A Poisson Hidden Markov Model for Multiview Video Traffic

Lorenzo Rossi, *Member, IEEE*, Jacob Chakareski, Pascal Frossard, *Senior Member, IEEE*, and Stefania Colonnese

***Abstract*—Multiview video has recently emerged as a means to improve user experience in novel multimedia services. We propose a new stochastic model to characterize the traffic generated by a Multiview Video Coding (MVC) variable bit-rate source. To this aim, we resort to a Poisson hidden Markov model (P-HMM), in which the first (hidden) layer represents the evolution of the video activity and the second layer represents the frame sizes of the multiple encoded views. We propose a method for estimating the model parameters in long MVC sequences. We then present extensive numerical simulations assessing the model's ability to produce traffic with realistic characteristics for a general class of MVC sequences. We then extend our framework to network applications where we show that our model is able to accurately describe the sender and receiver buffers behavior in MVC transmission. Finally, we derive a model of user behavior for interactive view selection, which, in conjunction with our traffic model, is able to accurately predict actual network load in interactive multiview services.**

***Index Terms*—Hidden Markov models, multiview video, telecommunication traffic, three-dimensional TV.**

I. Introduction

THE ADVENT of novel video services with multiple views of the same video scene, e.g., 3-D TV or free-view point video, poses many novel challenges in terms of coding, processing, and transmission of the multimedia content. As far as encoding techniques are concerned, the ISO/ITU-T Joint Video Team has recently finalized the H.264 Multiview Video Coding (MVC) standard, which is explicitly devoted to efficient compression of a multiview source [1]. It is expected that multiview video communication services will be traffic-intensive, which raises important questions in network dimensioning. In addition, the encoding dependencies between the different views renders resource allocation quite challenging in an MVC communication system.

Both problems of network dimensioning and resource allocation are usually addressed with the help of traffic models in classical video delivery services. Such tools have proved to be a valid support for efficient and accurate allocation of network resources by characterizing the compressed video content through statistical models. Video traffic models have been derived for different applications in teleconferencing [2], video broadcasting [3], [4], or streaming [5]. Different stochastic models based on autoregressive processes [2], transform expanded sample (TES) processes [6], and hidden Markov models (HMMs) [7] have been considered for network design, resource allocation, buffer dimensioning, and performance evaluation [8].

The type of data and the traffic characteristics [9] are quite different in multiview video services compared to classical video streaming. The traffic observed during a multiview video communication session is generated by dynamically multiplexing different encoded streams, corresponding to the hierarchically organized encoding of different views. The encoded video data are richer than in the single-view case. When n encoded frames are extracted for transmission according to the user's view switching pattern, they are selected from an $n \cdot N_{\mathrm{View}}$ encoded frames set, where the number of views may range from a few to a hundred. In turn, a hierarchical prediction encoding structure is superimposed to the $n \cdot N_{\mathrm{View}}$ encoded frames set. This results in statistical constraints on the generated traffic trace, which need to be addressed by new devoted models. For the aforementioned reasons, although there are many single-view video traffic models already available, they cannot be directly adapted to model MVC traffic or lead to poor modeling performance.

In this paper, we propose a new traffic model for MVC content that characterizes the frame size sequence observed at the output of an MVC variable bit rate (VBR) source. Specifically, building upon our preliminary work [10], we design a doubly stochastic source model, namely a Poisson hidden Markov model (P-HMM) [11], in which the first (hidden) layer consists of a nonstationary chain modeling the video activity level and the second layer represents the frame sizes of the different MVC encoded views. Furthermore, we extend the P-HMM parameter estimation algorithm for short observation sequences presented in [11] and adapt it to long sequences such as those encountered in video communication services. We assess the model's performance by extensive numerical simulations on classes of MVC sequences sharing common properties. We apply our model to predict the traffic load generated by two different network services based on a client–server video communication paradigm. In the first scenario, which we name Multiview TV, the server simultaneously streams all the MVC encoded views to the client. Our model is shown to be able to accurately predict the states of the sender and receiver buffers in Multiview TV. In the second scenario, named interactive TV, the client dynamically selects the views during the streaming

Manuscript received January 13, 2013; revised September 07, 2013; accepted January 12, 2014; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Weber. Date of publication February 25, 2014; date of current version April 14, 2015.

L. Rossi and S. Colonnese are with the DIET Department, Sapienza Università di Roma, 00184 Rome, Italy (e-mail: rossi@infocom.uniroma1.it).

J. Chakareski is with the Department of Electrical and Computer Engineering, The University of Alabama, Tuscaloosa, AL 35487 USA.

P. Frossard is with the Signal Processing Laboratory (LTS4), Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2014.2303162

session by means of a feedback channel. Due to the MVC encoding dependencies, the server transmits a composite stream comprising all the encoded data required to correctly decode the selected view. Finally, we introduce an Interactive TV user service request model in order to mimic the sequence of requested views selected by the user. In fact, the traffic generated during the interactive TV session depends both on the MVC encoded video traces and on the user's view selection. We show that the combination of our two models is able to accurately characterize the traffic in interactive multiview applications. To the best of our knowledge, this paper is the first attempt of modeling the end-to-end traffic generated during a general multiview session, including a flexible model of the user's switching behavior. The modeling of MVC traffic has not been deeply studied in the literature, and the recent work in [12] is the only existing work that has studied such traffic by analyzing the GOP correlation structure of 3-D video streams. However, it does not provide a comprehensive end-to-end traffic model contrarily to the framework we propose in this paper.

The main contributions of this paper can be summarized as follows.

- A nonstationary traffic model for VBR MVC sequences is introduced, with the ability to characterize different classes of MVC streams at different encoding settings. The model can predict actual network load in network applications.
- A maximum likelihood (ML) estimation procedure suitable to derive the traffic model parameters in long sequences is derived.
- A user behavior model for interactive view selection is combined with our traffic model to characterize interactive multiview traffic.

The rest of the paper is organized as follows. In Section II, we introduce the P-HMM; we also describe the P-HMM parameter set estimation procedure. In Section III, we validate the model in different stream settings. Network applications of our model are studied in Section IV, along with the view switching model. Section V concludes the paper.

II. MVC Source Modeling

A. MVC Coding Format

An MVC stream jointly encodes different video sequences captured by multiple cameras with overlapping fields of view. Let us denote by N_{View} the number of such sequences. One view, denoted as reference view, is independently encoded using temporal motion compensation and transform coding techniques, similarly to a classical video sequence encoded with the H.264 encoder [1]. The other $N_{\mathrm{View}} - 1$ views are encoded using inter-view prediction in addition to temporal prediction, in order to further improve the compression performance. In H.264 MVC [1], inter-view prediction is allowed between frames referring to the same time instant, whereas intra-view encoding dependencies are usually set to permit temporal scalability [13]. The encoding dependencies give rise to a generalized GOP structure of duration N_{GOP}, comprising $N_f = N_{\mathrm{View}} \times N_{\mathrm{GOP}}$ frames. Figs. 1 and 2 show two examples of such MVC GOPs. Given the complete MVC encoded bitstream, up to N_{View} flows are transmitted and decoded by the client. In most applications, all the views are transmitted together in a simulcast mode.

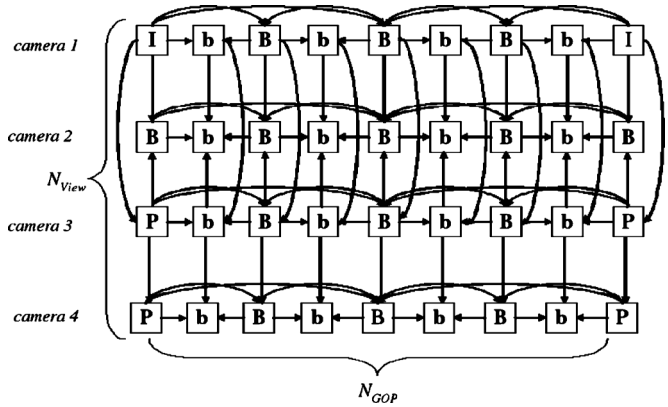


Fig. 1. GOP structure and encoding hierarchy ($N_{\mathrm{GOP}} = 8$).

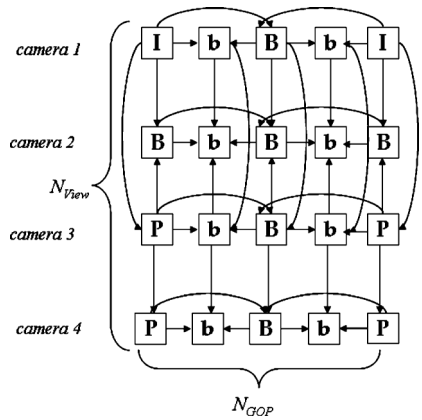


Fig. 2. GOP structure and encoding hierarchy ($N_{\mathrm{GOP}} = 4$).

In order to control the size of the bitstreams, different rate control algorithms [14] can be implemented in MVC encoders. We however focus in this paper on VBR streams, where the quantization step sizes are fixed. Their value only depends on the frame type, as commonly employed in most MVC applications [1]. H.264/MVC is the current standard for multiview video encoding standard, but our work can be applied to any encoding technique based on a similar framework with motion-compensation, such as the recent HEVC [15].

B. Traffic Model

We propose now a new model that is able to characterize the frame sizes in GOPs of an MVC compressed stream with a given GOP structure. In VBR operating mode, the bit rate of the MVC encoded views varies according to the video activity level, and the traffic model should match this nonstationary stochastic process. In fact, the video activity level changes unpredictably according to the scene content (e.g., indoor/outdoor scenes, moving objects, etc.), and we cannot assume any *a priori* relation between the duration of the current activity level and that of the previous levels. To represent this behavior, we resort to the class of hidden semi-Markov models (HSMMs) [16]–[18]. HMMs' state duration times are constrained to have a geometric distribution; HSMMs model the state duration by a general probability distribution, thus encompassing both stationary and nonstationary models. As far as the choice of the distribution is concerned, the work in [19]

analyzes different distributions, namely Gamma, Pareto, and Weibull, for modeling scene duration time of a single-view video stream. According to the results in [19], the Poisson distribution properly trades off compactness for accuracy in the traffic model. Here, we observe that the duration of a given video activity level in a multiview sequence has similar statistical characteristics to the activity duration in a single-view sequence since both are based on the real scene dynamics as represented by the video frames. Thus, we assume a Poissonian state duration distribution.

We build a new P-HMM [11] as a two-layer stochastic process in which the first (hidden) layer is a discrete-time Markov chain whose states represent different video activity levels, and the second layer represents the frame size sequence corresponding to a given activity level. We make the following assumptions about the nonstationarity of the sources.

1) The activity level of the video content varies in time according to a Poisson distribution.
2) The hidden-layer state transitions (i.e., the change of the activity level) occur at the beginning of a GOP.
3) The activity level is the same in all views since views are correlated.

With these simplifying assumptions, we model the duration of a scene of a given activity level, by a simple one parameter distribution. We discard the possible changes in the average frame size due to activity level changes observed in the middle of a GOP. We will show that, in spite of this approximation, the model closely matches the MVC source characteristics.

Let us denote the number of states (i.e., different video activity levels) in our model as N_s. The state duration obeys a Poisson distribution denoted by

$$d_i[k] = \frac{e^{-\lambda_i}\lambda_i^k}{k!}. \tag{1}$$

When a state transition occurs, the model is described by a state transition matrix Π, whose element π_{ij} denotes the probability of transition from state i to state j. Because of the explicit modeling of the state duration time distribution, $\pi_{ii} = 0$ for all i. Finally, π_i denotes the initial probability of the model being in state i.

The second layer in our model describes the frame sizes in an entire GOP. Formally, let us consider the random vector

$$x[n] = \left[x_0[n], \ldots, x_{N_f-1}[n]\right]$$

representing the set of frames sizes in the nth GOP of the compressed multiview content. The vector $x[n]$ is emitted in accordance to a multivariate probability mass function (*pmf*) depending on the actual hidden layer state. Given the current state in the first layer of the model, say i, a random vector $x[n]$ is generated according to the *pmf* $b_i[x[n]]$. For the sake of compactness, each *pmf* $b_i[\cdot]$, $i = 1, \ldots, N_s$, has a different number of bins depending on the coding mode (namely I-, P-, or B-frames) of the compressed picture to be generated. This choice is motivated by not imposing that the frame size *pmf* follows a fixed probability distribution (e.g., Gaussian, Gamma, ...), but instead by adapting the distribution shape to the actual MVC sequences. Moreover, by allowing a different number of bins for different frame types, we can adaptively control the model's complexity (i.e., the parameter set) in order to avoid model overfitting to the training data. Fig. 3 illustrates the hidden-state

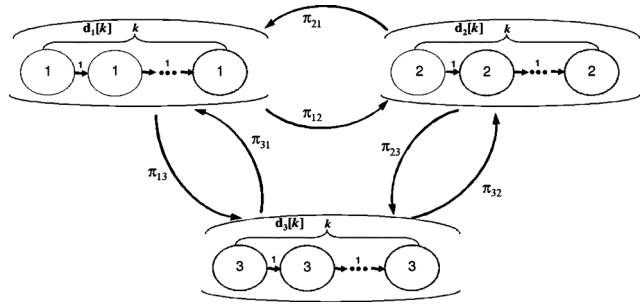


Fig. 3. State diagram for a three-state P-HMM. Enclosed between parentheses, the explicit duration k for every state is represented.

chain for a three-state P-HMM. The hidden state i, $i = 1, 2, 3$, lasts for k instants (expressed by the 1-probability transitions), where k is explicitly extracted according to state i duration distribution $d_i[k]$. Then, the next state $j \neq i$ is selected by means of the transition matrix Π. In every instant, a random vector $x[n]$ is extracted according to $b_i[\cdot]$ like in a typical HMM.

C. Parameter Estimation

The estimation of the model parameters is a crucial step in traffic characterization. Since the model belongs to the wide HMM family [17], we can resort to one of the estimation algorithms employed for such models. In particular, an estimation procedure called Expectation-Maximization (EM) algorithm [20] is widely used for HMMs. A version of the EM algorithm has been proposed for P-HMMs in [11]. However, it exhibits numerical instability when used for long data sequences [16]. We derive here a new EM algorithm for stable parameter estimation in long sequences. Our algorithm extends the method of [16] to the case of nonstationary hidden-state durations. We present the parameter estimation in detail below.

Suppose that we observe a video sequence composed of N GOPs. Let $y_0^{N-1} \stackrel{\text{def}}{=} \{y[n]\}_{n=0}^{N-1}$ denote the training sequence and $\Theta \in \mathbf{\Theta}$ the parameter set of our model, where $\mathbf{\Theta}$ is the parameter space and $\Theta \stackrel{\text{def}}{=} \{\Pi, \lambda_1, b_1[\cdot], \pi_1, \ldots, \lambda_{N_s}, b_{N_s}[\cdot], \pi_{N_s}\}$. The EM algorithm comprises two iterative computational steps. The first one is an expectation step that computes the auxiliary likelihood function $Q(\Theta \mid \Theta^{(m)}) = E\{\log(\text{Prob}\{S, x, \Theta\}) \mid y, \Theta^{(m)}\}$, in which $S \in \mathbf{S}$ represents a plausible state sequence and $\Theta^{(m)}$ is the mth estimate of the parameter set. Then, a maximization step maximizes the likelihood function, i.e.,

$$\Theta^{(m+1)} = \arg\max_{\Theta} Q(\Theta \mid \Theta^{(m)}). \tag{2}$$

The algorithm iterates between the two steps until convergence of the parameter set.

Before getting into more details, let us define a function $\dot{d}_i[k]$ representing the duration distribution for the state i and taking into account the finite length of the observed sequence, as

$$\dot{d}_i[k] = \begin{cases} 1 - D_i[k-1], & \text{if } k = N - n - 1 \\ d_i[k], & \text{otherwise} \end{cases} \tag{3}$$

where $d_i[k]$ has been defined in (1), and $D_i[k]$ is the cumulative distribution of the state duration times. Note that $D_i[k]$ is zero for $k < 0$. Then, the computations in our EM algorithm, as applied to P-HMMs, comprise: 1) the computation of forward probabilities; 2) the computation of backward probabilities; 3) the estimation of the parameter set. The first two steps

Algorithm 1: Computing forward probabilities $\alpha_n(i,k)$ and $\alpha_n(i)$

1: **for** $n = 0 \rightarrow N-1$ **do**
2: **for** $k = 0 \rightarrow N-n-1$ **do**
3: **if** $n = 0$ **then**
4: $\alpha_0(i,k) \propto \pi_i \dot{d}_i[k] b_i[y[0]]$
5: **else**
6: $\alpha_n(i,k) \propto b_i[y[n]] (\sum_{\substack{j=1 \\ j\neq i}}^{N_s} \alpha_{n-1}(j,0)\pi_{ji}\dot{d}_i[k] + \alpha_{n-1}(i,k+1))$
7: **end if**
8: **end for**
9: $\alpha_n(i) = \sum_k \alpha_n(i,k)$
10: **end for**

calculate three auxiliary variables, namely the conditioned probabilities of the state sequence to the observed sequence, representing the expectation steps of our EM algorithm (see [17] for further details). Then, the parameter set is expressed as function of these auxiliary variables. We first define the following forward probabilities[1] for $n = 0, \ldots, N-1$:

$$\alpha_n(i,k) \stackrel{\text{def}}{=} P(s_n = i, \ldots, s_{n+k} = i, s_{n+k+1} \neq i \mid y_0^n, \Theta^{(m)}), \quad k < N-n-1 \tag{4}$$

$$\alpha_n(i) \stackrel{\text{def}}{=} P(s_n = i \mid y_0^n, \Theta^{(m)}). \tag{5}$$

For $k = N-n-1$, the definitions above are slightly different in order to take into account the finite length of the actual sequence

$$\alpha_n(i, N-n-1) \stackrel{\text{def}}{=} P\left(s_n = i, \ldots, s_{N-1} = i \mid y_0^n, \Theta^{(m)}\right).$$

Those quantities are calculated by the recursive algorithm illustrated in Algorithm 1.[2] They represent the probability for the system to be in state i at time n and to stay in the same state for the next k instants, given the sequence observed until time n.

Then, the following backward probabilities are defined in a similar way

$$\gamma_n(i,k) \stackrel{\text{def}}{=} P(s_n = i, \ldots, s_{n+k} = i, s_{n+k+1} \neq i \mid y_0^{N-1}, \Theta^{(m)}), \quad n = 0, \ldots, N-1;\ k < N-n-1 \tag{6}$$

$$\xi_n(i,j,k) \stackrel{\text{def}}{=} P(s_{n-1} = i,\ s_n = j, \ldots, s_{n+k} = j, s_{n+k+1} \neq j \mid y_0^{N-1}, \Theta^{(m)}), \quad n = 1, \ldots, N-1;\ k < N-n-1. \tag{7}$$

Note that we resort to a different definition for the backward probabilities with respect to the usual β notation [17]. In [17], a β backward probability is defined representing the likelihood of the observed sequence, and γ and ξ are calculated by means of α and β. Here, we calculate directly γ and ξ in a backward iteration in order to avoid numerical issues arising from the sequence length. Algorithm 2 illustrates the backward probabilities computation, where δ_i^j denotes the Kronecker function.

Finally, the parameter set $\Theta^{(m+1)}$ in the maximization step can be calculated with the help of the forward and the backward probabilities above. We can write the initial probability of being in state i as

$$\pi_i = \sum_{k=0}^{N-1} \gamma_0(i,k). \tag{8}$$

Then, we can express the transition probabilities as

$$\pi_{ij} = \frac{\sum_{n=1}^{N-1}\sum_{k=0}^{N-n-1} \xi_n(i,j,k)}{\sum_{\substack{j=1 \\ j\neq i}}^{N_s}\sum_{n=1}^{N-1}\sum_{k=0}^{N-n-1} \xi_n(i,j,k)}. \tag{9}$$

The frame size distribution in each state is given by

$$b_i[x] = \frac{\sum_{n=0}^{N-1}\sum_{k=0}^{N-n-1} \gamma_n(i,k)\delta_x^{y[n]}}{\sum_{n=0}^{N-1}\sum_{k=0}^{N-n-1} \gamma_n(i,k)}. \tag{10}$$

Finally, the state duration is expressed as

$$\lambda_i = \left(\sum_{n=1}^{N-1}\sum_{k=0}^{N-n-2}\sum_{\substack{j=1 \\ j\neq i}}^{N_s} k\ \xi_n(j,i,k) + \sum_{k=0}^{N-1} k\ \gamma_0(i,k) \right) \cdot \left(\sum_{n=1}^{N-1}\sum_{k=0}^{N-n-2}\sum_{\substack{j=1 \\ j\neq i}}^{N_s} \xi_n(j,i,k) + \sum_{k=0}^{N-1} \gamma_0(i,k) \right)^{-1}. \tag{11}$$

Note that a single iteration of the estimation algorithm consists of calculating first the state sequence probabilities (4)–(7), which is the expectation step. Subsequently, we compute the new parameter estimates by averaging the observations weighted with the state probabilities (8)–(11), which corresponds to the maximization step of an iteration of the algorithm. Convergence is assured by Jensen's inequality [16].

Algorithm 2: Stable estimation of backward probabilities $\gamma_n(i,k)$ and $\xi_n(i,j,k)$, from (6) and (7)

1: **for** $n = N-1 \rightarrow 1$ **do**
2: **for** $k = N-n-1 \rightarrow 0$ **do**
3: **if** $n = N-1$ **then**
4: $\gamma_{N-1}(i,k) = \alpha_{N-1}(i)$
5: **else**
6: **if** $k \neq 0$ **then**
7: $\gamma_n(i,k) = \xi_{n+1}(i,i,k-1)$
8: **else**
9: $\gamma_n(i,0) = \sum_{\substack{j=1 \\ j\neq i}}^{N_s}\sum_{k=0}^{N-n-2} \xi_{n+1}(i,j,k)$
10: **end if**
11: **end if**
12: $\xi_n(i,j,k) = \frac{\alpha_{n-1}(i,0)\pi_{ij}\dot{d}_j[k] + \delta_i^j \alpha_{n-1}(i,k+1)}{\sum_{\substack{l=1 \\ l\neq i}}^{N_s} \alpha_{n-1}(l,0)\pi_{lj}\dot{d}_j[k] + \delta_l^j \alpha_{n-1}(l,k+1)} \cdot \gamma_n(j,k)$
13: **end for**
14: **end for**

III. Traffic Model Validation

In this section, we assess our model by comparing statistics evaluated on a pseudo-random synthetic traffic generated according to our P-HMM to the statistics evaluated on a composite MVC test sequence. The P-HMM parameters are estimated by

[1] Our definitions differ from [11] in order to avoid numerical instability.

[2] The normalization coefficient for $\alpha_n(i,k)$ in steps 4 and 6 is calculated by summation over i and k.

TABLE I
REFERENCE TEST SEQUENCES USED TO GENERATE THE COMPOUND SEQUENCE

sequence name	# frames	# views
Akko & Kayo	290	100
Champagne Tower	500	8
Uli	250	8
Jungle	250	8
Balloons	500	7
Kendo	400	7
Dog	300	80
Pantomime	500	80

applying the EM algorithm of Section II-C on the observed composite sequence. We also test the case in which the test sequence is different from the sequence used to train the model in order to show the model's ability to represent not only the training sequence, but also other sequences with similar content. A comparison to a well-known single-view VBR model [21] is also carried out.

A. MVC Encoder Settings

The composite MVC sequences considered in the model assessment are generated by randomly selecting and concatenating the views of several reference tests sequences with very different activity levels, as reported in Table I [22], [23]. All the sequences are in CIF format, with a frame rate of 25 fps. The resulting MVC composite sequence is approximately 6 min long and with $N_{\mathrm{View}} = 4$ views. We encode the sequence using the two different GOP structures reported in Figs. 1 and 2. Both structures exhibit motion compensation dependencies among the views for anchor and non-anchor frames [24]. The large number of dependencies accentuates the difference between MVC traffic and simple aggregations of single-view traffic. The two GOP structures differ in the number of I- and P-frames. The bit-rate variability of sequences encoded using the GOP in Fig. 1 is mainly due to the residuals of the motion compensation, whereas for sequences encoded using the GOP in Fig. 2, the bit-rate variability depends on the large number of intra- or P-frames. For each GOP structure, different MVC bitstreams have been generated by setting the quantization parameter of the reference view to 10 (high quality), 20 (medium quality), or 40 (low quality), and by adjusting the temporal layers quantization parameter accordingly.[3] We use JMVC v7.0 to encode the sequences [25], then we use the different MVC bitstreams to build traffic models.

As we discussed in Section II, the number of bins in the *pmf*s of the model may be different for different frame types in order to have a tradeoff between performance and model complexity. However, increasing the number of parameters may lead to model overfitting. In our tests, we heuristically set the number of bins as shown in Table II(a) and (b) for the GOP structures in Figs. 1 and 2, respectively. The number of bins is fixed for every kind of sequence, and it is selected according to the frame coding mode and the hierachical prediction structure (see Figs. 1 and 2). In Section III-B, we will evaluate the performance of our model with different test and training sequences in order to study overfitting problems and validate the values chosen for our parameter set. The bins are placed in the interval between the minimum and the maximum observed frame size for each frame type.

[3]Specifically, we have set the quantization parameters according to the default settings of JMVC [25].

TABLE II
NUMBER OF BINS IN THE *pmf* FOR EACH FRAME OF THE GOP GIVEN IN DISPLAY ORDER. (a) GOP STRUCTURE IN FIG. 1. (b) GOP STRUCTURE IN FIG. 2

(a)

View #0	50	10	10	10	20	10	10	10
View #1 to #3	30	10	10	10	20	10	10	10

(b)

View #0	50	10	20	10
View #1 to #3	30	10	20	10

In the simulations, we employ a three-state model in order to represent low-, medium-, and high-level activity. Our choice for the number of states is motivated by prior results in single-view sequence traffic modeling [26]. Since the activity levels correspond to the real scene dynamics, in our tests we can retain the same number of levels as in single-view modeling. Specifically, we adopt the most compact representation (three states) that permits to achieve an accurate description of the video traffic.

A first coarse estimation of the model parameters is performed by labeling each GOP of the actual sequence as low, medium or high according to its average frame size. The thresholds are set in order to have the same number of GOPs in the three states. Then, a coarse estimation is obtained by evaluating frame size histograms related to each state. Zero-valued bins are set to a low fixed value, and the histograms are normalized accordingly. The transition probability matrix is initialized with positive random values, and the Poisson mean values are set to 1 for each state. After that, the EM algorithm is performed as described in Section II-C using the coarse estimation as starting point. Estimation ends when the difference between the log likelihoods of the two most recent iterates is smaller than 0.01.

Finally, synthetic traffic is produced by first generating a state sequence according to the model and then producing synthetic traffic for a GOP for each state of the sequence by means of the frame size *pmf*s. The bins are converted to the mean frame size of the interval they represent. The synthetic traffic generation is described in Algorithm 3.

B. Performance Evaluation

We now assess the model's accuracy by first comparing the autocorrelation function (acf) and the Q-Q plot computed on a sequence of frame sizes of the actual MVC encoded sequence (comprising all the views) with the respective statistics evaluated on a pseudo-random traffic sequence generated by the P-HMM with parameters estimated from the corresponding composite sequence. Q-Q plot measures the similarity of two distributions by comparing their quantiles. The closer the Q-Q plot to the bisector of the first and third quadrant, the more similar the two distributions. Figs. 4–6 show the acf and the Q-Q plot for different compressed MVC sequences. It is clear that the statistics evaluated on the synthetic traffic (P-HMM-A) closely follow the statistics of the actual MVC sequences, for every GOP structure and quantization parameter. The close match with the actual data is due to the fact that we do not constrain the frame size *pmf*s to have an explicit probability distribution and that we employ the Poisson distribution for

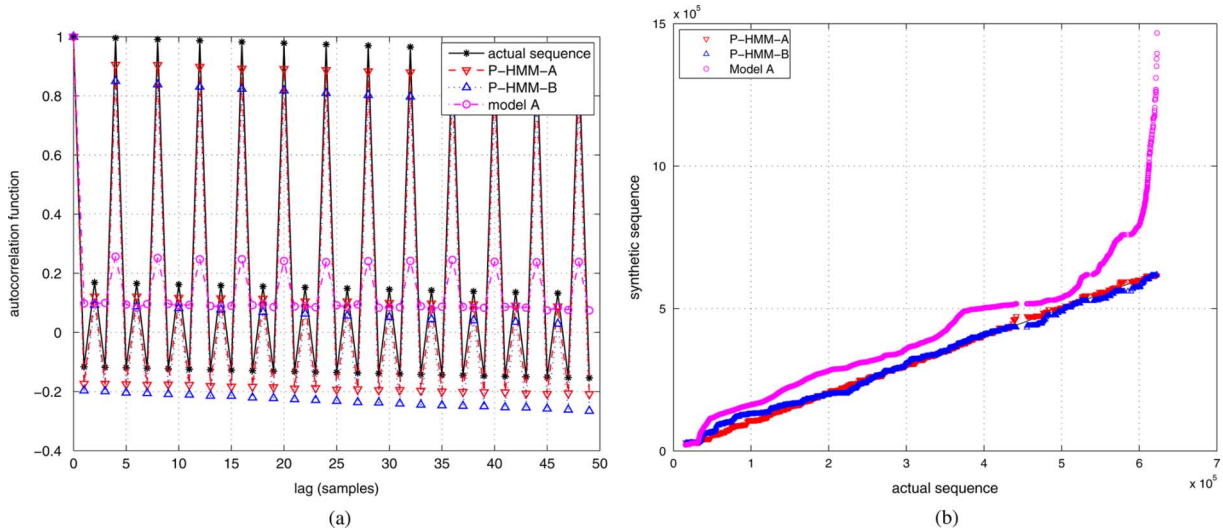


Fig. 4. Comparison of (a) the autocorrelation function and (b) the Q-Q plot estimated on the real MVC encoded sequence and on the synthetic P-HMM generated video sequence (GOP structure in Fig. 2, high-quality stream).

Algorithm 3: Synthetic traffic generation

1: Initial state i extracted according to the distribution $\pi_1, \pi_2, \ldots, \pi_{N_s}$
2: $n \leftarrow 0$
3: **while** $n < N$ **do**
4: k extracted from the Poisson distribution belonging to the current state, i.e., $d_i[k]$
5: **if** $n \geq N - k$ **then**
6: $k \leftarrow N - n - 1$
7: **end if**
8: **for** $j = 0$ to k **do**
9: generate a synthetic GOP $x[n+j]$ according to $b_i[\cdot]$;
10: **end for**
11: $n \leftarrow n + k$
12: perform state transition according to transition matrix Π.
13: **end while**

the state durations, thus forcing a nonstationary recurrence of the activity levels. In order to validate these conclusions, we compare our model to the well-known single-view VBR model described in [21]. Specifically, we focus on the "model A" in [21], whose main differences with our P-HMM model relate to the frame size distribution and the hidden chain describing the activity level. The model A employs three shifted gamma distributions for the sizes of frames (I, P, B) and a stationary seven-state Markov chain for the activity level. Moreover, an *ad hoc* parameter estimation procedure is used in [21]. We see in Figs. 4–6 that model A is not able to describe it correctly. In particular, we can see that model A does not depict accurately the shape of the actual sequence, both for the Q-Q plots and the autocorrelation function. We finally consider the case in which our model is trained with a different sequence with respect to the test sequence; we denote the synthetic traffic generated by this model as P-HMM-B. The purpose of P-HMM-B is to show how our model can approximate not only the training sequence but also other sequences belonging to the same class of MVC sources. The test sequences are generated similarly to the training sequences by a random selection and concatenation of the views from the reference sequences in Table I. P-HMM-A outperforms the other models in mimicking the overall frame size distribution, while P-HMM-B also achieves good adherence performance.

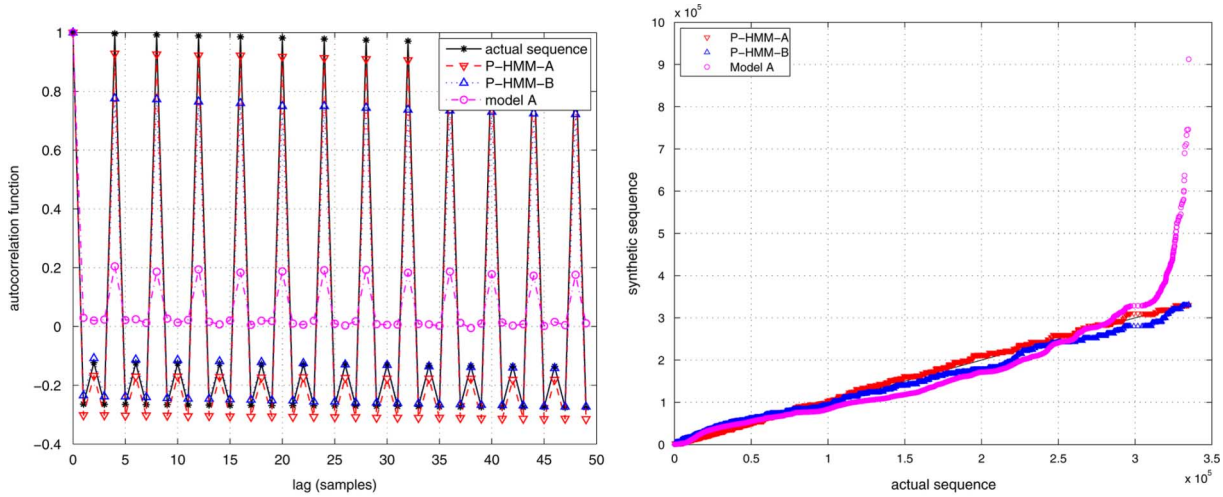
The same statistics have been calculated separately on the traffic related to each view in MVC streams. Figs. 7 and 8 show these statistics for the view #1 and view #3, respectively. Similar results have been obtained for different encoder settings. Again, it is clear that model A [21] is not able to capture the statistics for each single view. Conversely, our models can efficiently characterize the traffic for each view. The characterization of the first- and second-order statistics for every view makes the model attractive to describe real MVC traffic in network applications, where a subset of the views are transmitted to the receivers. The good results obtained with PHMM-B show that the model is able to capture features that are not only related to a single MVC stream, but also to a class of sequences sharing similar content characteristics. Finally, we remark that the slight divergence in terms of autocorrelation function is caused by neglecting the intra-GOP correlation in the model design.

IV. Traffic Model in Multiview Services

A. Applications Scenarios

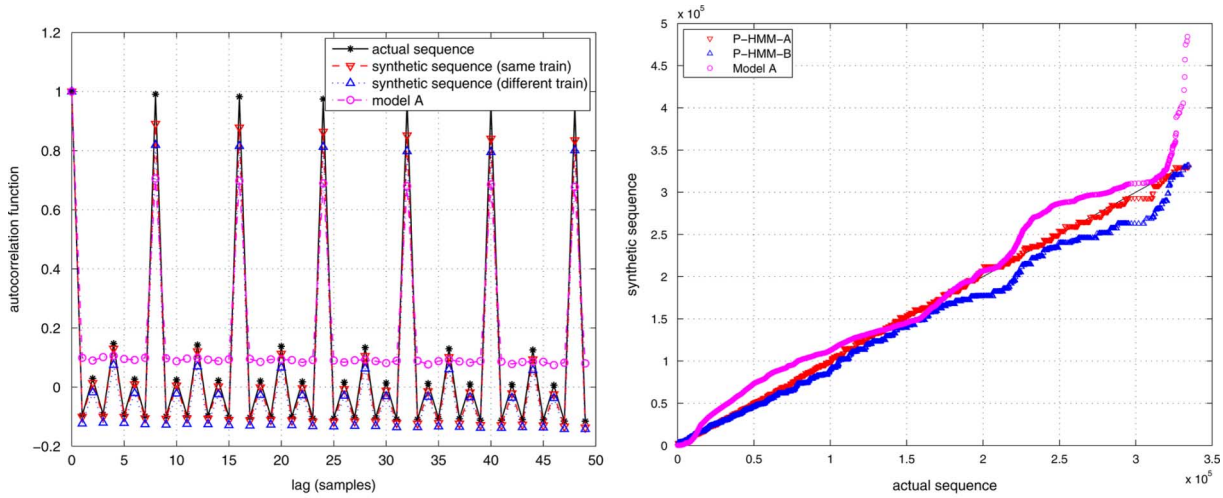
We examine the accuracy of our model in the context of multiview services. We consider two case studies illustrated in Fig. 9. In the first one, called "Multiview TV," the server sends all the MVC content to the user. In the second one, denoted as "Interactive TV," the user requests one view at a time and can switch dynamically among the available views during the playout, exploiting an out-of-band feedback control channel. Due to the coding dependencies, the reference views still have to be transmitted along with the target view in the Interactive TV service.

From the point of view of the MVC source model, the above services differ in that the traffic generated during the Multiview



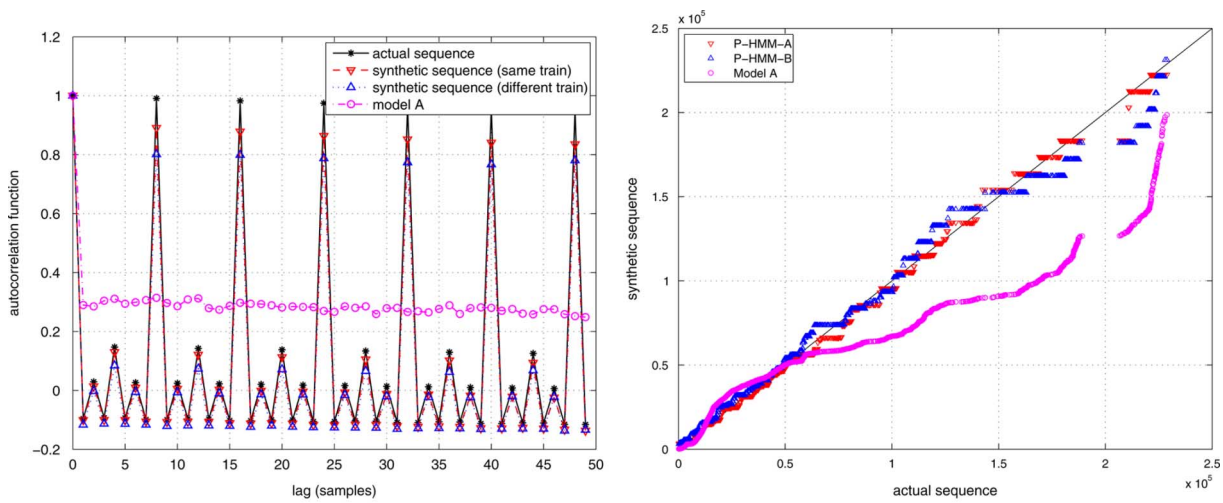
(a) (b)

Fig. 5. Comparison of (a) the autocorrelation function and (b) the Q-Q plot estimated on the real MVC encoded sequence and on the synthetic P-HMM generated video sequence (GOP structure in Fig. 2, medium-quality stream).



(a) (b)

Fig. 6. (a) Autocorrelation function and (b) Q-Q plot estimated on the actual MVC sequence and the synthetic sequences (GOP structure in Fig. 1, medium-quality stream).



(a) (b)

Fig. 7. (a) Autocorrelation function and (b) Q-Q plot estimated on the actual MVC sequence (View #1) and the synthetic sequences (GOP structure in Fig. 1, medium-quality stream).

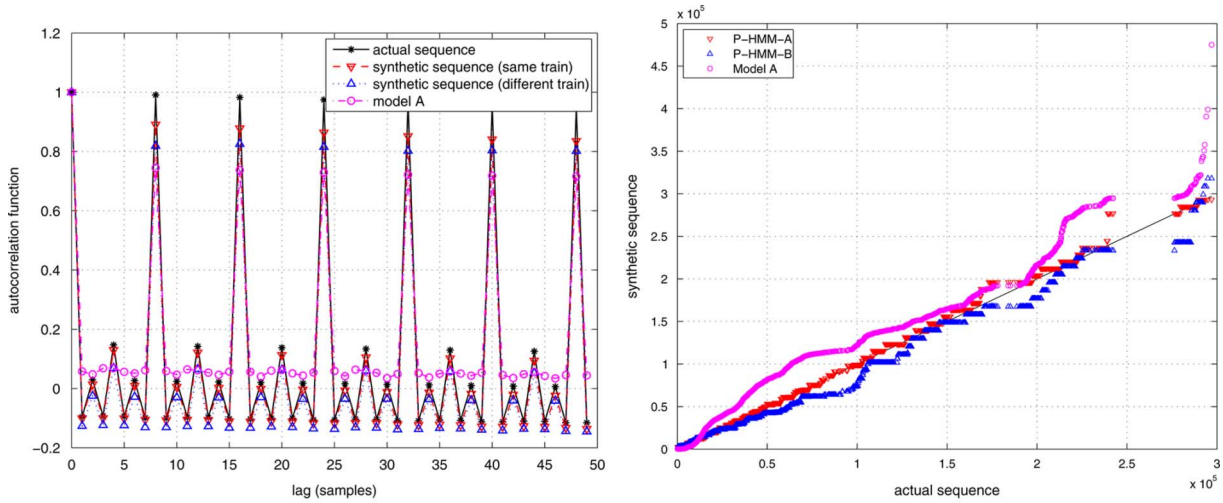


Fig. 8. (a) Autocorrelation function and (b) Q-Q plot estimated on the actual MVC sequence (View #3) and the synthetic sequences (GOP structure in Fig. 1, medium-quality stream).

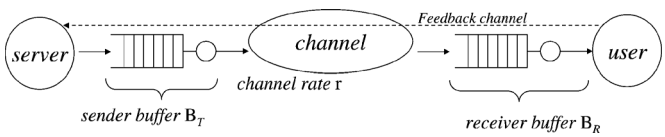


Fig. 9. System description for multiview services. The feedback channel is present only in the interactive TV case.

TV session depends only on the encoded video, whereas the traffic generated during the Interactive TV session depends also on the view selection process. Thus, in order to fully characterize the latter scenario, we develop a new model of the view switching sequence, inspired by related models in the context of channel switching in IPTV systems [27]–[30]. In the design of our view switching model, we employ the following assumptions.

1) A user watches the reference view most of the time.
2) Other views are occasionally selected by the user because they represent added content with respect to the main view (e.g., in a football game, these views may show close-up of the players).
3) A user selects views with preferences that depend on the present view the user is currently watching (first-order dependency).

We model the view switching sequence as a chain in which the states represent different views. The duration of stay in each view is explicitly modeled with a probability distribution. According to the above assumptions, we heuristically select sample values for the view transition probabilities, which are reported in Table III(a). The average and standard deviation of state durations are also set to sample values corresponding to the above assumptions [see Table III(b)]. We have found that the Gamma distribution is suitable and flexible for modeling the state duration time since it takes only nonnegative values with mean and standard deviation that are set independently. Formally, let $d_i^s[t]$ be the density function for the duration time in state i

$$d_i^s[k] \stackrel{\text{def}}{=} \frac{\beta_i^{\alpha_i}}{\Gamma(\alpha_i)} k^{\alpha_i - 1} e^{-\beta_i k} \text{ for } k \geq 0. \tag{12}$$

TABLE III
VIEW SWITCHING MODEL (VSM) PARAMETERS. (a) VSM TRANSITION MATRIX. (b) VSM STATE DURATION PARAMETERS

(a)

Views	#0	#1	#2	#3
#0	0	0.4	0.2	0.4
#1	0.4	0	0.4	0.2
#2	0.2	0.4	0	0.4
#3	0.4	0.2	0.4	0

(b)

Views	av. time	st. dev.
#0	6min	30s
#1 to #3	1min	10s

The parameters α_i and β_i are derived from the mean and the standard deviation of duration time for the corresponding state, respectively μ_i and σ_i, according to the following expressions:

$$\begin{cases} \alpha_i = \dfrac{\mu_i^2}{\sigma_i^2} \\ \beta_i = \dfrac{\mu_i}{\sigma_i^2}. \end{cases} \tag{13}$$

Although MVC coding techniques and standards are available, modeling the view-selection pattern is an emerging research topic [31], which has been addressed only in a few works [32], [33]. Furthermore, the analysis is limited to the case in which the user behavior is deterministic. Our goal in this paper is to formulate a general stochastic characterization of the problem, being both compact and flexible in the analysis. When referring to a flexible characterization, we mean that we search for a distribution that can characterize both a fast-view-surfing user behavior and a static user that rarely performs view switching. Within this scope, the Gamma distribution applies appropriately because it is able to describe heterogeneous users' behavior, even with only two parameters. For instance, a Gamma distribution parameterized by a small α and a large β can well describe a view-surfing user, whereas a

TABLE IV
AVERAGE FRAME-LOSS RATE DIVERGENCE BETWEEN REAL SEQUENCE AND SYNTHETIC SEQUENCES (MULTIVIEW TV CASE). (a) HIGH-QUALITY SEQUENCE. (b) LOW-QUALITY SEQUENCE

(a)

	c_r= 1		c_r= 1.5		c_r= 2	
buffer	sender	receiver	sender	receiver	sender	receiver
P-HHM-A	0.37	0.5	0.41	0.39	0.15	0.39
P-HHM-B	0.56	1.47	0.7	1.17	0.63	1.06
Model A	1.58	1.84	1.42	1.23	1.11	0.83
nested AR	3.18	3.00	2.05	2.89	1.43	2.17

(b)

	c_r= 1		c_r= 1.5		c_r= 2	
buffer	sender	receiver	sender	receiver	sender	receiver
P-HHM-A	0.18	0.39	0.11	0.87	0.09	1.27
P-HHM-B	0.46	1.13	0.61	1.66	0.62	0.87
Model A	1.47	5.68	1.61	5.52	1.68	4.33
nested AR	1.93	8.04	2.09	7.62	2.15	4.46

Gamma distribution parameterized by large α and β describe a more static user. Moreover, many well-known distributions are particular instances of the Gamma distribution, such as the exponential, the Chi square, and the Erlang distribution. Finally, we note that the specific model of user behavior employed in the performance analysis is however not critical, and a similar study could be conducted with other behavior models.

B. Performance Analysis

We compare the traffic load due to the H.264 MVC source and the synthetic video traffic trace generated by our P-HMM model in both network scenarios defined above. We also include the comparison to a non-Markovian single-view traffic model, [34]. We denote the traffic generated by this method as nested AR. We consider that the MVC traffic is fed into the transmission buffer B_T that is characterized by a buffer size b_T and an output rate r (see Fig. 9). The transmission buffer adopts a first-in–first-out (FIFO) scheduling policy. The buffer output is encapsulated into networks packets in accordance with network packetization rules and transmitted to the destination through the channel. Each packet might be affected by a different (random) delay during transmission. The delay $d[n]$ is the sum of the channel delay $d_C[n]$ and the transmission buffer delay $d_T[n]$. For modeling the channel delay $d_C[n]$, we resort to the quite general and complete channel model introduced by Miao and Chou[4] [35]. After an initial prefetch delay D (namely $D = 2$ s in our study) from the arrival time of the first frame, the playout buffer is drained at a rate given by the MVC compressed stream. If frames are not available in the playout buffer at their decoding deadline, they are considered as lost. We consider three different values for the channel rate, namely 1, 1.5, or 2 times the average bit rate of the MVC source rate. We denote the ratio between the channel rate and then average source rate by the factor c_r.

We have generated a 25-min-long multiview test sequence by concatenating the streams described in Table I, similarly to the sequences used in Section III-A. A three-state model is built by running the EM algorithm on the actual sequence using the same procedure as Section III-A. For each sequence, we generated two streams with high $(Q = 10)$ and low $(Q = 40)$ quality, respectively. We then study the accuracy of our model by comparing the traffic and more particularly the loss rate due to late packets, for both the synthetic traffic and the actual MVC sequence. The loss rate is defined as the ratio between the number of lost frames and the number of transmitted frames at both the sender and receiver buffers. The frame-loss rate is averaged over 10 Monte Carlo simulations. We compare the synthetic traffic and the actual sequence at different values of the buffers size in order to show that the model is able to describe the MVC traffic both in case of acceptable frame-loss rate and very large loss rate.

First, we compare the sender buffer frame loss rate for different values of the transmission buffer size. Figs. 10(a) and 11(a) show respectively these results for the Multiview TV and the Interactive TV cases, for $c_r = 2$ and low stream quality value. It can be seen that the actual MVC sequence and the synthetic sequence share a similar frame loss rate for different channel rates and transmission buffer sizes. Note that the close similarity is due to the model's capability to describe higher order statistics by means of the nonstationary activity level chain.

Finally, we compare the overall frame-loss rate, i.e., the sum of lost frames at both the sender and receiver buffers, divided by the total number of frames, as a function of the receiver buffer size.[5] Figs. 10(b) and 11(b) show these results for the Multiview TV and Interactive TV cases, respectively. Table V summarizes these results for other test settings, quantifying the model's accuracy as the average absolute difference e of the frame-loss rate, between the real sequence and the synthetic sequence.

The average is taken over the different buffer sizes under consideration, and the frame-loss rate is expressed in percent. It can be seen that the synthetic sequence closely follows the behavior

[4]Specifically, we have adopted the same numerical channel model parameters as in [35], $\alpha = 80,\ n = 4,\ \chi = 0.025$.

[5]To determine the overall frame-loss rate, we set the transmission buffer size to be large enough to guarantee that the frame-loss rate at the transmission side is not higher than 5%.

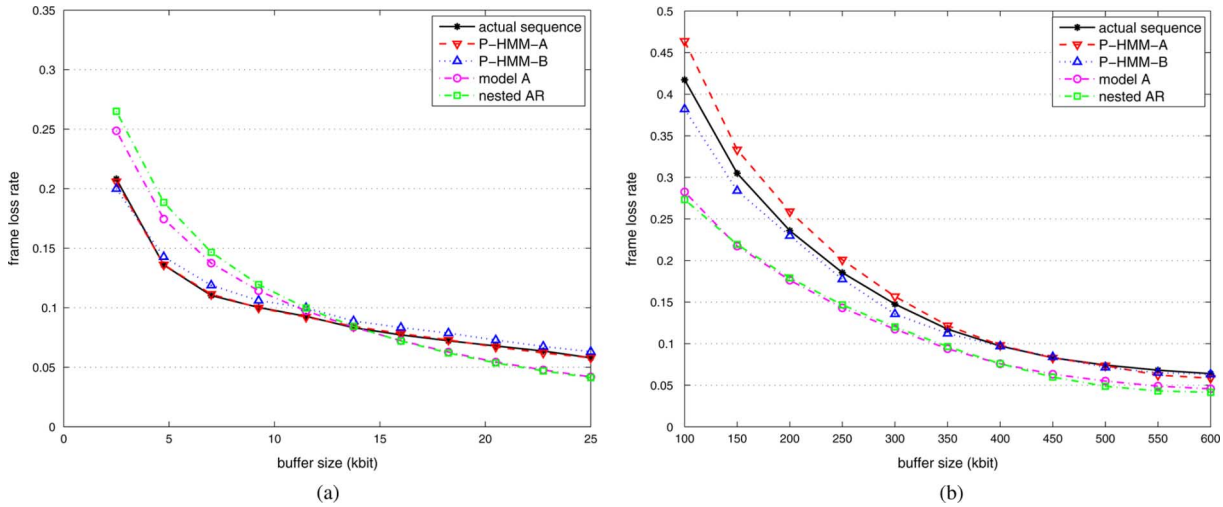


Fig. 10. Frame loss rate in Multiview TV services (low-quality stream, $c_r = 2$). (a) Transmission buffer. (b) Playout buffer.

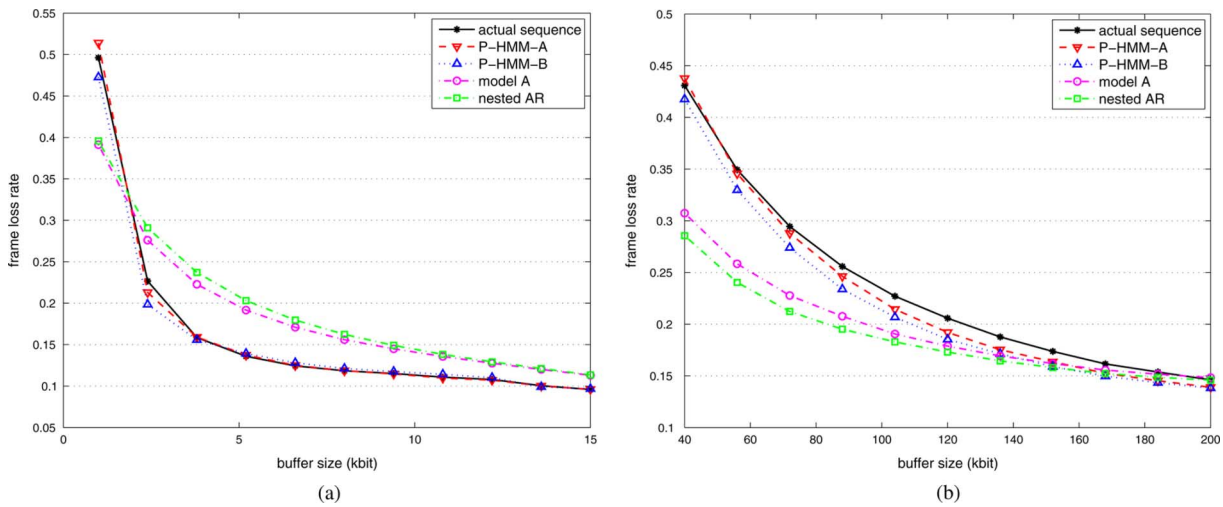


Fig. 11. Frame loss rate in Interactive TV services (low-quality stream, $c_r = 2$). (a) Sender buffer. (b) Receiver buffer.

TABLE V
AVERAGE FRAME-LOSS RATE DIVERGENCE BETWEEN REAL SEQUENCE AND SYNTHETIC SEQUENCES (INTERACTIVE TV CASE). (a) HIGH-QUALITY SEQUENCE. (b) LOW-QUALITY SEQUENCE

(a)

	c_r= 1		c_r= 1.5		c_r= 2	
buffer	sender	receiver	sender	receiver	sender	receiver
P-HHM-A	0.29	0.23	0.32	0.18	0.12	0.23
P-HHM-B	1.08	0.88	1.18	0.75	0.65	0.72
Model A	2.43	1.83	1.69	0.55	1.05	0.86
nested AR	1.24	2.17	0.63	0.89	0.75	0.74

(b)

	c_r= 1		c_r= 1.5		c_r= 2	
buffer	sender	receiver	sender	receiver	sender	receiver
P-HHM-A	0.39	1.21	0.34	0.91	0.34	0.91
P-HHM-B	0.59	3.53	0.65	1.50	0.67	1.61
Model A	4.02	10.24	4.22	3.85	4.27	3.94
nested AR	4.57	14.20	4.78	5.11	4.82	4.80

of the actual MVC source; specifically, the difference between the frame-loss rate of the model and the one of the actual sequence is smaller than 0.03 for most of the playout buffer sizes under examination. Since the frame arrival time at the playout buffer depends on the size of the previous frames that are transmitted, the close similarity between the synthetic and the actual traffic demonstrates the accuracy of the model in characterizing MVC traffic statistics. In addition, even the P-HMM-B model,

which is trained on different sequences than the test sequences, is able to capture relevant traffic features of actual data, thus providing a good frame-loss rate estimation for both transmission and playout buffer, as seen from Table V. Finally, it is worth noting that both model A and nested AR fail in predicting the buffer losses, as they have not been designed for the specific structure of MVC streams. These simulations are aimed to show that our model effectively allows to predict the real frame loss experienced by the multiview sequence, which can be quite high unless a suitable buffering is performed. Thereby, our model permits to properly select the operating ranges of the system parameters, namely the transmitter/receiver buffer sizes. Therefore, the P-HMM can replace real MVC sequences in the dimensioning of transmit and receive buffers. It can be employed both for synthetic trace generation as well as for theoretical network performance analysis.

V. Conclusion

In this paper, we have presented a new stochastic model characterizing the frame size sequence for MVC VBR sources. The model exploits a Poisson hidden Markov model representing the random frame sizes of the different MVC encoded views as a function of the random real video scene activity variations. We have also derived a stable EM algorithm that is applicable to long data sequences for the P-HMM parameter estimation. We have shown through extensive simulations that our model accurately predicts the sequence of frame sizes in an MVC stream. We have also applied our model to traffic load prediction in two different network scenarios, namely a multiview TV service and an interactive TV service. Simulation results show that the synthetic traffic generated by the proposed model strongly resembles the traffic due to real MVC video traces. The model is able to accurately characterize a class of MVC streams sharing similar content characteristics with the training data. The model is therefore an appropriate tool for different networking problems, such as network dimensioning, resource allocation, and call admission control.

References

[1] Y. Chen, Y. Wang, K. Ugur, M. M. Hannuksela, J. Lainema, and M. Gabbouj, "The emerging MVC standard for 3D video services," *EURASIP J. Adv. Signal Process.*, vol. 2009, p. 786015, 2009.

[2] S. Xu and Z. Huang, "A gamma autoregressive video model on ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 2, pp. 138–142, Apr. 1998.

[3] N. Ansari, H. Liu, Y. Q. Shi, and H. Zhao, "On modeling MPEG video traffics," *IEEE Trans. Broadcast.*, vol. 48, no. 4, pp. 337–347, Dec. 2002.

[4] D. P. Heyman and T. V. Lakshman, "Source models for VBR broadcast-video traffic," *IEEE/ACM Trans. Netw.*, vol. 4, no. 1, pp. 40–48, Feb. 1996.

[5] S. Colonnese, S. Rinauro, L. Rossi, and G. Scarano, "H.264 video traffic modeling via hidden Markov process," presented at the 17th EUSIPCO, Glasgow, U.K., Aug. 24–28, 2009.

[6] A. Matrawy, I. Lambadaris, and C. Huang, "MPEG4 traffic modeling using the transform expand sample methodology," in *Proc. 4th IEEE IWNA4*, 2002, pp. 249–256.

[7] S. Colonnese, P. Frossard, S. Rinauro, L. Rossi, and G. Scarano, "Joint source and sending rate modeling in adaptive video streaming," *Signal Process., Image Commun.*, vol. 28, no. 5, pp. 403–316, May 2013.

[8] Z. Zang, J. Kurose, J. D. Salehi, and D. Towsley, "Smoothing, statistical multiplexing, and call admission control for stored video," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 6, pp. 1148–1166, Aug. 1997.

[9] A. Pulipaka, P. Seeling, M. Reisslein, and L. J. Karam, "Traffic and statistical multiplexing characterization of 3D video representation formats," *IEEE Trans. Broadcast.*, vol. 59, no. 2, pp. 382–389, Jun. 2013.

[10] L. Rossi, J. Chakareski, P. Frossard, and S. Colonnese, "A non-stationary hidden Markov model for multiview video traffic," in *Proc. 17th IEEE ICIP*, 2010, pp. 2921–2924.

[11] M. Russell and R. Moore, "Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition," in *Proc. ICASSP*, 1985, vol. 10, pp. 5–8.

[12] M. E. Sousa-Vieira, "On modeling 3-D video traffic," *Adv. Multimedia Model.*, vol. 732, pp. 327–335, 2013.

[13] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[14] S. Park and D. Sim, "An efficienct rate-control algorithm for multiview video coding," in *Proc. IEEE 13th Int. Symp. Consumer Electron.*, 2009, pp. 115–118.

[15] G. Van Wallendael, S. Van Leuven, J. De Cock, F. Bruls, and R. Van De Walle, "3D video compression based on high efficiency video coding," *IEEE Trans. Consumer Electron.*, vol. 58, no. 1, pp. 137–145, Feb. 2012.

[16] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1518–1569, Jun. 2002.

[17] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[18] S. Z. Yu, "Hidden semi-Markov models," *Artif. Intell.*, vol. 174, pp. 215–243, 2010.

[19] H. Li, G. Liu, Z. Zhang, and Y. Li, "Adaptive scene-detection algorithm for VBR video stream," *IEEE Trans. Multimedia*, vol. 6, no. 4, pp. 624–633, Aug. 2004.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.

[21] U. K. Sarkar, S. Ramakrishnan, and D. Sarkar, "Modeling full-length video using Markov-modulated gamma-based framework," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 638–649, Aug. 2003.

[22] "MPEG-FTV test sequence download page," [Online]. Available: http://www.tanimoto.nuee.nagoya-u.ac.jp/~fukushima/mpegftv/

[23] J. Daase, U. Goelz, P. Kauff, K. Mueller, O. Schreer, A. Smolic, R. Tanger, and T. Wiegand, "Fraunhofer HHI test data sets for MVC," ISO/IEC JTC1/SG29/WG11, Document MPEG2005/M11894, 2005.

[24] H. Xiong, H. Lv, Y. Zhang, L. Song, Z. He, and T. Chen, "Subgraphs matching-based side information generation for distributed multiview video coding," *EURASIP J. Adv. Signal Process.*, vol. 2009, pp. 1–18, 2009.

[25] "Joint Multiview Coding (JMVC) v7.0," available via CVS at garcon.ient.rwth-aachen.de.

[26] N. D. Doulamis, A. D. Doulamis, G. E. Konstantoulakis, and G. I. Stassinopoulos, "Efficient modeling of VBR MPEG-1 coded video sources," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 1, pp. 93–112, Feb. 2000.

[27] C. Y. Lee, C. K. Hong, and K. Y. Lee, "Reducing channel zapping time in IPTV based on user's channel selection behaviors," *IEEE Trans. Broadcast.*, vol. 56, no. 3, pp. 321–330, Sep. 2010.

[28] A. M. Kermarrec, E. Le Merrer, y. Liu, and G. Simon, "Surfing peer-to-peer IPTV: Distributed channel switching," in *Proc. EuroPar*, 2009, pp. 574–586.

[29] H. Joo, H. Song, D. B. Lee, and I. Lee, "An effective IPTV channel control algorithm considering channel zapping time and network utilization," *IEEE Trans. Broadcast.*, vol. 54, no. 2, pp. 208–216, Jun. 2008.

[30] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain, "Watching television over an IP network," in *Proc. ACM Internet Meas. Conf.*, 2008, pp. 71–84.

[31] J. Chakareski, "Adaptive multi-view video streaming: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 51, no. 5, pp. 94–100, May 2013.

[32] Z. Pan, Y. Ikuta, M. Bandai, and T. Watanabe, "A user dependent system for multi-view video transmission," in *Proc. IEEE AINA*, Mar. 22–25, 2011, p. 732, 739.

[33] Z. Pan, Y. Ikuta, M. Bandai, and T. Watanabe, "User dependent scheme for multi-view video transmission," in *Proc. IEEE ICC*, 2011, pp. 1–5.

[34] D. Liu, E. I. Safa, and W. Sun, "Nested auto-regressive processes for MPEG-encoded video traffic modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 2, pp. 169–183, Feb. 2001.

[35] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.

Lorenzo Rossi (S'08–M'12) was born in Rome, Italy. He received the Laurea degree (*cum laude*) in telecommunication engineering and Ph.D. degree in information and communication engineering from the Università "La Sapienza" di Roma, Rome, Italy, in 2006 and 2011, respectively.

Between 2011 and 2013, he was with the Corporate Research and Development Center, Vitrociset, where he worked as a scientific expert and technical coordinator in different European and Italian research projects. He currently collaborates with the Dipartimento di Elettronica e Telecomunicazioni (DIET), Università "La Sapienza." His research interests include video streaming services, statistical signal processing, image processing, and data fusion.

Dr. Rossi received the Accenture Degree Award in 2007.

Jacob Chakareski received the M.Sc. degree from the Worcester Polytechnic Institute (WPI), Worcester, MA, USA, in 1999, and the Ph.D. degree from Rice University, Houston, TX, USA, and Stanford University, Stanford, CA, USA, in 2006, all in electrical and computer engineering.

He is an Assistant Professor of electrical and computer engineering with the University of Alabama, Tuscaloosa, AL, USA. He was a Senior Scientist with Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland, where he conducted research, supervised students, and lectured. He also held research positions with Microsoft, Hewlett-Packard, and Vidyo, a leading provider of Internet telepresence solutions. He is an Advisory Board member of Mainframe2, an innovative cloud computing start-up with a bright future. He has authored one monograph, three book chapters, and over 100 international publications and holds five US patents. His current research interests include graph-based information processing, computer networks, immersive communication, and social computing. He eagerly pursues ultrasound applications in telemedicine, remote sensing, and biomedicine and is passionate about bridging science and technology via entrepreneurial activity.

Dr. Chakareski is a member of Tau Beta Pi and Eta Kapa Nu. He actively participates in technical and organizing committees of major IEEE conferences. He was the Publicity Chair of the Packet Video Workshop 2007 and 2009 and the Workshop on Emerging Technologies in Multimedia Communications and Networking at ICME 2009. He has organized and chaired a special session on telemedicine at MMSP 2009. He was the Technical Program Co-Chair of Packet Video 2012 and the General Co-Chair of the IEEE SPS Seasonal School on Social Media Processing 2012. He was a Guest Editor of the *Peer-to-Peer Networking and Applications* 2013 special issue on P2P cloud systems. He is a recipient of the Technical University Munich Mobility Fellowship, the University of Edinburgh Chancellor's Fellowship, and fellowships from the Soros Foundation and the Macedonian Ministry of Science. He was the recipient of the Texas Instruments Graduate Research Fellowship at Rice University, the Swiss NSF Ambizione Career Development Award, the Best Student Paper Award at the SPIE VCIP 2004 Conference, and the Best Paper Award of the *Stanford Electrical Engineering and Computer Science Research Journal* for 2003.

Pascal Frossard (S'96–M'01–SM'04) received the M.S. and Ph.D. degrees in electrical engineering from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1997 and 2000, respectively.

Between 2001 and 2003, he was a Member of the Research Staff with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, where he worked on media coding and streaming technologies. Since 2003, he has been a faculty member with EPFL, where he heads the Signal Processing Laboratory (LTS4). His research interests include image representation and coding, visual information analysis, distributed image processing and communications, and media streaming systems.

Dr. Frossard has been the General Chair of IEEE ICME 2002 and Packet Video 2007. He has been the Technical Program Chair of EUSIPCO 2008 and a member of the organizing or technical program committees of numerous conferences. He has been an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING since 2010, the IEEE TRANSACTIONS ON MULTIMEDIA from 2004 to 2012, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 2006 to 2011. He has been the Vice-Chair of the IEEE Image, Video and Multidimensional Signal Processing Technical Committee since 2007, and an elected member of the IEEE Visual Signal Processing and Communications Technical Committee since 2006 and of the IEEE Multimedia Systems and Applications Technical Committee since 2005. He has served as Vice-Chair of the IEEE Multimedia Communications Technical Committee from 2004 to 2006 and as a member of the IEEE Multimedia Signal Processing Technical Committee from 2004 to 2007. He received the Swiss NSF Professorship Award in 2003, the IBM Faculty Award in 2005, the IBM Exploratory Stream Analytics Innovation Award in 2008, and the IEEE TRANSACTIONS ON MULTIMEDIA Best Paper Award in 2011.

Stefania Colonnese was born in Rome, Italy. She received the Laurea degree (*magna cum laude*) from the Università "La Sapienza" di Roma, Rome, Italy, in 1993, and the Ph.D. degree from the Università di Roma "Roma Tre," Rome, Italy, in 1997, both in electronic engineering.

She has been active in the MPEG-4 standardization activity, and she has been involved in the N2 Core Experiment on automatic Video Segmentation. In 2001, she joined the Dipartimento di scienza e tecnica dell'Informazione e della Comunicazione (INFO-COM), now Dipartimento di Elettronica e Telecomunicazioni (DIET), Università "La Sapienza," as an Assistant Professor. She has been a Visiting Scholar with The State University of New York at Buffalo, Buffalo, NY, USA, in 2011, and a Visiting Teacher within the Erasmus teaching Staff Mobility Program at Università Paris 13, Villetaneuse, France, in 2012. Her current research interests lie in the areas of signal and image processing, video communications, multicamera video processing, and networking.

Dr. Colonnese has served as reviewer of several journals, including the IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON IMAGE PROCESSING, *Journal of Signal Processing: Image Communication*, and IEEE SIGNAL PROCESSING LETTERS. She is currently an Associate Editor of the *International Journal of Digital Multimedia Broadcasting*. She served in the Technical Program Committee of IEEE/EURASIP EUVIP 2011 (Paris, France, 2011) and of Computational Modeling of Objects Presented in Images: Fundamentals, Methods and Applications Compimage 2012 (Rome, Italy, 2012). She is currently Student Session Chair for IEEE/EURASIP EUVIP 2013.