Abstract:
The growing importance of multimedia applications, which usually implies the transmission of Variable Bit Rate (VBR) video streams, needs an accurate analysis of bandwidth requirements according to specified Quality of Service (QoS) guarantees. In this paper an original algorithm has been proposed for statistical aggregate bandwidth estimation of smoothed Variable Bit Rate (VBR) video streams, respecting a specified loss probability (the QoS parameter taken into account). This algorithm is based on concepts of multi class network of queues whose service centers are represented by the bandwidth levels assumed by smoothed films. The mean service times and the transition probabilities between service centers are directly derived from real video traces. The analytical results have been compared with simulation results and with the results obtained by another analytical method already present in the literature, i.e., the Chernoff bound, discussing the effectiveness of the proposed method.

Keywords: Video Distribution, VBR traffic, Smoothed Video Streams, Bandwidth Evaluation, Multi Class Queueing Networks.
I. INTRODUCTION

In this study we will refer to applications like Video on Demand (VoD), Distance Learning, Internet video broadcast, etc [1]. A common aspect of such applications is the transmission of video streams that require a sustained relatively high Variable Bit Rate (VBR) bandwidth with a stringent Quality of Service (QoS) guarantee. An efficient delivery of video streams to final users while sharing network resources, guaranteeing the needed QoS, requires specific planning of network infrastructures and suitable Admission Control schemes [2-4] based on a statistical estimation of the aggregate bandwidth needed by a certain number of video streams.

As well known, standard video coding techniques (e.g., MPEG, H323, etc.) usually produce a VBR traffic. As an example, in Figure 1 the bit rate of 32,000 frames of an MPEG-1 coded trace of the “James Bond” film is reported. As highlighted in [4, 6] and as will be shown shortly, a consistent gain in network resource utilization can be obtained by exploiting smoothing, which will likely be implemented in real systems [4, 6]. The smoothing algorithms reduce the peak rate and variability of each stream present in the network exploiting a transmission buffer that regularizes the bit flow and a client buffer from which the original unsmoothed video stream is extracted, decoded and displayed. Obviously, the bit rate must be chosen appropriately in order to avoid buffer overflow or underflow. As an example, in Figure 2 the bit rate of the “James Bond” film, smoothed with a client buffer of 1024 Kbytes, is reported.

![Figure 1. 32.000 frames of the film “James Bond” without smoothing.](image-url)
As can be inferred, the smoothing technique consistently reduces the peak rate and the rate variability, reducing also the amount of bandwidth required by statistically multiplexed flows, as will be shown in Section III. In particular, the greater will be the dimension of the client buffer utilized for the video smoothing, the longer will be the constant bit rate pieces of the video stream and thus the longer will be the corresponding time intervals. It is worth noting that smoothed video streams, characterized by a small number of long CBR pieces, represent exactly the source traffic model exploited in [5].

Various smoothing techniques have been proposed in literature [6, 7], nevertheless the study developed in this paper, as will be clear in the sequel, is valid whatever smoothing technique is exploited.

A significant problem in the considered systems is the aggregate bandwidth estimation needed for planning the transmission infrastructure and for admission control. In this paper, an efficient algorithm of aggregate bandwidth estimation, while respecting the QoS parameter given by the loss probability, is proposed and analyzed, in the hypothesis of bufferless systems. Such a hypothesis can be supported by the following considerations. First of all, it has been shown [8, 9] that large buffers are of substantial benefit in reducing losses in the high frequency domain and the only effective way to reduce losses in the low frequency domain is to allocate sufficient bandwidth for each network link. Since the smoothed traffic mainly has a slow time scale variability, the adoption of large network buffers is not useful in this scenario. A second consideration is that the employment of large
buffers may introduce delays along the network nodes that can be intolerable for the correct delivery of video streams and thus cannot meet the QoS specifications.

Another significant aspect must be taken into account in the analysis of these systems. In fact, several studies on the bursty nature of the VBR video traffic that characterizes this kind of application highlight some important properties of compressed video sources. In particular, it has been shown [10-12] that the MPEG video traffic exhibits properties of self-similarity and Long Range Dependence (LRD) that may require sophisticated analytical models to adequately take into account the network traffic behavior. However, since in this case small network buffers must be utilized, as discussed in [13], long range dependence does not have a predominant effect on VBR coded videos. As a matter of fact, in this paper the network switch will be modeled as a bufferless multiplexer, without exploiting long range dependence techniques.

The proposed algorithm models the entire system of video streams as a multi class network of queues [14-16]. The service centers of the system are identified with all possible bit rate levels assumed by the considered smoothed films. Since there is no limit to the number of video streams that can assume a specific bit rate level, the resulting queuing system can be considered with infinite servers. The state probability is calculated under the hypothesis of statistical independence among video streams, and the aggregate bandwidth is derived with a statistical approach, as shown in the next sections.

Different approaches on statistical estimation of bandwidth occupied by a number of video streams entering a bufferless multiplexer have been proposed in [4, 17, 18]. In [4] a Chernoff-Bound-Based admission control method, valid in both cases of a single type of video streams and different types of video streams, estimates the aggregate bandwidth for independent video sources. In [17] three different methods for admission control schemes, that make use of Large Deviation (LD) approximations, the Normal Approximation and the Chernoff bound have been evaluated. They have been tested on the simplified case of a single type of video trace, the “Star Wars” film.

The analytical results provided by the algorithm proposed in this paper have been compared with the Chernoff bound algorithm as described in [4], for different values of loss probability and smoothing buffer. In particular, the Chernoff-Bound-Based admission control algorithm has been implemented in different ways, as will be explained in detail in Section III. In the first case, the bit rate range of each type of video stream has been divided in K bins (the total number of bins has been chosen the same for all the types of video streams) all equal in size. Different values of K have been taken into account, from a minimum of 5 to a maximum of 20 bins. The marginal probabilities have
been calculated using the histogram method, as illustrated in [4]. In the second case, the marginal
distributions of the different types of video streams have been described considering a number of
equispaced bins equal to the number of bandwidth levels assumed by the smoothed streams (that is
generally different for each type of video stream), while in the third case the exact bandwidth values
of each type of video stream, as produced by the smoothing algorithm, have been adopted as input
data to the Chernoff bound algorithm.

The paper is structured as follows. In Section II the aggregate bandwidth is calculated exploiting
an approach based on queueing networks. In particular, after a system description, in Section II.1, the
queueing networks solution is presented in Section II.2, while the algorithm for aggregate bandwidth
evaluation is explained, in the general case in which there are different types of video streams, in
section II.3. In section III some numerical results are presented, comparing the analytical results
provided by the algorithm with simulation results and with results obtained by the Chernoff bound.
Finally, in section IV some conclusions are given.

II. STATE PROBABILITY EVALUATION

II.1 System description

Let us suppose to have \( F \) different types of smoothed films in the video server. As soon as a
generic user chooses a generic film, let us say, the \( f^{th} \) type of film, with probability \( \beta_f \)
\((1 \leq f \leq F, \beta_1 + \ldots + \beta_F = 1)\), a new video stream starts to be transmitted competing for network
resources.

Globally, let us suppose that in the network there are generically \( N \) video streams. Our purpose is
to evaluate the aggregate bandwidth necessary to transmit all the \( N \) video streams while respecting
the considered QoS parameter, i.e., the loss probability.

Such a problem can be solved exploiting queueing networks theory with different classes of
customers [15, 16]. Each service center is represented by the bit rate assumed by each smoothed
film. Thus, the total number of service centers, indicated generically by \( M \), will be equal to the
totality of bit rate levels assumed by all the types of stored films. We generically call \( \lambda_m \),
\( 1 \leq m \leq M \), the bit rate needed by a video stream in this state. If more than one type of video stream
assumes the same bit rate, there will be a single service center to keep into account the common bit
rate assumed by all the considered video streams. Anyway, to take into account the possibility that a
single video stream can assume the same bit rate in two different time intervals, we suppose that each
service center can have different classes of customers. In fact, the service center corresponding
generically to the $m^{th}$ bit rate will have a number of different classes of customers equal to all the
time intervals in which the $m^{th}$ bit rate is assumed, for each type of video stream. If we model the
entire system of the $N$ video streams in this way, the effective dynamic evolution of the bandwidth
occupation can be kept adequately into account.

Supposing that there are generically $F$ different types of films and $M$ different bit rates, the system
state can be represented by:

$$(n_{111}, n_{111}, ..., n_{1F1}, ..., n_{1FM}, ..., n_{MF1}, ..., n_{MF2r}) = \{n_{jk}\}$$  (1)

This state vector describes all the $n_{ik}$ users (i.e., video streams) that are in the service center $i$ (and so they are using a bit rate $\lambda_i$) of the film type $j$ and that are of class $k$, for $1 \leq i \leq M$, $1 \leq j \leq F$ and $1 \leq k \leq r_{ij}$, $r_{ij}$ is the total number of classes that the service center $i$ of the film type $j$
can assume. It is clear that $r_{ij}$ represents the number of times in which the bandwidth level $\lambda_i$
appears in the film of type $j$. Obviously, $\sum_i \sum_j \sum_k n_{jk} = N$, i.e., there are always $N$ streams in the
network.

At the end of a generic video stream, another video stream starts immediately, the $f^{th}$ type of video
stream, with probability $\beta_f$. The simplified case of a single type of video stream can be considered
as a particularization of the general case of $F$ types of video streams, with $F = 1$. In each case, it is
necessary to find the probability that the entire system of $N$ video streams can occupy a given
aggregate bandwidth. The analytical approach to solve this problem is presented in the next sections.

II.2 Queueing networks solution

As explained in the previous section, our model corresponds to a network of queues with different
classes of customers for each service center. In particular, in our approach each of the service centers
is identified with one of the bandwidth levels which can be assumed by any type of video stream.
Since in our model there is no limit to the number of video streams that can utilize contemporarily a
given bandwidth level, we will consider each service center with infinite servers. Furthermore, the
probability that a user leaves the service center $m$ of the $f^{th}$ type of video stream of class $k$ and goes to
the service center $i$ of the $j^{th}$ type of video stream of class $l$ is represented by the transition probability
$p_{m,f:k; i,j,l}$. Each of these probabilities represents the transition probability from the $m^{th}$ bandwidth
level of the $f^{th}$ type of film of class $k$ to the $i^{th}$ bandwidth level of the $j^{th}$ type of film of class $l$.
Each of these probabilities can assume only the values 0, 1 or $\beta_f$ ($1 \leq f \leq F$) and can be easily
derived observing the sequence of the bandwidth levels in each film. In particular, from the analysis of the temporal evolution of the single video trace, it can be noted that the probability $p_{m,f,k,i,j,l}$ assumes the value 1 if $f=j$ and the two bandwidth levels $m$ of class $k$ and $i$ of class $l$ are temporally consecutive, otherwise the same probability assumes the value 0. If the bandwidth level $m$ is the last one (after that the film finishes), the probability $p_{m,f,k,i,j,l}$ assumes the value $\beta_j$ if the bandwidth level $i$ is the first one of the $j^{th}$ type of film, and 0 otherwise.

Since only the temporal evolution of the single source is analyzed in detail for the evaluation of the probabilities $p_{m,f,k,i,j,l}$, all temporal dependencies among video streams are not taken into account. From this last observation we can argue that the proposed method is suitable only for independent streams. As a matter of facts, in this paper all the issues of dependent streams [4, 19] have not been taken into account.

To derive the state probability, we have to solve, as first step, a homogeneous system given by the following flow balance equations [15]:

$$
\sum_{m=1}^{M} \sum_{f=1}^{F} \sum_{k=1}^{r} e_{mfg} p_{m,f,k,i,j,l} = \sum_{i=1}^{M} \sum_{j=1}^{F} \sum_{l=1}^{r} e_{ijkl} p_{m,f,k,i,j,l}, \quad 1 \leq i \leq M, 1 \leq j \leq F, 1 \leq l \leq r_j. \quad (2)
$$

Each term $e_{ijl}$ represents the relative arrival rate of class $l$ customers to the service center $i$ of the $j^{th}$ type of video stream. Calling synthetically $\{n_{ijl}\}$ the system state as expressed in (1), the corresponding state probability for a network of queues with infinite servers is expressed in product form as follows [15]:

$$
p(n_{ijl}) = \frac{1}{G} \prod_{i=1}^{M} \prod_{j=1}^{F} \prod_{l=1}^{r} \frac{1}{n_{ijl}} \left( \frac{e_{ijl}}{\mu_{ijl}} \right)^{n_{ijl}} \quad (3)
$$

where $G$ is a normalization constant, $e_{ijl}$ are a solution of the homogeneous system (2) and $1/\mu_{ijl}$ are the mean service times of the class $l$ customer at the service center $i$. This quantity represents the time of permanence of the $j^{th}$ type of video stream in the bandwidth level $\lambda_i$ of class $l$.

We can express the state probability in a more synthetic way. In particular, we can consider only the number of users that are in the $i^{th}$ bandwidth level of the $j^{th}$ type of video stream, whose state probability can be obtained by summing together the state probability of all classes of customers belonging to the same bandwidth level of the same type of video stream. Exploiting the multinomial formula, we have:

$$
p(n_{11},\ldots,n_{1F},\ldots,n_{M1},\ldots,n_{MF}) = p(n_{ij}) = \frac{1}{G} \prod_{i=1}^{M} \prod_{j=1}^{F} \frac{1}{n_{ij}} \left( \sum_{l=1}^{r} \frac{e_{ij}}{\mu_{ij}} \right)^{n_{ij}}
$$
or more synthetically

\[ p(n_i) = \frac{1}{G} \prod_{i=1}^{M} \prod_{j=1}^{F} \rho_{ij}^{n_{ij}}, \rho_{ij} = \sum_{l=1}^{r_j} e_{il}^{(i)} \]  

(4)

From (4) we can derive the state probability that refers only to the M bit rates assumed by all video streams, simply through the following expression:

\[ p(n_i) = p(n_1, \ldots, n_M) = \frac{1}{G} \sum_{n_{ij}+\ldots+n_{ij} = n_j} \prod_{j=1}^{F} \rho_{ij}^{n_{ij}} \sum_{n_{ij}+\ldots+n_{ij} = n_M} \prod_{j=1}^{F} \rho_{ij}^{n_{ij}} \]

and after some simple algebra:

\[ p(n_1, n_2, \ldots, n_M) = \frac{1}{G} \prod_{i=1}^{M} \left( \sum_{j=1}^{F} \rho_{ij} \right)^{n_i} = \frac{1}{G} \prod_{i=1}^{M} \frac{1}{n_i!} \rho_{i}^{n_i} \]

(5)

with \( \rho_i = \sum_{j=1}^{F} \rho_{ij} \). The normalization constant G can be derived from (5) by imposing that:

\[ \sum_{n_1+n_2+\ldots+n_M = N} p(n_1, n_2, \ldots, n_M) = 1 \]

from which, exploiting again the multinomial formula, we have:

\[ G = \frac{1}{N!} \left( \sum_{i=1}^{M} \rho_i \right)^{N} \]

(6)

II.3 Bandwidth estimation

Given the system state (5), the aggregate bandwidth needed in correspondence of a state probability \( p(n_i) \) is given by \( \Lambda = \sum_{i=1}^{M} n_i \lambda_i \).

Thus the aggregate bandwidth that assures a given loss probability in the general case of N video streams can be evaluated considering the sum of all state probabilities that give an aggregate bandwidth lower than \( \Lambda_s \). We have

\[
\begin{align*}
\left\{ \begin{array}{l}
p(\Lambda \leq \Lambda_s) = (1 - p_i) = \sum_{n_i=0}^{N} \cdots \sum_{n_M=0}^{N} \frac{1}{G} \prod_{i=1}^{M} \rho_{i}^{n_i} \\
\sum_{i=1}^{M} n_i \lambda_i \leq \Lambda_s \\
\sum_{i=1}^{M} n_i = N
\end{array} \right. \\
\end{align*}
\]

(7)

where \( p_i \) is the loss probability, while the unknown is the aggregate bandwidth \( \Lambda_s \).
The solution of (7) is explained in detail in Appendix A and consists of solving the equation:

\[
(1 - p_l) = \frac{1}{G} \frac{1}{N!} \Lambda_S^N \frac{d^{\Lambda_S}}{dz_2^{\Lambda_S}} \left[ \frac{z_2^{\Lambda_S + 1} - 1}{z_2 - 1} \left( \sum_{i=1}^{M} \rho_i z_2^{\lambda_i} \right)^N \right]_{z_2=0}.
\]

It can be noted that the term:

\[
\frac{1}{\Lambda_S^N} \frac{d^{\Lambda_S}}{dz_2^{\Lambda_S}} \left[ \frac{z_2^{\Lambda_S + 1} - 1}{z_2 - 1} \left( \sum_{i=1}^{M} \rho_i z_2^{\lambda_i} \right)^N \right]_{z_2=0}
\]

represents the \( \Lambda_S^n \) coefficient of polynomial:

\[
G(z_2) = \frac{z_2^{\Lambda_S + 1} - 1}{z_2 - 1} \left( \sum_{i=1}^{M} \rho_i z_2^{\lambda_i} \right)^N = (1 + z_2 + \ldots + z_2^{\Lambda_S}) \left( \sum_{i=1}^{M} \rho_i z_2^{\lambda_i} \right)^N
\]

We generically call this term with \( a_{\Lambda_S} \). To find the quantity \((1 - p_l)\) we have to calculate \( a_{\Lambda_S} \) of \( G(z_2) \) and the correspondent exponent of \( z_2 \) is the needed aggregate bandwidth \( \Lambda_S \) in correspondence of the value \((1 - p_l)\).

It can be noted that the aggregate bandwidth \( \Lambda_S \) is unknown, while the loss probability \( p_l \) is known. To derive \( \Lambda_S \) knowing the value of \( p_l \), let us suppose, without loss of generality, that all the bandwidth values \( \lambda_i \) are ordered in an ascending way \( \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_M \) and suppose to arbitrarily fix the value of \( \Lambda_S \), which varies from a minimum of \( N\lambda_1 \) to a maximum of \( N\lambda_M \). Our purpose is to find all the contributes of the function \( G(z_2) \) to the term \( a_{\Lambda_S} \). In particular, let us suppose to have entirely developed the polynomial \( P_2(z_2) \), that can be written as follows:

\[
P_2(z_2) = c_{N\lambda_1} z_2^{N\lambda_1} + c_{N\lambda_1+1} z_2^{N\lambda_1+1} + \ldots + c_{N\lambda_M} z_2^{N\lambda_M}
\]

where the coefficients \( c_{N\lambda_1} \), \( c_{N\lambda_1+1} \), \ldots \( c_{N\lambda_M} \) derive from the development of the \( N^{th} \) power of the polynomial \( \sum_{i=1}^{M} \rho_i z_2^{\lambda_i} \).

the polynomial \( P_1(z_2) \) has then to be multiplied by the factor:

\[
P_1(z_2) = (1 + z_2 + \ldots + z_2^{\Lambda_S})
\]

Let us suppose, for the moment, that \( \Lambda_S = N\lambda_1 \). After the product between \( P_1(z_2) \) and \( P_2(z_2) \) we obtain that the coefficient of \( z_2^{\Lambda_S} \) is given by the product of the first coefficient of \( P_1(z_2) \), \( c_{N\lambda_1} \), for the first coefficient of \( P_2(z_2) \), that is 1. For this reason we obtain:
\[ P(\Lambda \leq \Lambda_s) = \frac{c_{\Lambda_s}}{N!G}. \]

Now let us suppose that \( \Lambda_s = N\lambda_1 + 1 \). The coefficient of \( z_2^{N\lambda_1+1} \) is given by the sum of two terms. The first of them is the product of \( c_{N\lambda_1} z_2^{N\lambda_1} \) of \( P_1(z_2) \) for the term \( z_2 \) of \( P_2(z_2) \), while the second is the product of \( c_{N\lambda_1+1} z_2^{N\lambda_1+1} \) of \( P_1(z_2) \) for the term 1 of \( P_2(z_2) \). For this reason, the coefficient of \( z_2^{\Lambda_s} \) is \( c_{N\lambda_1} + c_{N\lambda_1+1} \) and so:

\[ P(\Lambda \leq \Lambda_s) = \frac{c_{N\lambda_1} + c_{N\lambda_1+1}}{N!G}. \]

If we repeat the same procedure for increasing values of \( \Lambda_s \), we obtain the general formula:

\[ P(\Lambda \leq \Lambda_s) = (1 - p_1) = \frac{1}{N!G} \sum_{B=\Lambda_s}^{\Lambda_s^k} c_B \]

Since \( P(\Lambda \leq \Lambda_s) \) is a given value, to find \( \Lambda_s \) we simply have to sum the coefficients of

\[ P_2(z_2) = \left( \sum_{i=1}^{M} \rho_i z_2^{\lambda_i} \right)^N \]

until their sum reaches or is superior to the value \( N!G \cdot (1 - p_1) \), that is a priori known.

It is worth noting that the coefficients of \( P_2(z_2) \) can be obtained very easily exploiting the Fast Fourier Transform (FFT) [20]. In fact, these coefficients can be obtained by N-fold convolutions of the vector that contains all the coefficients of the polynomial \( \sum_{i=1}^{M} \rho_i z_2^{\lambda_i} \). If we transform this vector using the Discrete Fourier Transform (DFT), exploiting numerically the FFT, in the transformed domain the convolution operation described above corresponds to the elevation of each of the terms of the transformed vector to the \( N^{th} \) power. To derive \( P_2(z_2) \) we have to transform back the obtained vector to the original domain. Supposing that the vector of the coefficients of the polynomial \( \sum_{i=1}^{M} \rho_i z_2^{\lambda_i} \) is composed by \( \lambda_M + 1 \) terms, the polynomial \( P_2(z_2) \) will be composed by \( N\lambda_M + 1 \) terms. So, the initial vector to be transformed will be formed by the original \( \lambda_M + 1 \) terms padded with other \( (N-1)\lambda_M \) zeros.

**III. NUMERICAL RESULTS**

In this section some numerical results of the application of the proposed algorithm are presented, comparing them with corresponding results of the Chernoff bound approach and with simulation.
results. N independent video streams, deriving from the first 40,000 frames of three different types of films, “Star Wars”, “James Bond” and “Terminator 2”, have been considered, assuming that the transmission rate is 30 frames/s and that the probability of choosing one of the three types of video streams is the same \( \beta_1 = \beta_2 = \beta_3 = 1/3 \). Comparisons among the mentioned approaches have been performed for different values of smoothing buffer size, considering different values of loss probability. It has been pointed out in Section I that if the buffer size increases, the peak rate and the bit rate variability of the smoothed video streams is consistently reduced. It follows that more bandwidth resources can be saved if the video streams are smoothed with larger buffers values. This consideration can be verified observing Figure 3, where the comparison between analytical results and simulation results for 90 films of three different types is reported. Each of the curves refers to a specified value of the smoothing buffer and, for each curve, bandwidth values have been derived in correspondence of loss probabilities ranging from \( 10^{-7} \) to \( 10^{-2} \). It can be easily noted that for each value of the smoothing buffer size, the aggregate bandwidth estimated with the analytical method slightly overestimates simulation results. This is especially true for lower values of loss probability. For higher loss probability, the aggregate bandwidth obtained with the algorithm almost coincides with the correspondent value of simulated bandwidth.

![Figure 3. Analytical bandwidth results vs loss probability for 90 films of three different types and comparison with simulation.](image)

The analytical results obtained have been compared also with the Chernoff-Bound-Based method in the hypothesis of independence of video traffic [4], for different values of loss probability and smoothing buffer. The Chernoff bound approach calculates the probability \( p(\Lambda \leq \Lambda_s) \) to exceed the
bandwidth level $\Lambda_s$, assumed as unknown, given a specified value of loss probability. In particular, given $n_f$ sources of type $f$, with $1 \leq f \leq F$, it is supposed that each source is characterized by a stationary distribution given by the random variable $b_f$ that represents the bit rate distribution of the $f^{th}$ type of source. Supposing that $b_f$ can assume generically the $K_f$ states $r_1^{(f)} \leq r_2^{(f)} \leq \ldots \leq r_{K_f}^{(f)}$, the probability that the $f^{th}$ source of type $f$ occupies a bit rate $b_{j_f}$, for $1 \leq j \leq n_f$, is indicated by:

$$p(b_{j_f} = r_k^{(f)}) = p_k^{(f)}.$$ 

Assuming that all the $b_{j_f}$ are independent, the aggregate bandwidth $\Lambda_s$ exploiting the Chernoff bound method can be calculated as follows [4]:

$$\Lambda_s = \sum_{f=1}^{F} n_f \frac{d}{d\theta} \left[ M_f(\theta^\ast) \right] \left[ M_f(\theta^\ast) \right]$$

(9)

where $\theta^\ast$ is the solution to the non linear equation:

$$\log(p_f) = \Omega(\theta) - \theta \frac{d}{d\theta} \left[ \Omega(\theta) \right] - \log(\theta) - 0.5 \log \left[ \frac{d^2}{d\theta^2} \left[ \Omega(\theta) \right] \right] - 0.5 \log(2\pi)$$

and $\Omega(\theta) = \sum_{f=1}^{F} n_f \log M_f(\theta) \cdot M_f(\theta) = \sum_{k=1}^{K_f} \frac{p_k^{(f)}}{\beta_f} e^{n_s(\theta)}$.

The Chernoff bound algorithm has been implemented utilizing different approaches to describe the marginal distributions of the three mentioned types of video streams. In the first approach, the entire bit rate range of the three types of film has been divided into a number of bins, generically called $K$, all equal in size. The size of the histogram bins has been chosen the same for all the types of video sources, so that $K_1 = K_2 = K_3 = K$. The size of each bin has been calculated dividing by $K$ the greatest of the peak rates of the three types of video streams. The histogram method [4] has been utilized to derive the marginal probabilities $p_k^{(f)}$ and the correspondent bit rate $r_k^{(f)}$ is identified with the highest bit rate of the considered histogram bin, corresponding to its right bound. Each of the $p_k^{(f)}$ has been weighted for the correspondent value of $\beta_f$ and the (9) has been exploited with $F=1$, considering a unique marginal distribution that assumes generically $K = K_1 + K_2 + K_3$ states, with probabilities $p_k' = \beta_f p_k^{(f)}$ and correspondent bit rates $r_k = r_k^{(f)}$, for $1 \leq k' \leq K'$ and $1 \leq k_f \leq K_f$. The chosen values of $K$ are $K=5$, $K=10$, $K=15$ and $K=20$.

In the second approach, for each type of video stream, the marginal probabilities have been derived dividing the bit rate range of each type of film into a number of bins equal to the number of
bit rate levels experimentally observed for each video stream. It derives that the bin size is the same for each type of film, but the number and the size of histogram bins are different if the type of video stream changes. It has been verified that in this case $K_1 \neq K_2 \neq K_3$ for each value of smoothing buffer adopted. The Chernoff bound algorithm has been exploited utilizing (9) for $F=1$ and deriving all the values of $p_k^{i\ell}$ and $r_k$, as explained in the first approach, by considering a unique marginal distribution that assumes the $K' = K_1 + K_2 + K_3$ bit rates $r_k$ with probability $p_k^{i\ell}$. This second approach utilizes always the highest number of histogram bins for each type of video stream, so we have called this approach “Chernoff K max”.

In the third approach, each value of $r_k^{i(f)}$ has been identified exactly with a bit rate value experimentally observed from the real video traces, while the correspondent probability $p_k^{i(f)}$ has been obtained dividing the number of video frames with bit rate $r_k^{i(f)}$ for the total number of video frames of each stream (40.000 frames). The values of all the $K'$ probabilities $p_k^{i\ell} = \beta_j p_k^{i(f)}$ have been derived and (9) has been utilized with $F=1$. Since the bit rates $r_k$ are not approximated in this case, this approach has been identified with the words “Chernoff exact”.

The results obtained by the mentioned approaches that make use of the Chernoff bound algorithm have been compared with the analytical results obtained with the proposed algorithm and with some simulation results, for different values of loss probability and smoothing buffer.

The simulation results of aggregate bandwidth, in correspondence of a given value of loss probability and a specified number N of video streams, have been derived as follows. For each transmitted frame, all the N bit rates of the video streams are summed. Then a bandwidth value is appropriately chosen in such a way that the number of frames in which the aggregate bandwidth is higher than the chosen bandwidth, divided by the total number of frames considered in the simulation, gives us the loss probability of interest. To guarantee the presence of N video streams, it has been assumed that when a stream finishes, another stream starts immediately from the beginning. The video stream that starts is one of the three considered types, with probability $\beta_i = 1/3$.

The starting points of the N streams have been chosen randomly in a time interval equal to the entire duration of the video streams, simulating in this way independence among all streams. From these considerations derives that the number of video streams of each type varies continuously along the entire simulation duration, while the total number N of video streams remains unchanged. The duration of a simulation run has been chosen equal to 20.000.000 video frames. In correspondence of each number of video streams, the simulation run has been repeated 20 times. The aggregate
bandwidth has then been calculated as the arithmetic mean of the 20 aggregate bandwidth values found.

In Figure 4 and 5 a comparison between the bandwidth results provided with our algorithm, the Chernoff bound algorithm implemented with the different approaches previously discussed and some simulation results is represented. The video streams have been smoothed using a buffer of size 1024 Kbytes. The total number of films has been varied from a minimum of 6 to a maximum of 90, with step of 6. In Figure 4, a loss probability of $10^{-6}$ has been chosen, while in Figure 5 the considered value of loss probability is $10^{-5}$.

![Figure 4](image_url)

*Figure 4. Aggregate bandwidth for three types of film ($p_f = 10^{-6}$) calculated with different approaches and comparison with simulation.*
Figure 5. Aggregate bandwidth for three types of film \( (p_e = 10^{-5}) \) calculated with different approaches and comparison with simulation.

From the analysis of Figure 4 and Figure 5, it can be observed that the bandwidth results obtained with the Chernoff bound algorithm and the algorithm developed in this paper are quite conservative if compared with simulation results. In particular, if the number of histogram bins increases, the marginal distributions of the video streams are more accurately described and bandwidth results better approximate simulation results. The “Chernoff exact” curve does not consider any bit rate approximation and describes accurately the marginal distribution of the video streams, like the proposed algorithm. These last two curves better approximate simulation results. Nevertheless, it can be noted that the proposed algorithm behaves slightly better than the “Chernoff exact” curve, even if the differences among bandwidth values are small. This last result can be better appreciated in Table 1, where all the bandwidth differences between the “Chernoff exact” algorithm and the proposed algorithm are better highlighted for all the considered smoothing buffer sizes. The bandwidth results refer to 90 video streams of the three mentioned types.

Table 1. Bandwidth values for the “Chernoff exact” method and the algorithm. The bandwidth values are in Mbit/s.

<table>
<thead>
<tr>
<th>Loss probability</th>
<th>Chernoff 1024 Kbytes</th>
<th>Algorithm 1024 Kbytes</th>
<th>Chernoff 512 Kbytes</th>
<th>Algorithm 512 Kbytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(10^{-7})</td>
<td>52.060</td>
<td>52.044</td>
<td>52.436</td>
<td>52.420</td>
</tr>
<tr>
<td>(10^{-6})</td>
<td>50.987</td>
<td>50.967</td>
<td>51.321</td>
<td>51.299</td>
</tr>
<tr>
<td>(10^{-5})</td>
<td>49.822</td>
<td>49.795</td>
<td>50.110</td>
<td>50.082</td>
</tr>
<tr>
<td>(10^{-4})</td>
<td>48.534</td>
<td>48.496</td>
<td>48.776</td>
<td>48.735</td>
</tr>
<tr>
<td>(10^{-3})</td>
<td>47.076</td>
<td>47.015</td>
<td>47.267</td>
<td>47.203</td>
</tr>
<tr>
<td>(10^{-2})</td>
<td>45.359</td>
<td>45.242</td>
<td>45.494</td>
<td>45.373</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Loss probability</th>
<th>Chernoff 256 Kbytes</th>
<th>Algorithm 256 Kbytes</th>
<th>Chernoff 64 Kbytes</th>
<th>Algorithm 64 Kbytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>(10^{-7})</td>
<td>52.837</td>
<td>52.819</td>
<td>53.965</td>
<td>53.944</td>
</tr>
<tr>
<td>(10^{-6})</td>
<td>51.677</td>
<td>51.655</td>
<td>52.675</td>
<td>52.649</td>
</tr>
<tr>
<td>(10^{-5})</td>
<td>50.421</td>
<td>50.391</td>
<td>51.284</td>
<td>51.251</td>
</tr>
<tr>
<td>(10^{-4})</td>
<td>49.038</td>
<td>48.996</td>
<td>49.760</td>
<td>49.713</td>
</tr>
<tr>
<td>(10^{-3})</td>
<td>47.477</td>
<td>47.411</td>
<td>48.050</td>
<td>47.976</td>
</tr>
</tbody>
</table>
In Figure 6 and Table 2 the comparison among bandwidth values obtained with different Chernoff bound approaches and the proposed algorithm is represented. The total number of video streams is fixed to 90 of three different types, while the loss probability values vary from a minimum of $10^{-7}$ to a maximum of $10^{-2}$. In Figure 6 a smoothing buffer of 512 Kbytes is considered, while in Table 2 the adopted smoothing buffer size is of 256 Kbytes.

![Figure 6. Aggregate bandwidth for 90 films of three types. The smoothing buffer size is 512 Kbytes.](image)

### Table 2. Aggregate bandwidth for 90 films of three types. The smoothing buffer size is 256 Kbytes. The bandwidth values are in Mbit/s.

<table>
<thead>
<tr>
<th>Loss probability</th>
<th>Chernoff K=20</th>
<th>Chernoff K max</th>
<th>Chernoff exact</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-7}$</td>
<td>59.391</td>
<td>53.566</td>
<td>52.837</td>
<td>52.819</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>55.175</td>
<td>52.407</td>
<td>51.677</td>
<td>51.655</td>
</tr>
<tr>
<td>$10^{-5}$</td>
<td>53.856</td>
<td>51.152</td>
<td>50.421</td>
<td>50.391</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>52.403</td>
<td>49.769</td>
<td>49.038</td>
<td>48.996</td>
</tr>
<tr>
<td>$10^{-3}$</td>
<td>50.762</td>
<td>48.210</td>
<td>47.477</td>
<td>47.411</td>
</tr>
<tr>
<td>$10^{-2}$</td>
<td>48.836</td>
<td>46.381</td>
<td>45.547</td>
<td>45.521</td>
</tr>
</tbody>
</table>

From Figure 6 and Table 2 it can be observed the difference between the “Chernoff exact” method and the proposed algorithm. This difference has been noted also for other numbers of video streams.
streams and loss probability values, even if the correspondent results have been omitted in this paper for brevity. We can say in conclusion that the analytical results provided by our algorithm are closer to simulation results than the other analyzed methods, at the same time slightly overestimating simulation results and thus respecting the specified QoS guarantees.

IV. CONCLUSIONS

In this paper we have presented an original method for statistical evaluation of the aggregate bandwidth occupied by a certain number of video streams that share network resources, respecting the QoS parameter of loss probability, in the hypothesis of bufferless systems. The study is based on queueing networks. In particular, each of the bandwidth levels of all the types of video streams has been identified with a service center with different classes of customers.

The technique utilized is totally new and it is based on the a priori knowledge of the sequences of bandwidth levels of each video streams. The dynamic evolution of the aggregation of all streams is taken into account through the probabilities of transition from a service center to another. The algorithm is very easy to implement, and the solution is very fast to obtain utilizing the FFT algorithm.

The proposed method has been tested in the general case in which there are different types of video streams, comparing the obtained analytical results with simulation results and with different approaches of the Chernoff bound algorithm for different smoothing buffer size and loss probability. In each analyzed case, the obtained analytical results are slightly conservative if compared with simulation results, providing the needed QoS guarantees.

APPENDIX A

In this section the solution of the system (7) is explained exploiting the integral method proposed in [21]. In particular, in the system (7) there is an equality constraint and an inequality constraint. Let us define the following functions:

\[ \delta(k) = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } k \neq 0 \end{cases} \]

representing generically the equality constraint, and:

\[ \Phi_{\delta}(k) = \begin{cases} 1 & \text{for } k \leq N \\ 0 & \text{for } k > N \end{cases} \]

representing instead the inequality constraint.

In the particular case of the constraints expressed in the system (7), we define the functions:
\[ \delta_1 = \delta (n_1 + \ldots + n_M - N) ; \]  
(A.1)

\[ \Phi_2 = \begin{cases} 
1 \text{ if } \sum_{i=1}^{M} n_i \lambda_i \leq \Lambda, \\
0 \text{ otherwise} 
\end{cases} \]  
(A.2)

So, the first equation of (7) can be rewritten including the two constraints expressed by \( \delta_1 \) and \( \Phi_2 \):

\[ (1 - p_i) = \sum_{n_i=0}^{\infty} \cdots \sum_{n_M=0}^{\infty} \frac{1}{G_i \prod_{i=1}^{M} n_i!} \delta_1 \Phi_2 \]  
(A.3)

The correspondent integral forms for (A.1) and (A.2) are respectively:

\[ \delta_1 = \int z_1^{(n_1+\ldots+n_M-1)} \left[ \frac{1}{z_1^{N+1}} \right] dz_1 \]  
(A.4)

\[ \Phi_2 = \int z_2^{(\sum_{i=1}^{M} n_i \lambda_i)} \left[ \frac{z_2^{\Lambda+1} - 1}{z_2^{\Lambda+1}(z_2-1)} \right] dz_2 \]  
(A.5)

where the contour integrals of (A.4) and (A.5) are calculated on the unit circle. (A.4) is justified considering the well known property of contour integrals:

\[ \int_a^b z^{-1}dz = \int_a^b f(z(t))f'(z(t))dt \]

and in the case of unit circle we have \( z = re^{i\theta} = e^{i\theta} \), for \( 0 \leq \theta \leq 2\pi \), so:

\[ \int_0^{2\pi} z^{-1}dz = \int_0^{2\pi} e^{(k-1)i\theta}j e^{i\theta}d\theta = \int_0^{2\pi} e^{jk\theta}d\theta = e^{j2k\pi} = \frac{-1}{k} \]  
(A.6)

The last equation is not null only for \( k=0 \), where assumes the value \( j2\pi \). In conclusion:

\[ \int z^{-1}dz = (j2\pi)\delta(k) \]  
(A.7)

The constant term \( j2\pi \) can be omitted, as will be more clear in the sequel, and so we obtain:

\[ \delta(k) = \int z^{-1}dz \]

The function \( \Phi_0(k) \) can be seen as the sum of \( N+1 \) discrete impulses:

\[ \Phi_0(k) = \sum_{i=0}^{N} \delta(k-i) = \sum_{i=0}^{N} \int z^{-i}dz = \int z^{-1} \left( \sum_{i=0}^{N} z^{-i} \right)dz \]

Considering (A.6) and remembering that \( \left( \sum_{i=0}^{N} z^{-i} \right) = \frac{z^{-(N+1)} - 1}{z^{-1} - 1} = \frac{1}{z^N} \left( \frac{z^{(N+1)} - 1}{z-1} \right) \), we obtain

(A.5).
Replacing (A.4) and (A.5) in (A.3) we have:

\[ P(\Lambda \leq \Lambda_s) = (1 - p_i) \frac{1}{G} \sum_{n_i = 0}^{\infty} \sum_{n_n = 0}^{\infty} \int \left[ \frac{1}{z_1^{N+1}} \int \left[ \frac{2^{z_2 - 1}}{z_2^{\Lambda_2 + 1}(z_2 - 1)} \prod_{i=1}^{M} \frac{\rho_i^{n_i}}{n_i!} \left( \frac{z_i}{z_2} \right)^{n_i} \right] d\zeta_2 dz_2 \right] \]

\[ = \frac{1}{G} \left[ \frac{1}{z_1^{N+1}} \int \left[ \frac{2^{z_2 - 1}}{z_2^{\Lambda_2 + 1}(z_2 - 1)} \prod_{i=1}^{M} \frac{\rho_i^{n_i}}{n_i!} \left( \frac{z_1}{z_2} \right)^{n_i} \right] d\zeta_2 dz_2 \right] \]

\[ = \frac{1}{G} \left[ \frac{1}{z_1^{N+1}} \int \left[ \frac{2^{z_2 - 1}}{z_2^{\Lambda_2 + 1}(z_2 - 1)} \prod_{i=1}^{M} \frac{\rho_i^{n_i}}{n_i!} \left( \frac{z_i}{z_2} \right)^{n_i} \right] d\zeta_2 dz_2 \right] \]

\[ = \frac{1}{G} \left[ \frac{1}{z_1^{N+1}} \int \left[ \frac{2^{z_2 - 1}}{z_2^{\Lambda_2 + 1}(z_2 - 1)} \prod_{i=1}^{M} \frac{\rho_i^{n_i}}{n_i!} \left( \frac{z_i}{z_2} \right)^{n_i} \right] d\zeta_2 dz_2 \right] \]

where it has been assumed that \( p_i = \sum_{i=1}^{\infty} \frac{e^z}{\mu_i} \).

Remembering that \( \sum_{n_i = 0}^{\infty} \left( \frac{\rho_i z_i^{\lambda_i}}{n_i!} \right)^{n_i} = e^{p_i z_i^{\lambda_i}} \), we obtain:

\[ P(\Lambda \leq \Lambda_s) = \frac{1}{G} \left[ \frac{1}{z_1^{N+1}} \int \left[ \frac{2^{z_2 - 1}}{z_2^{\Lambda_2 + 1}(z_2 - 1)} \prod_{i=1}^{M} e^{p_i z_i^{\lambda_i}} \right] d\zeta_2 dz_2 \right] \]

\[ = \frac{1}{G} \left[ \frac{1}{z_1^{N+1}} \int \left[ \frac{2^{z_2 - 1}}{z_2^{\Lambda_2 + 1}(z_2 - 1)} \prod_{i=1}^{M} e^{p_i z_i^{\lambda_i}} \right] d\zeta_2 dz_2 \right] \]

\[ = \frac{1}{G} \left[ \frac{1}{z_1^{N+1}} \int \left[ \frac{2^{z_2 - 1}}{z_2^{\Lambda_2 + 1}(z_2 - 1)} \prod_{i=1}^{M} e^{p_i z_i^{\lambda_i}} \right] d\zeta_2 dz_2 \right] \]

(A.8)

The expression (A.8) can be solved exploiting the residue theorem for all the two integral expressions included in (A.8). In particular, the residue theorem states that:

\[ \oint \frac{f(z)}{z} dz = (j2\pi) \sum_{i} \text{Res}(f, z_i) \]

where \( z_i \) are the poles of \( f(z) \) inside the \( \Gamma \) curve. The factor \( j2\pi \) is simplified when considering (A.7). Calling \( f(z_i) = \frac{1}{z_i^{N+1}} e^{z_i^{\lambda_i}} \sum_{i=1}^{M} \frac{\rho_i z_i^{\lambda_i}}{n_i!} \) from the residue theorem we have:

\[ \oint f(z_i) dz_i = \frac{1}{N!} \left. \frac{d^N}{dz_i^N} \left[ f(z_i) z_i^{N+1} \right] \right|_{z_i = 0} = \left( \sum_{i=1}^{M} \frac{\rho_i z_i^{\lambda_i}}{n_i!} \right)^N \]

since the only pole of \( f(z_i) \), of order \( N + 1 \), is in \( z_i = 0 \). Utilizing again the residue theorem for the second integral expression, it is finally found that
\[
P(\Lambda \leq \Lambda_s) = \frac{1}{G} \frac{1}{N!} \Lambda_s^N \left[ \frac{z^2}{z - 1} \left( \sum_{i=1}^{M} \rho_i z_i^\lambda \right)^N \right]_{z=1}
\]

(A.9)

Remembering the value of the constant term \( G \) as expressed in (6), the expression of \( P(\Lambda \leq \Lambda_s) \) is:

\[
P(\Lambda \leq \Lambda_s) = (1 - p_l) \frac{1}{N!} \sum_{i=1}^{M} \rho_i^N \left[ \frac{z^2}{z - 1} \left( \sum_{i=1}^{M} \rho_i z_i^\lambda \right)^N \right]_{z=1}
\]

(A.10)

in which the unknown is \( \Lambda_s \) while \( p_l \) is known a priori. Let us finally note that the numerator of (A.10) is the coefficient of the \( \Lambda_s^b \) term of the Taylor series of the polynomial:

\[
\frac{z^2}{z - 1} \left( \sum_{i=1}^{M} \rho_i z_i^\lambda \right)^N = \left( 1 + z + z^2 + \ldots + z^{\Lambda_s} \right) \left( \sum_{i=1}^{M} \rho_i z_i^\lambda \right)^N
\]

(A.11)

REFERENCES


