On Network Planning for Video Services Over IP Networks

S. Sharafeddine

To cite this article: S. Sharafeddine (2010) On Network Planning for Video Services Over IP Networks, International Journal of Computers and Applications, 32:3, 297-308
To link to this article: http://dx.doi.org/10.2316/Journal.202.2010.3.202-2671

Published online: 11 Jul 2015.
ON NETWORK PLANNING FOR VIDEO SERVICES OVER IP NETWORKS

S. Sharafeddine*

Abstract

Video transmission over IP requires very high data rate for a good performance. Variable bit rate (VBR)-coded video is more popular than constant bit rate (CBR)-coded video due to its statistical multiplexing gain and stable quality level. This phenomenon forms a real challenge for network planning and especially for appropriate capacity evaluation. In this work, we propose a new capacity planning model that is especially designed for VBR video transmission over IP networks. This new method takes the quality level and the desired delay threshold as its main design parameters. We investigate this new method by means of simulations and we finally validate its applicability for real-time video transmission with respect to different system parameters.

Key Words

Real-time traffic, IP network planning, capacity assignment, quality of service

1. Introduction

Video communication services are a major capacity consuming applications for IP networks. Variable bit rate (VBR)-coded video is preferred over constant bit rate (CBR)-coded video due to its advantages in statistical multiplexing gain and consistent video quality. Though, VBR traffic causes a serious challenge for network planners to assess the best tradeoff capacity share required to achieve high performance and low costs. In this work, we introduce a dimensioning model based on the waiting time distribution of video frames and a desired outage probability that defines the extent to which the given delay threshold is adhered to. The concept of evaluating the bandwidth needs of stream-type traffic such as voice and video has been widely investigated in the literature where an effective bit rate is calculated based on the traffic characteristics and requirements. Several authors such as [1–3] have applied this concept in the context of ATM call admission control. Effective rates have also been evaluated in the context of IP network dimensioning. For example, the authors in [4] propose a traffic shaping algorithm for effective rate enforcement in IP networks, whereas the authors in [5] evaluate the effective rate per traffic flow using real measurement traces from operational IP networks. The main advantage of our proposed dimensioning method as compared to the existing literature is its simplicity and practicality. We test our method and verify that providing the obtained capacity share to video traffic leads to meeting the delay requirements.

We begin with an overview on moving picture experts group (MPEG)-coded traffic pattern and provide a brief description about a few theoretical video models. In Section 2, we roughly assess the performance of the theoretical video models in modeling real videos so results can be used to drive network simulations. Later on, the capacity assignment strategy or dimensioning model is introduced in Section 3 where several factors affecting this model are handled and their influence is evaluated. Finally, Section 4 wraps up the article.

1.1 Overview on MPEG Video Encoding

MPEG represents a family of standards used for coding audio-visual information in a digital compressed format. It has been a very common standard for video streaming and real-time video applications on the Internet. This is due to its high compression ratios with minimal impact on the perceived quality.

In MPEG, three different types of frames are defined, each of which has its own properties and coding mechanisms. The frame types are named as such: “intra” picture (I-frame), “predictive” picture (P-frame) and “bidirectionally-predictive” picture (B-frame). The I-frame is coded independently of previous or future frames. I-frame coding is based on the frame itself and it is similar to static JPEG image (i.e., intra-frame coding). The P-frame is coded based on the forward prediction that is performed with respect to the past I-frame or P-frame. P-frame coding uses motion compensation prediction to provide more compression. Finally, the B-frame is coded based on the forward prediction that is performed with respect to the past I-frame or P-frame. P-frame coding uses motion compensation prediction to provide more compression. Figure 1 summarizes the different frames and their dependencies. Sequences of MPEG video comprise group of pictures (GOP) where each GOP...
comprises video frames of the three different types. GOPs usually occur in a periodic fashion in between two I-frames. A GOP commonly consists of 12 frames interleaved in the following sequence: “BBPBBBPBBPB.”

MPEG-coded videos have either a CBR or a VBR. With CBR compression, the quantization scale is modified so as to achieve a fixed rate leading to quality degradation in high-motion scenes and waste of bandwidth in low-motion scenes. With VBR compression, on the other hand, more bandwidth is allocated to scenes that are hard to compress; in consequence, clear distinction is made between successive scenes of fast-moving high-action videos. In general, VBR encoding provides superior video quality with shorter compression delays. A better channel allocation is obtained in VBR video transmission as compared to CBR video transmission. As a result, VBR encoding is more commonly used in IP networks. However, due to VBR transmission, more complexity and higher challenges arise in determining the amount of capacity resources required for a successful transmission of video streams.

In this work, we use real videos that are MPEG coded to formulate our dimensioning model but then validate our results with MPEG-independent video models as well.

1.2 Theoretical Video Models

Transport of compressed video is expected to pervade computer networks in the near future. Video is commonly encoded in VBR traffic to improve video quality and reduce encoding delays and yet to make efficient use of the available capacity using statistical multiplexing. Statistical multiplexing leads to variable buffering delays and losses, which negatively affect the video quality. From the perspective of a network planner, it is important to assess the impact of video traffic on the given network. To this end, various statistical source models are developed to ease the performance analysis by evaluating QoS metrics such as packet loss, delay and jitter. A survey on a number of used VBR source models is provided in [6]. It is not our intention to provide an exhaustive study of the available source models but only an overview of the models suitable for our purpose.

Source models available in the literature can be classified into two main categories, namely Markov-based models and self-similar models. The former category has an advantage of a lower computational complexity as compared to the latter, but requires more parameters (the coefficients of the Markov-chain). The autocorrelation function of Markovian models matches pretty well with that of real video sources in the short range, the fact that makes them short-range dependence models (SRD).

As to the latter category comprising the self-similar models, few parameters are required with a higher complexity in generating video samples. The fractional autoregressive integrated moving average (F-ARIMA) is an example of the self-similar models and can be used to generate traffic whose autocorrelation function can match any kind of desired autocorrelation functions. Self-similar models are distinguished with their long-range dependence (LRD) as opposed to Markov-based models.

Whether SRD or LRD is the most relevant for network resource dimensioning has been an ongoing debate. Efforts have been put in developing a new model that offers both SRD and LRD features [7]. This new model is a hybrid model sharing common properties with both Markovian models and self-similar models. It is based on the so-called \(M/G/\infty\) input process that has been shown as more adequate in modelling video sources than DAR(1) (Markovian model) and F-ARIMA (self-similar model) for example. A rather comprehensive study on the issue of modelling video traffic is available in [8]. In this work, we focus on the \(M/G/\infty\) model and compare results with the F-ARIMA model.

1.2.1 F-ARIMA Model

The F-ARIMA model is developed and analysed in [9–11]. This model requires a set of at least seven parameters to generate a trace file. The parameters are:

- the seed for the internal random number generator,
- the Hurst parameter of the sequence to be generated (it characterizes the sequence self-similarity),
- the ratio \(M_X/M_{GOP}\) where \(M_X\) is the average size of the frames of type \(X\) whether I-, P-, or B-frame and \(M_{GOP}\) is the average size of a GOP,
- the type of marginal distribution of the GOP size that can be any of the following: exponential, beta, gamma, lognormal, Pareto, uniform, and Weibull,
- the parameters required to characterize the marginal distribution of the GOP size.

According to [7], the frame size of an MPEG video follows a gamma distribution at the main region while it is better represented by a Pareto distribution at the tail region. For network dimensioning purposes, special interest lies in the tail distribution the fact that makes us consider the Pareto distribution in our work to generate video traffic traces. The F-ARIMA model generates sets of GOPs whose sizes follow the selected distribution. Within each GOP, the model applies the given ratios of item 3 in the above list to derive the individual sizes of the frames constituting each GOP. These ratios are kept constant throughout the generated distribution.

1.2.2 \(M/G/\infty\) Model

\(M/G/\infty\) has been introduced in [7] as a compromise between Markovian and LRD models. It is shown in the
same work as the only model capable of consistently providing close predictions to the actual queuing performance. The distinctive feature of this model is that it captures the frame size distribution of real video sequences. The main part of the frame size distribution is modelled as a gamma distribution and the tail part as a Pareto distribution. Therefore, the required parameters are those for the gamma and Pareto distribution parts as well as the transition point between the two distributions.

\( M/G/\infty \) can be defined with the following seven parameters:

- The seed for the internal random number generator.
- The Hurst parameter of the sequence to be generated.
- The mean and standard deviation of the sequence.
- The transition point \( (x^*) \) between the gamma and Pareto distribution parts of the video sequence.
- The scale \( (a) \) and shape \( (\alpha) \) parameters for the Pareto distribution part.

The overall frame size distribution of the \( M/G/\infty \) model should be smooth especially at the transition points at which the nature of the distribution differs. If continuity and smoothness at the transition point are ensured, the scale and shape parameters for the Pareto distribution part are not necessary anymore.

2. Video Sequences: Analysis and Insights

2.1 Real Video Samples

For our investigations, we use real video traces with different properties in terms of motion, frame size, and quality. These video traces are taken from [12, 13] and they include movies, cartoons, TV sequences, and videoconferences. Each of these videos is MPEG-4 encoded in three different qualities: low, medium, and high.

In the simulations, we use the high-quality version of the videos where a large mean bit rate range is offered starting from 0.2 Mbps to 1.1 Mbps. Among the available video sequences, we extract two videos on which detailed analysis is based while we use all the available sequences for performing the simulations and validating the obtained results. The two videos selected for the particular analysis are Soccer and Teaching. Soccer is a high-motion video with a high mean bit rate and homogenous frames sizes even for I- and B-frame types. Teaching, on the other hand, is a very low-motion video with a low mean bit rate and largely varying frame sizes.

2.2 Comparison of Theoretical and Real Video Models

F-ARIMA and \( M/G/\infty \) models are implemented and run to simulate video traces with realistic video characteristics. As an example, we configure both theoretical models to fit the Soccer video and compare the resulting traffic rate distributions. We note that the traffic rate distribution is identical in behaviour to that of the frame size except for a scaling factor: one frame is deterministically generated every 40 ms.

2.2.1 Frame Size Distribution

The F-ARIMA model is simulated to generate video traces and tuned to match the Soccer video flow.

As mentioned previously, the F-ARIMA model uses either of seven possible marginal distributions among which are the gamma and Pareto distributions. All possible distributions are nonetheless tested, however, poor results are obtained in matching the resulting distribution with the real video except for the gamma and Pareto models. In Fig. 2, we demonstrate the fitting capability of the F-ARIMA model to the Soccer video once using the Pareto distribution and another time the gamma distribution.

![Figure 2. Fitting capability of F-ARIMA model to real Soccer video.](image)

When comparing the different curves in Fig. 2, we note that both models achieved good fitting results to the real video. In [7], authors show that in general the gamma model performs better for the main part of the distribution while the Pareto performs better for the tail part. For link dimensioning purposes, we select the Pareto distribution as the more appropriate model in configuring the F-ARIMA model due to its good matching capability to the tail distribution of real videos.

Now, we aim to match the \( M/G/\infty \) model to Soccer video traffic distribution. To do so, we determine the transition point between the gamma and Pareto parts and adapt both distributions to fit the main and the tail parts of the real video distribution, respectively. In Fig. 3, we present the CCDF obtained using the \( M/G/\infty \) model and compare it to that obtained using the F-ARIMA model with Pareto distribution. We conclude that both F-ARIMA and \( M/G/\infty \) are capable of modelling real video traffic; however, \( M/G/\infty \) results in a more accurate fit.

2.2.2 Frame Size Sequence

In addition to the frame size distribution, the frame size sequence has a key role in affecting the required capacity. The frame size sequence influences in fact the frame waiting time. For example, for the same set of frame sizes, the
frame waiting time differs depending on the sequence in which the frames appear on the link. Thus, to evaluate the link capacity that is capable of keeping the frame waiting time within a given threshold, the frame sequence should be analysed.

Figure 4 presents a typical sequence of real video frames belonging to the Soccer video and compares it to the sequence of frames generated using F-ARIMA and $M/G/\infty$ models, which are configured to match the CCDF of Soccer frame size. The typical MPEG pattern with large I-frames followed by two small B- and P-frames is preserved in the F-ARIMA model as manifested in the figure. The sizes of different frame types of the F-ARIMA model have a fixed ratio to the GOP size and thus we can observe that all B- and P-frames inside one GOP have equal sizes. Regarding the $M/G/\infty$ model, the frame sequence is more chaotic and the typical MPEG pattern is not clear. This is due to the fact that the $M/G/\infty$ model does not differentiate between the different frame types even though it generates accurate frame size distribution.

In brief, while the $M/G/\infty$ model can better match the CCDF of frame size, it fails to reproduce the typical “IBBP...” sequence characterizing the MPEG format. F-ARIMA, on the other hand, provides the typical MPEG frame sequence, and hence results in a more realistic frame waiting time. As a result, it is necessary that we consider both models in our study because each captures a different behaviour of real video traffic.

2.3 Video Traffic Modelling and Characteristics

2.3.1 Single Video Flows

For each video sequence, we are interested in the following traffic statistics parameters:

- Mean bit rate, $r$
- Frame-based standard deviation of the traffic rate, $\sigma_{\text{frame}}$
- GOP-based standard deviation of the traffic rate, $\sigma_{\text{GOP}}$
- Frame-based 1-percentile of the traffic rate, $\gamma_{1\%}^{\text{frame}}$
- GOP-based 1-percentile of the traffic rate, $\gamma_{1\%}^{\text{GOP}}$.

With a sufficiently large number of VBR videos, the aggregated traffic rate comes close to a Gaussian distribution [14] as will be demonstrated later in this section. As the tail behaviour of the traffic rate distribution is particularly important for link dimensioning, we consider the 1-percentile traffic rate $\gamma_{1\%}$ rather than the standard deviation for a closer match with the Gaussian distribution tail [15]. When the Gaussian distribution is fitted to the mean and 1-percentile traffic rate, its corresponding standard deviation is computed as follows:

$$\sigma_{1\%} = \frac{\gamma_{1\%} - r}{2.326}$$

Figure 5 illustrates the computation of the 1-percentile traffic rate and the corresponding standard deviation of the Gaussian distribution. In this figure, three distributions...
are plotted: the traffic rate distribution of the Soccer video, the Gaussian distribution fitted to \( r \) and \( \sigma_{\text{frame}} \), and the Gaussian distribution fitted to \( r \) and \( \sigma_{1\% \, \text{frame}} \). A similar way is performed on a GOP basis. We note that for a single video flow, the traffic rate is close to a normal distribution. This is due to the fact that the Soccer video is high in motion and the frame sizes are largely varying leading to a normal distribution.

In Table 1, we present the traffic statistics of the Soccer and Teaching videos.

<table>
<thead>
<tr>
<th>Video Trace</th>
<th>( r )</th>
<th>( \sigma_{\text{frame}} )</th>
<th>( \sigma_{\text{GOP}} )</th>
<th>( \sigma_{1% , \text{frame}} )</th>
<th>( \sigma_{1% , \text{GOP}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soccer</td>
<td>0.996</td>
<td>0.407</td>
<td>0.320</td>
<td>2.152</td>
<td>1.856</td>
</tr>
<tr>
<td>Teaching</td>
<td>0.397</td>
<td>0.280</td>
<td>0.168</td>
<td>1.395</td>
<td>0.768</td>
</tr>
</tbody>
</table>

### 2.3.2 Aggregated Video Flows

In a multiservice IP network, traffic flows sharing common characteristics and performance constraints are grouped into one traffic class and treated unanimously. Interactive video traffic forms one traffic class whose capacity requirements are to be evaluated so as to achieve premium quality level as desired.

Aggregating all active video flows into one traffic class, we can treat the total aggregated flow as a single “virtual” video flow and compute the corresponding traffic statistics. As all video flows are assumed as MPEG coded with 25 frame/s, one frame of each flow is expected to appear every 40 ms interval. As a result, the total frames within one 40 ms interval are grouped and considered as one “virtual” frame belonging to the “virtual” flow whose size is simply the total sizes of the associated frames. The GOPs of the “virtual” flow are identified in a similar way. The mean bit rate and the standard deviation of the “virtual” video flow process \( Y \) composed of \( K \) individual video processes \( X_i, i = 1 \ldots K \), are simply given by:

\[
\begin{align*}
  r_Y &= \sum_{i=1}^{K} r_{X_i} \\
  \sigma_Y^2 &= E[(Y - r_Y)^2]
\end{align*}
\]

where \( r_Y \) and \( \sigma_Y \) denote the mean bit rate and the standard deviation of the “virtual” flow process \( Y \), and \( r_{X_i} \) denotes the mean bit rate of the flow process \( X_i \). The different variants of the standard deviation (\( \sigma \) and \( \sigma_{1\%} \)) can be computed in a similar way. Generally speaking, the individual video flow processes are considered as uncorrelated and thus the new standard deviation becomes:

\[
\sigma_Y = \sqrt{\sum_{i=1}^{K} \sigma_{X_i}^2} 
\]

where \( \sigma_{X_i} \) is the standard deviation of \( X_i \).

In case of any correlation among the different video flows, the resulting standard deviation of the “virtual” flow increases. In the extreme case when \( Y = K \cdot X \), where \( X_1 = X_2 = \ldots = X_K = X \), the new standard deviation becomes:

\[
\sigma_Y = K \cdot \sigma_X
\]
“×” notation denotes that a number of identical flows are aggregated. We realize the resulting distribution to be as close to Gaussian as it was in the single flow case which indicates that no significant effect has been observed when 100 flows are aggregated as compared to the single flow case. Though the identical flows are aggregated within one full GOP, high-motion sequences that last for more than one GOP stayed to be hung together producing large “virtual” frames and similarly for low-motion sequences. Whereas, in the case of different and uncorrelated video flows, the match with the Gaussian distribution is more accurate as observed in Fig. 6. We can conclude that the Gaussian approximation of an aggregate of video flows is quite applicable in most scenarios even when the active flows are somehow correlated. This conclusion is valid because in one single flow, the frame sizes have generally a high range of variability making the distribution close to Gaussian. As a result, the mean and standard deviation of the aggregated traffic are a proper means of characterizing the traffic.

3. The Capacity Assignment Strategy

We start with a simple dimensioning model where we set the video capacity share equal to the mean bit rate. Doing so, we can serve all video traffic, however, no guarantees are given with regards to the service time that might get too long exceeding the real-time criterion. To reduce the service time to acceptable values, video capacity share should be increased. We consider this increase to be multiples of the standard deviation especially because the PDF of the traffic rate tends to a Gaussian distribution [16]. Therefore:

$$C_{video} = r + m \cdot \sigma$$

whereby $C_{video}$ is the video capacity share, $r$ is the mean bit rate of the video traffic whether a single flow or an aggregate of flows is considered, and $m$ is a positive real number. $\sigma$ can be either the standard deviation of the actual traffic itself or that of the Gaussian distribution fitted to the 1-percentile of the actual traffic. The traffic parameters of (6) can also be computed on a GOP basis ($r$ remains unchanged). Therefore, four variants of $\sigma$ are to be considered, namely: $\sigma_{frame}$, $\sigma_{GOP}$, $\sigma_{frame}^{1%}$, and $\sigma_{GOP}^{1%}$. Generally, we can assume that individual video flows are independent and thus traffic statistics of the aggregate flow can be easily determined if the statistics of the individual flows are known. A similar relation coupling the required capacity with the traffic statistics has also been used in [17] and the consequent works [15, 18].

Referring to (6), the key parameter in the dimensioning model is $m$. Our intention is then to determine the behaviour of $m$ and try to evaluate its range of variability. Once done, we are able to determine the capacity requirements of an aggregate flow on one network link knowing that network delay should be constrained. If done in a deterministic way, huge capacity requirements are needed as illustrated in [19]. Therefore, we need to provide premium quality with slightly softened guarantees to notably save resources. Our criterion in determining the value of $m$ is then:

$$P(W \geq \hat{D}) \leq P_{out}$$

where $W$ is the waiting time, $\hat{D}$ is the delay threshold, and $P_{out}$ is the outage probability that determines the frequency in which the delay threshold is exceeded.

3.1 Single Video Flow

The minimal link capacity that assures the QoS criterion of (7) has to be evaluated. For this purpose, we started with a simple network composed of a video source that generates frames according to the test trace files taken from [12]. These video frames are packetized so they can be sent over an Ethernet link that has an MTU of 1,500 bytes. Packets arrive to a network node and are stored in a buffer waiting their turn where packets are serviced according to an outgoing link with a given capacity. The frame waiting time is obtained by recording the waiting time of the last packet of the frame. Figure 8 plots the CCDF curves corresponding to $P_{out} = 10^{-3}$ and $\hat{D} = 5$ ms, 10 ms, and 40 ms, respectively. For each delay threshold, a different capacity is required to serve the video frames within the given threshold. For a delay constraint of 5 ms, a link capacity of at least 20.32 Mbps is needed. If higher network delays are allowed, link capacity decreases to 10.17 Mbps and 2.54 Mbps for 10 ms and 40 ms delay constraints, respectively.

![Figure 8. CCDF of frame waiting time of Soccer for various delay thresholds.](image)

For further analysis, we select $P_{out} = 10^{-3}$ and $\hat{D} = 40$ ms, and evaluate the required capacity share for serving one video flow within the given constraints. Later in this article, we study the impact of varying the delay threshold on the dimensioning model. Based on the computed capacity value, we determine $m$ for all available video files depending on the type of standard deviation. We also generate random videos using F-ARIMA and


Table 2

Statistics Related to the Value of $m$ using the Different Standard Deviation Variants

<table>
<thead>
<tr>
<th></th>
<th>$\sigma_{\text{frame}}$</th>
<th>$\sigma_{\text{GOP}}$</th>
<th>$\sigma_{1%\text{frame}}$</th>
<th>$\sigma_{1%\text{GOP}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean value of $m$</td>
<td>4.47</td>
<td>10.2</td>
<td>3.10</td>
<td>9.47</td>
</tr>
<tr>
<td>Standard deviation of $m$</td>
<td>0.90</td>
<td>13.31</td>
<td>0.48</td>
<td>16.13</td>
</tr>
</tbody>
</table>

$M/G/\infty$ models and compute the corresponding values of $m$ similarly.

From the obtained values of $m$, we realize that $m$ varies within a certain range depending on the standard deviation variant selected. The value of $m$ remains almost constant with a small range of variation when the frame-based standard deviation variants are used ($\sigma_{\text{frame}}$ and $\sigma_{1\%\text{frame}}$). In Table 2, we compute the mean and the standard deviation values of $m$ to assess its variability behaviour. If $\sigma_{\text{frame}}$ is used, the standard deviation of $m$ does not exceed 0.9 and yet if $\sigma_{1\%\text{frame}}$ is used, it stays below 0.48. This infers that $m$ is independent of the type of video and its motion level, this represents an advantage for network planning. However, if the GOP-based standard deviations are used, $m$ has a wide range of variation depending on the motion level of the videos. The standard deviation of $m$ even exceeds its mean value in some cases. It grows large especially in low-motion videos where very little changes are expected from one scene to the other. This behaviour can be explained by the fact that in low-motion videos, all GOPs have almost the same size where a large I-frame is always followed by very small B- and P-frames. Having this in mind, the standard deviation of the total GOP size gets to shrink. To compensate for this effect, $m$ grows large enough to result in an appropriate capacity share that is capable of serving all traffic within the desired time limits.

3.2 Aggregated Video Flows

In this section, we discuss the most practical case that assumes different videos aggregated asynchronously in a way where I-frames are grouped together, as are P and B frames. Other possible but unlikely cases are handled in [20] such as identical flows aggregated synchronously and asynchronously and different flows aggregated synchronously.

Different videos started randomly within one GOP generate a “virtual” flow with smoother traffic. While aggregating a number of flows, statistical multiplexing comes into play and capacity is saved. In Figure 9(a), we plot the value of $m$ in regards to the number of videos. We show that $m$ slightly decreases and fluctuates within a fixed interval between 1 and 2 when the number of videos exceeds 5. For a small number of aggregated flows, traffic characteristics of the final “virtual” flow are highly dependent on each of the individual flows. Thus, it is recommended to apply the results obtained for a relatively high number of videos, where the statistical multiplexing effect takes place and results get decoupled from the characteristics of individual flows. In the figure, the “Real Videos” and the “$M/G/\infty$” curves are very close highlighting the fact that the latter model is applicable for simulating real video traffic. Therefore, while planning a given network, $M/G/\infty$ model can be used in generating example flows within the network and testing the network readiness. Whereas for the case of F-ARIMA model, the corresponding curve notably falls behind the actual value. Figure 9(b) plots similar curves to the former figure with $\sigma_{1\%\text{frame}}$. We note that similar results are obtained and the same conclusions can be drawn. Interestingly enough, the ratio between GOP-based standard deviations and frame-based standard deviations remains almost constant when the number of flows increases.

Concerning the results of F-ARIMA videos, high variation in $m$ is obtained. For a large number of video files, $m$ approaches 0 when any of the standard deviation variants is used. For example for 100 video flows, $m$ is computed as 0.02 if $\sigma_{\text{frame}}$ is used and 0.01 if $\sigma_{1\%\text{frame}}$ is used. Thus, the required capacity share is almost equal to the mean bit rate of the traffic [refer to (6)]. This behaviour has occurred due to the very systematic way in the F-ARIMA-created MPEG pattern that shows complementarity effect among the flows. The aggregate flow of 100 F-ARIMA videos has $\sigma_{\text{frame}} = 39.62$ and $\sigma_{\text{GOP}} = 0.85$. The difference...
between both deviations is huge, indicating that the “virtual” GOPs have almost equal sizes leading to very small standard deviation while the “virtual” frames maintain their variability among each other leading to high standard deviation.

### 3.3 Choice of the Standard Deviation Variant

In this section, we intend to investigate each of the standard deviation variants and decide upon one of them. Obviously, frame-based standard deviations provide more granular information than that of GOP-based deviations. For a low-motion video, the ratio \( \sigma_{\text{frame}} / r \), which represents the coefficient of variation, increases because B- and P-frames get smaller in size relative to I-frames due to the fact that low-motion videos are characterized by little changes from one scene to the other and so the difference in information between the scenes is minor leading to small B- and P-frames. At the same time, the ratio \( \sigma_{\text{GOP}} / r \) is almost constant yet it even decreases in some cases with respect to the number of videos. The ratio \( C_{\text{video}} / r \) increases for a low-motion sequence, due to large I-frames again that cause high I-frame rate with respect to the mean bit rate and consequently high capacity should be provided to absorb the resulting bursts.

In few words, for low-motion videos, \( C_{\text{video}} / r \) increases, \( \sigma_{\text{frame}} / r \) increases, and \( \sigma_{\text{GOP}} / r \) stays constant or even decreases. Consequently, \( m(\sigma_{\text{frame}}) \) remains unchanged while \( m(\sigma_{\text{GOP}}) \) increases. We consider for example Parkplatz video that represents a video sequence of a surveillance camera in a car park, \( m(\sigma_{\text{GOP}}) \) is computed to be around 72 while \( m(\sigma_{\text{frame}}) \) keeps its value of 3. For high-motion sequences, the different frame types have comparable sizes due to the fact that the difference in information between one scene and the other is large and hence B- and P-frames encode rather lots of information causing high frame sizes. Therefore, frame-based and GOP-based standard deviations lead to similar values of \( m \).

Based on the previous argument, we can conclude that frame-based standard deviations are more adapted for network dimensioning than GOP-based standard deviations due to their motion-independence feature especially for single flows. Section 3.1 illustrates that \( m \) maintains an almost constant value for nearly all types of videos whether \( \sigma_{\text{frame}} \) or \( \sigma_{\text{GOP}} \) is used.

At this point, we are interested in finding out whether \( \sigma \) or \( \sigma_{1\%} \) is more appropriate for our purpose. In fact, for dimensioning purposes, the tail distributions of the waiting time represent our region of interest because we aim to account for most cases except for these few cases that are covered by the tail distribution. If the tail distribution was inaccurately determined then we might discard more packets than intended (allow more packets to experience high delays) thus leading to higher outage probability and vice versa. As a result, it is very important to provide accurate estimation of the tail distribution. Clearly, using \( \sigma \) provides better match to the main part of the traffic rate PDF than \( \sigma_{1\%} \), while using \( \sigma_{1\%} \) provides a better match at the tail distribution of the traffic rate PDF. Frames forming the tail distribution of the traffic rate PDF are the largest frames and, thus, they contribute to the highest frame waiting time. In the simulation results as well, we note that the variability of \( m(\sigma_{1\%}) \) is the least among others whether a single flow is considered or an aggregate of flows. Finally, we can decide for \( \sigma_{1\%} \) as the most appropriate standard deviation variant to be used for network dimensioning.

### 3.4 Choice of Theoretical Video Models

F-ARIMA-generated videos provide a typical MPEG pattern composed of I-, B-, and P-frames. The frame waiting time within network nodes is highly dependent on the frame sequence rather than the frame size distribution. When F-ARIMA-generated videos are aggregated synchronously, meaning that all I-frames are grouped together, B-frames and so is P-frames, the frame sequence of the “virtual” flow is highly dependent on the individual sequences and thus the resulting waiting time distribution is also dependent on the individual sequences. As \( M/G/\infty \) does not produce the MPEG typical pattern, it performs poorly in the case of synchronous aggregation. As a result, the F-ARIMA model is the more suitable in the synchronous aggregation case though the \( M/G/\infty \) generates more realistic frame size distributions which can perfectly fit with real video frame size distributions.

On the contrary, when a number of different videos are aggregated randomly within one GOP duration, the different frame types of the different videos mix together and thus the “virtual” I-frame of the resulting aggregate flow can be comprised of I-, B-, and P-frames. This causes rather smoothed traffic where the initial traffic sequence of the individual flows has slight impact on the final traffic sequence of the aggregate flow. In conclusion, the \( M/G/\infty \) model is the more suitable in this case. In practise, video flows are randomly aggregated and so it is advised to use the \( M/G/\infty \) model unless otherwise specified (i.e., synchronous aggregation).

### 3.5 MPEG Adaptation for Real-Time Transmission

The frame-type sequence of MPEG-coded videos comprising I-, B-, and P-frames is not well adapted to real-time transmission due to the high coding and decoding delays required. The presence of B-frames imposes extra delay to the coding process due to the fact that B-frames make use of forward and backward prediction. For this reason, MPEG coding without B-frames is needed. In regards to \( M/G/\infty \), the model generates frame sizes with a realistic distribution but no special pattern of the frames as in MPEG-coded sequences is taken into account. \( M/G/\infty \) has been shown earlier to produce satisfactory results which closely match those of real videos in case of random aggregation of different flows. This implies that our results are independent of the MPEG pattern whether a B-frame is used or not though the real videos considered comprise B-frames.

In the modified pattern of MPEG coding, the coder encodes P-frames, which are typically one-quarter of an I-frame, instead of coding B-frames, which are generally
one-eighth of an I-frame. However, we recall that network delay is mostly influenced by the largest frames because they cause the longest frame waiting time which represents the main interest in the context of network provisioning. As a result, the replacement of B-frames with P-frames helps in notably reducing the coding delay and has very slight effect on the frame waiting time. This makes our results still applicable to the case of MPEG coding without B-frames.

To verify this conclusion related to the applicability of our results to the modified MPEG coding, we carry out new simulations. Among the available real videos, none is coded without B-frames. So, we use the F-ARIMA model to generate sequences without B-frames, send them to a network node, and record the frame waiting time when the capacity share allocated to video traffic is set according to our dimensioning model. The value of $m$ is set to the value found out earlier (i.e., $m = 3$), the outage probability to $10^{-3}$, and the delay threshold to 40 ms assuming that the flow crosses one network node.

In Table 3, we present the results obtained and check whether the QoS criterion is met. In the given examples, the provided link capacity was enough to serve the available traffic within the given 40 ms delay limit. Although for the case of different flows aggregated randomly, the $M/G/\infty$ model generates closer results to actual flows than the F-ARIMA model with modified MPEG coding, these examples give us insights about the actual performance of real videos.

### 3.6 Video Coding Quality Level

Videos can be coded with different quality levels. To study whether the quality level affects the dimensioning model, we tested the model with MPEG-coded video having low, medium, and high quality levels. It is noted in [13] that the variability of the frame sizes increases if the encoding quality decreases. In fact, for lower quality level, a higher compression ratio can be obtained for most of the frames if some details of the picture are ignored and not encoded. However, the changes from one scene to the other should still be encoded. This makes traffic rate more variable at the frame level within one GOP. Consequently, the ratio of the frame-based standard deviation to the mean bit rate increases. GOP size variability decreases because all GOPs have almost equal sizes making GOP-based standard deviations less influenced by the coding quality level. Table 4 presents the different statistics parameters for an example video, Aladdin. If high-quality coding is used in this example, $\sigma_{\text{frame}}/r = 0.88$ and if low-quality coding is used, $\sigma_{\text{frame}}/r = 1.39$ (the ratio increases for lower quality). On the contrary, $\sigma_{\text{GOP}}/r = 0.59$ for high-quality videos and 0.54 for low-quality videos. Hence, the coding quality level has a slight influence on $\sigma_{\text{GOP}}/r$ ratio.

As to the ratio of $C_{\text{video}}$ to the mean bit rate, it increases as so as to abide by the QoS criterion in case of bursty traffic which is resulted due to the increased variability in frame sizes for reduced quality level. For example, $C_{\text{video}}/r = 7.5$ for high-quality coding, 9.6 for medium-quality coding, and 10.2 for low-quality coding. If $C_{\text{video}}/r$ and $\sigma_{\text{frame}}/r$ increase for lower quality coding, $m$ then remains almost constant if frame-based deviations are used and particularly if $\sigma_{\text{frame}}^{1\%}$ is used where $m$ evaluates to an approximate value of 4 as demonstrated in Table 5. As mentioned earlier, $\sigma_{\text{GOP}}/r$ remains unchanged causing $m$ to increase to compensate for the increase in the required capacity share. If GOP-based standard deviations are used, $m$ has a wide range of variability. This is shown in Table 5 where $m(\sigma_{\text{GOP}})$ varies from 9.91 to 22.99 and $m(\sigma_{\text{GOP}}^{1\%})$ from 7.12 to 17.20.

From Table 5, we realize that $m(\sigma_{\text{frame}}^{1\%})$ is the most appropriate variant as it is the least one affected by the quality level. As a result, $m(\sigma_{\text{frame}}^{1\%})$ can be applied directly for link dimensioning without considering the quality level used.

### Table 3
MPEG-Coded Videos without B-frames Tested on a Network Link Dimensioned according to (6): $m = 3$, $\hat{D} = 40$ ms, and $P_{\text{out}} = 10^{-3}$

<table>
<thead>
<tr>
<th>Video Trace</th>
<th>$m$</th>
<th>$\sigma_{\text{frame}}^{1%}$</th>
<th>$m(\sigma_{\text{frame}}^{1%})$</th>
<th>$C_{\text{video}}$</th>
<th>$P(W \geq 40 \text{ ms}) \leq 10^{-3}$?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2 \times$ F-ARIMA</td>
<td>1.508</td>
<td>0.853</td>
<td>3</td>
<td>4.07</td>
<td>$\checkmark$</td>
</tr>
<tr>
<td>$10 \times$ F-ARIMA</td>
<td>7.451</td>
<td>3.297</td>
<td>3</td>
<td>17.34</td>
<td>$\checkmark$</td>
</tr>
<tr>
<td>$100 \times$ F-ARIMA</td>
<td>42.058</td>
<td>3.406</td>
<td>3</td>
<td>52.27</td>
<td>$\checkmark$</td>
</tr>
<tr>
<td>2 F-ARIMA</td>
<td>1.583</td>
<td>0.961</td>
<td>3</td>
<td>4.47</td>
<td>$\checkmark$</td>
</tr>
<tr>
<td>10 F-ARIMA</td>
<td>11.778</td>
<td>1.479</td>
<td>2</td>
<td>14.74</td>
<td>$\checkmark$</td>
</tr>
<tr>
<td>100 F-ARIMA</td>
<td>107.39</td>
<td>3.985</td>
<td>2</td>
<td>115.36</td>
<td>$\checkmark$</td>
</tr>
</tbody>
</table>

### Table 4
Traffic Characteristics of Aladdin Video with Different Quality Levels

<table>
<thead>
<tr>
<th>Aladdin Video</th>
<th>$r$</th>
<th>$\sigma_{\text{frame}}$</th>
<th>$\sigma_{\text{GOP}}$</th>
<th>$\sigma_{\text{frame}}^{1%}$</th>
<th>$\sigma_{\text{GOP}}^{1%}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low quality</td>
<td>0.059</td>
<td>0.082</td>
<td>0.032</td>
<td>0.429</td>
<td>0.159</td>
</tr>
<tr>
<td>Medium quality</td>
<td>0.083</td>
<td>0.090</td>
<td>0.052</td>
<td>0.438</td>
<td>0.251</td>
</tr>
<tr>
<td>High quality</td>
<td>0.241</td>
<td>0.213</td>
<td>0.143</td>
<td>1.033</td>
<td>0.627</td>
</tr>
</tbody>
</table>
Table 5

Computed Values of $C_{\text{video}}$ and $m$ for Aladdin Video with Different Quality Levels: $P_{\text{out}} = 10^{-3}$ and $\hat{D} = 40$ ms

<table>
<thead>
<tr>
<th>Aladdin Video</th>
<th>$C_{\text{video}}$</th>
<th>$m(\sigma_{\text{frame}})$</th>
<th>$m(\sigma_{\text{GOP}})$</th>
<th>$m(\sigma_{\text{1% frame}})$</th>
<th>$m(\sigma_{\text{1% GOP}})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low quality</td>
<td>0.60</td>
<td>9.02</td>
<td>22.99</td>
<td>4.64</td>
<td>17.20</td>
</tr>
<tr>
<td>Medium quality</td>
<td>0.79</td>
<td>5.73</td>
<td>9.91</td>
<td>3.38</td>
<td>7.12</td>
</tr>
<tr>
<td>High quality</td>
<td>1.81</td>
<td>7.37</td>
<td>10.99</td>
<td>4.60</td>
<td>9.43</td>
</tr>
<tr>
<td>Average of $m$</td>
<td>–</td>
<td>7.37</td>
<td>14.63</td>
<td>4.21</td>
<td>11.25</td>
</tr>
<tr>
<td>$\sigma$ of $m$</td>
<td>–</td>
<td>1.34</td>
<td>5.92</td>
<td>0.58</td>
<td>4.31</td>
</tr>
</tbody>
</table>

3.7 Influence of the Delay Threshold

So far, a delay threshold of 40 ms is required for an outage probability of $10^{-3}$. In this section, we investigate the impact of the delay threshold on the obtained results. In some practical scenarios, 40 ms per-hop delay for real-time services might get above the allowed end-to-end delay limit. A real-time flow is expected to cross a number of network nodes to reach its destination where the queuing delay of each node is unpredictable due to the existence of other cross traffic as one cause. Network delay increases with the number of traversed hops and network load. In this section, we test our results for the same outage probability of $10^{-3}$ but with reduced delay threshold values such as 20 ms, 10 ms, and 5 ms so they can apply in a multihop network. These delay thresholds are assumed on a per-hop basis. Figures 10 and 11 plot the values of $m$ obtained with respect to the number of video flows for different per-hop delay thresholds. The former figure assumes identical video flows while the latter assumes different video flows. For the results, we use $m(\sigma_{\text{frame}})$ as recommended earlier.

Similar trend is observed in both figures where the four curves corresponding to the different delay thresholds tend to converge as the number of aggregated flows increases. In random aggregation of flows, the frames belonging to the different videos are distributed within one GOP and it is unlikely that large frames get together. Hence, buffers are unlikely to be overloaded all through the transmission time because “virtual” frames have moderate sizes. As a result, for a large number of flows, the delay threshold has no impact on the value of $m$.

When the required delay threshold is reduced, the link capacity is increased. This imposes a decrease in the total size of the frames present in the buffer at one time. A slight increase in the link capacity is needed to assure half of the initial delay limit; this causes the value of $m$ to converge to a common value for the different thresholds. In Fig. 10, the four curves of the given delay limits seem to converge to the value of 3. This occurs at the point when the total number of identical videos reach 100.

As to Fig. 11, a similar behaviour is realized; however, the curves do not converge completely at 100 flows. When the number of aggregated videos is relatively small, the large I-frames of the videos have a significant influence on the tail of the frame waiting time distribution. This is especially apparent for a single flow case. Even for the two-flow case, we note that the capacity share is doubled if the desired delay threshold is halved. In Table 6, “Soccer + Teaching” aggregate flow requires 2.66 Mbps, 4.98 Mbps, 9.87 Mbps, and 19.75 Mbps if the delay threshold is set to 40 ms, 20 ms, 10 ms, and 5 ms, respectively. This shows that the capacity share is nearly doubled when the threshold is halved. For identical flows aggregation (e.g., 2× Soccer), the same effect is observed.
when the delay threshold is set to 20 ms, 10 ms, and 5 ms where the capacity share is given by 5.17 Mbps, 10.17 Mbps, and 20.35 Mbps. Capacity doubling is not apparent for 2× Soccer when the delay threshold is reduced from 40 ms to 20 ms; this can be explained by the fact that the 40 ms threshold is already long enough and it is equal to the inter-frame duration of a single video, hence this allows two frames only to meet in the buffer. The waiting time can then be halved but the required capacity need not be doubled to serve the traffic within the given limit.

### 4. Conclusions

In this work, we investigated a new dimensioning model for interactive video service using real video files. The model allows for statistical QoS guarantees by defining an outage probability which determines the percentage of frames that are allowed to exceed the desired delay threshold without affecting the desired quality level. We have shown that for VBR videos, unlike CBR video and voice transmission, it is large waste of resources to account for the maximum waiting time in the capacity assignment process so we rather use the frame waiting time. Though the available video files used in this work are MPEG coded, we showed that the obtained results are directly applicable to general forms of video traffic. This is done by using the $M/G/\infty$ model to generate general video patterns and comparing its results to those obtained using MPEG-coded videos. A close match is resulted. We have also studied the way in which the requested delay threshold affects the capacity share needed. Nonetheless, we have shown that the change in the quality level of the coded videos has a slight impact on the achieved results.

### References


Biography

Sanaa Sharafeddine received her B.E. and M.E. degrees in Computer and Communications Engineering from the American University of Beirut (AUB), Lebanon, in 1999 and 2001, respectively. From 2002 to 2005, she was a member of the research staff at the Institute of Communication Networks at Technische Universität München (TUM), Germany. She received her Dr.-Ing. degree in June 2005. In October 2005, she joined the Department of Computer Science and Mathematics at Lebanese American University (LAU) as an assistant professor. Her research interests include IP network planning, quality of service provisioning for real-time services over IP networks, peer-to-peer networking, and pervasive and mobile computing.