariable function. The TLF is in fact a nonlinear function of the $\overline{T}_{on}$ vector. Furthermore, burst durations can assume all the possible integer values in a limited interval; this introduces bounds on $\overline{T}_{on}$ and the subset of $\mathbb{N}_+^n$ in which the minimum is searched.

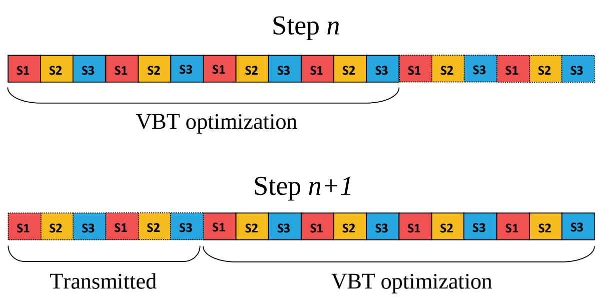


Figure 7. The TOW sliding procedure with $W_S = 4$ and $N_B = 2$ .

In this work the Sequential Quadratic Programming (SQP) non-linear optimization method is exploited [16]. The general problem is to find:

$$\min\{f(x)\} \text{ where } f:\mathbb{R}^n \to \mathbb{R} \tag{21}$$

under the constraints:

$$g(x) \le 0 \text{ where } g:\mathbb{R}^n \to \mathbb{R}^m \tag{22}$$

In our case, $f(x) = TLF(\overline{T}_{on})$ is the nonlinear function, and the constraints can be easily built by imposing that:

$$T_{on}^{(i,j)} \ge 1 \Rightarrow 1 - T_{on}^{(i,j)} \le 0, \begin{cases} \forall 1 \le \mathrm{i} \le \mathrm{N}_s \\ \forall 1 \le \mathrm{j} \le \mathrm{W}_s \end{cases} \tag{23}$$

since each burst duration cannot be less than 1 frame time. The basic SQP idea is to model the problem (21), (22) at a given approximate solution, say $x^k$, by a quadratic subproblem, and then to use the solution to this subproblem to construct a better approximation $x^{k+1}$, starting from an initial point , say $x^0$. The sequence of the approximated solution is hoped to converge to the final solution $x^*$. For our purposes, the SQP algorithm proposed in [17] is particularly suitable to be adopted. It provides a method for global convergence, starting from any $x^0$, if $f(x)$ is defined in a convex set. “Global convergence” means that the algorithm converges to some local solution from any remote starting point $x^0$. This is an advantage since the convergence to the solution is almost independent from $x^0$. Nevertheless it has not to be confused with the concept of “global solution”, that is that local solution $x^*$ providing the least value of $f(x)$. The second advantage of the chosen SQP algorithm is a relatively rapid convergence to $x^*$, making this algorithm suitable for real-time calculations. The definition of the quadratic subproblem model and of the so called “merit function”, that measures the step-by-step progress towards the solution, are out of the scopes of this work (please refer to [16] and [17] for further details).

Let us note that in general $f(x)$ is defined in $\mathbb{R}^n$. Nevertheless, $\overline{T}_{on}$ is a vector of strictly positive natural numbers, ranging from 1 to (theoretically) $\infty$. So the solution $\overline{T}_{on,opt}$ has to be found first in $\mathbb{R}_+^n$, where $\mathbb{R}_+^n$ is the subset of the strictly positive rational numbers, and then rounded so that $\overline{T}_{on,opt} \in \mathbb{N}_+^n$.

By [17] we know that the global convergence is obtained if $f(x)$ is defined in a convex set. To prove this assert for $TLF(\overline{T}_{on})$, let us introduce the following

*Definition 1*: A set $S$ is convex if and only if for any couple of points $\overline{x},\overline{y}$ in the set, all the points in the segment linking $\overline{x}$ and $\overline{y}$ are also in the set. That is, given:

$$\overline{x},\overline{y} \in S\ ;\ \beta \in \mathbb{R},\ 0 \le \beta \le 1 \tag{24}$$

then:

$$\overline{z} = \beta\overline{x} + (1-\beta)\overline{y} \in S\ . \tag{25}$$

We can now prove the following

*Theorem 1*: The subset $\mathbb{R}_+^n$ is a convex set.

*Proof*: A generic point $\overline{a} = (a_1, a_2, ..., a_n)$ belongs to $\mathbb{R}_+^n$ if and only if:

$a_i > 0,\ 1 \le \mathrm{i} \le \mathrm{n}$

that can be written in compact form as:

$$\mathbf{I}\overline{a} > \mathbf{0}\ . \tag{26}$$

where $\mathbf{I}$ is the n x n Identity matrix and $\mathbf{0}$ is the null vector of $\mathbb{R}^n$. If we now consider two vectors $\overline{x},\overline{y} \in \mathbb{R}_+^n$, surely it will be:

$$\mathbf{I}\overline{x} > \mathbf{0}\ ;\ \mathbf{I}\overline{y} > \mathbf{0}\ . \tag{27}$$

Given now $\overline{z}$ as defined in (25), it will be:

$$\mathbf{I}\overline{z} = \beta\mathbf{I}\overline{x} + (1-\beta)\mathbf{I}\overline{y} > \mathbf{0} \tag{28}$$

by (27) and the second of (23). And (28) proves the statement.

## III. Numerical Results

To test VBT effectiveness several simulation have been performed by multiplexing four video streams ( $N_S = 4$ ) in different simulation scenarios. Experimental results have been obtained in a MATLAB environment. VBT performance has been compared with the TDM transmission proposed in [3] that considers a fixed burst duration for all services. We call this technique “Constant Burst Time (CBT) algorithm” for notation simplicity. The four chosen video streams, all of length 5.000 video frames, have different quality coding degrees. They are: a piece of the “Jurassic Park” film (MPEG-4 coded with high quality), a piece of a video clip (MPEG-4 coded with low quality), a piece of the “Star wars IV” film (MPEG-4 coded with high quality) and a piece of the “The silence of the lambs” film (MPEG-4 coded with medium quality). Their main statistics are resumed in TABLE 1.

The first proposed experiment shows the influence of the TOW length in losses calculation for three different values of $N_B$. The TOW length has been varied from $W_S = 4$ until $W_S = 10$ burst cycles, with a step of 1 burst cycle. A constant available bandwidth of 3 Mbps has been set for all services. The receiving buffer size is of 320 kB. Figure 8 depicts the total amount of VBT losses.

TABLE 1.
MAIN VIDEO TRACE STATISTICS FOR THE 4 CHOSEN SERVICES.

| | Jurassic Park | Video Clip | Star Wars IV | The Silence of the Lambs |
|---|---|---|---|---|
| Compression ratio (YUV:MP4) | 9.92 | 38.17 | 27.62 | 43.43 |
| Mean frame size (bytes) | 3.8e+03 | 1e+03 | 1.4e+03 | 8.8e+02 |
| Var frame size | 5.1e+06 | 1.3e+06 | 8.2e+05 | 1.1e+06 |
| Cov of frame size | 0.59 | 1.14 | 0.66 | 1.21 |
| Min frame size (bytes) | 72 | 31 | 26 | 28 |
| Max frame size (bytes) | 16745 | 9025 | 9370 | 11915 |
| Mean bit rate (bit/s) | 7.7e+05 | 2e+05 | 2.8e+05 | 1.8e+05 |
| Peak bit rate (bit/s) | 2.4e+06 | 8.5e+05 | 1.2e+06 | 1.8e+06 |
| Peak/Mean of bit rate | 3.15 | 4.29 | 4.29 | 10.07 |

As expected, losses decrease with $W_S$ increase because service data can be better distributed in a larger number of burst durations to reduce losses. Furthermore, losses increase with $N_B$ increase, because a smaller slide length allows a better refinement in the calculation of burst durations among subsequent steps. Let us note that for $W_S = 4$ and $N_B = 3$ losses are evaluated over non overlapped and uncorrelated TOWs; they are thus proportionally much higher than the other experimented cases.

In this simulation CBT performs much worse than VBT, so CBT losses have not been reported in Figure 8 to improve its readability. To calculate CBT losses, the same four services have been considered with the same available bandwidth information (3 Mbps). The TOW length has been set to the whole streams length to find the minimum of CBT losses independently from $W_S$. Then, burst durations have been increased from 3 to 100 frame times, with step 1 frame time. Total losses have been calculated in each step and its minimum has been found among all steps. Results show that the minimum of CBT losses is reached for a service burst duration of 17 frame times, with a burst cycle of 68 frame times; total losses are 103.2 Mbits, approximately an order of magnitude higher than the maximum amount of VBT losses (observed in Figure 8 for $W_S = 4$ and $N_B = 3$). This confirms the VBT consistent efficiency due to the dynamic variation of burst durations.

The second proposed experiment, represented in Figure 9, shows the influence of available bandwidth over loss calculation for CBT and VBT algorithms.

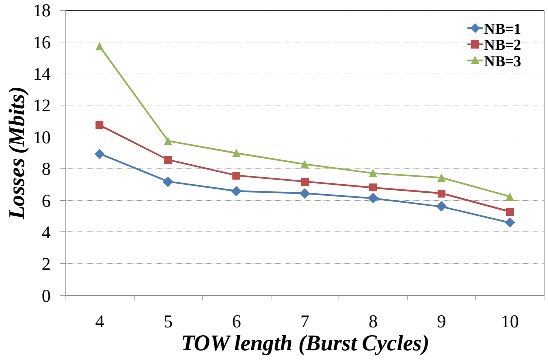


Figure 8. VBT total losses vs TOW length for three different slide lengths.

The same four pieces of video streams highlighted in Table 1 have been used for simulation.

For both algorithms, the available bandwidth assumes a constant value ranging from 1 to 5 Mbps. Receiving buffer size is 320 kB. The chosen TOW length for VBT is $W_S = 8$. Four different slide lengths ( $N_B = 1, 3, 5$ and $7$ ) have been chosen to show their influence over losses.

CBT losses have been evaluated as the minimum among all service burst durations ranging from 3 to 100 frame times, as previously explained. As expected, losses increase when decreasing the available bandwidth. Nevertheless, VBT performs better than CBT in all experimented scenarios. Loss differences between CBT and VBT and among the different VBT slide lengths are almost imperceptible for an available bandwidth of 1 Mbps. This means that the influence of both VBT variation of burst durations and the overlap degree among consecutive TOWs have an almost null effect in loss reduction. Losses are instead almost exclusively due to the very stringent bandwidth limitation. For increasing available bandwidth values, differences between VBT and CBT are consistently higher. In these cases the burst duration adjustment adopted by VBT is much more effective if compared with the static burst duration assignment adopted by CBT. The VBT different slide lengths have instead a relatively small influence on losses; differences are slightly more evident for an available bandwidth of 2 and 3 Mbps.

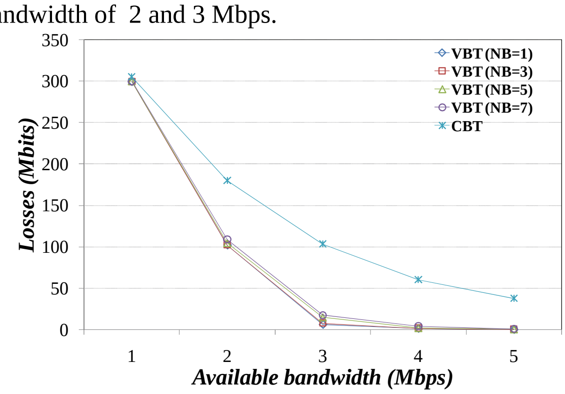


Figure 9. Losses vs available bandwidth for VBT and CBT schedules.

Let us note that for 4 and 5 Mbps, VBT experimented losses are almost null. Specifically, they are strictly null for 5 Mbps bandwidth and $N_B = 1$ and $3$, and under 0.7% for all other NBs. Let us note that for 4 and 5 Mbps, CBT losses are of 60.3 Mbits and 37.6 Mbits, respectively.

TABLE 2 depicts the influence of different service priorities. The same four services previously introduced have been used in this experiment. The receiving buffer size is 320 kB, with a constant available bandwidth of 3 Mbps for all services. In this experiment the TOW length is set to 8 burst cycles and the slide length to 1 burst cycle. The priority factor has been increased from 1 to 10 for the services 1 and 3. Consequently, losses in percentage decrease for services 1 and 3, and increase for services 2 and 4 that have a lower priority factor. Specifically, losses increase more for service 4 than for service 2 since service 4 has a higher bit rate variability than service 2 (see the last row of TABLE 1) that makes the optimal scheduling more difficult to perform.

Let us note that losses in percentage are not the same for all services even if they have all the same priority factor (see the 3rd column in TABLE 2); this could seem a violation of the fairness principle. Nevertheless, the optimization process is performed over TOWs of limited size, giving a suboptimal solution for service loss minimization. In other words, there could be $\overline{T}_{on}$ configurations where the single service losses are higher than the others, making the percentage loss balancing unfeasible. That is, the Jain index as defined in (17) is never equal to 1. Nevertheless VBT always chooses the fairest configuration among all feasible.

The last experiment shows the impact of receiving buffer size over losses for CBT and VBT, as represented in Figure 10. In this experiment, the same four pieces of video streams presented in TABLE 1 have been transmitted by VBT and CBT schedulers. Regarding VBT, the TOW length has been set to 8 burst cycles with a slide length of 1 burst cycle. Regarding CBT, services have been scheduled with the same procedure previously described. Losses have been evaluated for different client buffer sizes ranging from 128 to 1024 kB, with step of 128 kB. VBT losses have been evaluated for three different available bandwidth values (3, 4 and 5 Mbps), while the available bandwidth for CBT has been set to 5 Mbps to reduce losses for bandwidth limitation.

TABLE 2.
SERVICE LOSSES WITH AND WITHOUT DIFFERENT PRIORITY DEGREES.

| Service # | Service name | VBT (no priority) | VBT (with priority) | Priority factor |
|---|---|---|---|---|
| | | Lost Bits (%) | | |
| 1 | Jurassic Park | 1.434 | 0.17 | 10 |
| 2 | Video clip | 0.251 | 0.715 | 1 |
| 3 | Star Wars IV | 0.382 | 0.109 | 10 |
| 4 | The silence of the lambs | 1.523 | 4.123 | 1 |

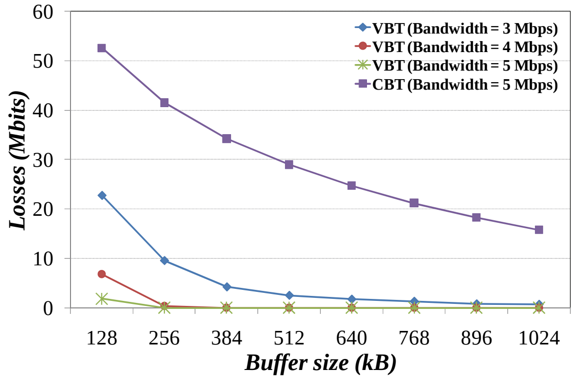


Figure 10. Losses vs receiving buffer size for VBT and CBT schedules.

As expected, losses decrease with buffer increase for both VBT and CBT. In fact, a larger buffer size allows storing more data at client buffer and reduces the loss probability for buffer underflow.

Again, VBT performs much better than CBT. Let us note that CBT losses are still present even if the available bandwidth is relatively high: with a buffer of 1024 kB CBT total losses are of 15.8 Mbits. VBT experimented losses, with the same available bandwidth of 5 Mbps, are instead null right from a buffer size of 256 kB. For a bandwidth of 4 Mbps losses are null right from a 384 kB buffer size.

## IV. CONCLUSIONS

In this paper the Variable Burst Time algorithm that schedules the transmission of compressed VBR multimedia data in DVB-H systems, has been presented and analyzed. It is an on-line algorithm that dynamically regulates all burst durations to reduce losses taking into account service data, receiving buffer size, available bandwidth information and service priority factors. Its flexibility and effectiveness in reducing losses are testified by several numerical results, obtained by comparison with the classical time sliced transmission proposed in the ETSI specifications of DVB-H standard in all the practical scenarios of interest.

## REFERENCES

[1] ETSI EN 300 744: “Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television”.
[2] ETSI EN 302 304: “Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H)”.
[3] ETSI TR 102 377: “Digital Video Broadcasting (DVB); DVB-H Implementation Guidelines”.
[4] ISO/IEC 13818-1 : “Information Technology - Generic Coding of Moving Pictures and Associated Audio Information - Part 1: Systems”.
[5] ETSI EN 301 192: “Digital Video Broadcasting (DVB); DVB Specification for Data Broadcasting”.
[6] 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.

[7] 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.

[8] W.-C. Feng, J. Rexford, "A Comparison of Bandwidth Smoothing Techniques for the Transmission of Prerecorded Compressed Video", *IEEE INFOCOM*, 1997.

[9] W.-C. Feng, J. Rexford, "Performance Evaluation of Smoothing Algorithms for Transmitting Prerecorded Variable-Bit-Rate Video", *IEEE Transactions on Multimedia*, Vol. 1, No.3, pp. 302-313, September 1999.

[10] S.Sen, J.L. Rexford, J.K. Dey, J.F. Kurose, D.F. Towsley, "Online Smoothing of Variable Bit-Rate StreamingVideo", *IEEE Transactions on Multimedia*, Vol.2, n.1, pp.37-48, March 2000.

[11] G.Cao, W.C.Feng, W.Singhal, "Online VBR Video Traffic Smoothing", *8th International Conference on Computer Communications and Networks*, pp. 502-507, 1999.

[12] C. Bewi, R. Pereira, M. Merabti, "Network Constrained Smoothing: Enhanced Multiplexing of MPEG-4 Video", *Proc. of 7th International Symposium on Computers and Communications (ISCC'02)*, pp. 114-119, July 2002.

[13] J.-Y. Le Boudec, P. Thiran, *Network Calculus: A Theory of Deterministic Queueing Systems for the Internet*, Book Springer Verlag, May 2004.

[14] W.-C. Feng, "Rate-Constrained Bandwidth Smoothing for the Delivery of Stored Video", *Proc. SPIE Multimedia Computing and Networking*, San Jose, CA, February 1997.

[15] R. Jain, D.M. Chiu, W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems", DEC Research Report TR-301, 1984.

[16] P.T. Boggs, J.W. Tolle, "Sequential Quadratic Programming", in *Acta Numerica*, Cambridge University Press, Cambridge, UK, pp. 1-51, 1995.

[17] S.P. Han, "A Globally Convergent Method for Nonlinear Programming", *Journal of Optimization Theory and Applications*, Vol. 22, N.3, pp. 297-309, July 1977.

**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 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 teaches fundamental courses in the field of telecommunications.

His research areas span the fields of LAN/MAN Architectures and Protocols, Optical Networks, Network Reliability, Cellular Radio Networks, Wireless LANs, Ad Hoc Networks, Telecommunication Services, etc.

**Domenico Striccoli** was born in Altamura (BA), Italy, on July 15, 1972. He 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 in Taranto as assistant professor in telecommunications. Actually he teaches fundamental courses in the field of Telecommunications at DIASS Department in Taranto.

His scientific interests span different aspects of telecommunications networks, including dynamic predictive bandwidth allocation, Video on Demand, MPEG coding techniques, smoothing algorithms, Call Admission Control schemes, Quality of Service, bandwidth estimation algorithms, etc.

**Francesco Capozzi** was born in Bari, Italy, on April 14, 1984. He received the first level Degree in Telecommunications Engineering in September 2005 and the master Degree in Telecommunication Engineering in April 2008, both with honors, from the Politecnico di Bari. Since September 2008, he collaborates with Nokia Siemens Networks at Aalborg University (Denmark) as research assistant. Since January 2009, he is a Ph.D. student in Electronics Engineering at the Politecnico di Bari.

His current research interests are mainly focused on the study of new Radio Resource Management algorithms for the next generations of cellular networks (LTE networks).