A Poisson Hidden Markov Model for Multiview Video Traffic

Lorenzo Rossi, Jacob Chakareski, Pascal Frossard, 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—Digital video broadcasting, three dimensional TV, hidden Markov models, multiview video.

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 a 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].

There is however a lack of traffic models for multiview video communication services. We propose here a new traffic model for MVC content in order to characterize the frame size sequence observed at the output of an MVC variable bit rate (VBR) source. Specifically, building on our preliminary work [9], we design a doubly stochastic source model, namely a Poisson Hidden Markov Model (P-HMM) [10], in which the first (hidden) layer consists of a non-stationary chain modeling the video activity level and the second layer represents the frame sizes of the different MVC encoded views. Besides, we extend the P-HMM parameter estimation algorithm for short observation sequences presented in [10] and adapt it to long sequences such as those encountered in video communication services. We assess the model performances 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 one, we name Multiview TV, the server simultaneously streams all the MVC encoded views to the client. Our model is shown to be able to predict the state of sender and receiver buffers in Multiview TV. In the second one, named Interactive TV, the client dynamically selects the views during the streaming session by means of a feedback channel. Due to the MVC encoding dependencies, the server transmits a composite stream encompassing all 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.

The main contribution of this paper can be summarized as follows:

- a non-stationary 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 Poisson Hidden Markov Model (P-HMM) and discuss its feasibility; 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

A MVC stream jointly encodes different video sequences captured by multiple cameras with overlapping fields of view. Let us denote by $N_{\text{View}}$ 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 $N_{\text{View}} - 1$ 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 [11]. The encoding dependencies give rise to a generalized Groups Of Pictures (GOP) structure of duration $N_{\text{GOP}}$, constituted by $N_f = N_{\text{View}} \times N_{\text{GOP}}$ frames. Figures 1 and 2 show two examples of such MVC GOPs.

![GOP structure and encoding hierarchy (N_{GOP} = 8).](image1)

![GOP structure and encoding hierarchy (N_{GOP} = 4).](image2)

In order to control the size of the bitstreams, different rate control algorithms [12] can be implemented in MVC encoders. We however focus in this paper on Variable Bit Rate (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].

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 non-stationary stochastic process. We propose a two-layer stochastic process model inspired from [13]. We build a new Poisson Hidden Markov Model (P-HMM) [10], in which the first (hidden) layer is a discrete time Markov chain whose states represent different video activity levels; the second layer represents the frame size sequence corresponding to a given activity level.

We make the following assumptions about the non-stationarity 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 approximate the duration of scenes with 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

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

When a state transition occurs, the model is described by a state transition matrix $\Pi$, whose element $\pi_{ij}$ 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, $\pi_i$ 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

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

representing the set of frames sizes in the $n$-th GOP of the compressed multiview content. The vector $x[n]$ is emitted in accordance to a multivariate probability mass function

$$P(x[n] = x) = \prod_{i=0}^{N_f-1} P(x_i[n] = x_i),$$

where

$$P(x_i[n] = x_i) = \frac{e^{-\lambda_i} \lambda_i^{x_i}}{x_i!}.$$
(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 \( b_i[] \), \( i = 1, \ldots, N_s \), 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) according to the desired accuracy performance.

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 [14], we can resort to one of the estimation algorithms employed for such models. In particular, an estimation procedure called Expectation-Maximization (EM) algorithm [15] is widely used for HMMs. A version of the EM algorithm has been proposed for P-HMMs in [10]. 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 non-stationary hidden state durations. We present the parameter estimation in detail below.

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

\[
\Theta^{(m+1)} = \arg \max_{\Theta} Q(\Theta | \Theta^{(m)}). \tag{1}
\]

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

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

\[
\hat{d}_i[k] = \begin{cases} 
1 - D_i[k - 1] & \text{if } k = N - n - 1 \\
d_i[k] & \text{otherwise}
\end{cases} \tag{2}
\]

The parameter \( D_i[k] \) is the cumulative distribution of the state duration times. Then, the computations in our EM algorithm, as applied to P-HMMs, comprise i) the computation of forward probabilities, ii) the computation of backward probabilities, iii) the estimation of the parameter set. The first two steps 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 [14] for further details). Then, the parameter set is expressed as function of these auxiliary variables. We first define the following forward probabilities \(^1\) for \( n = 0, \ldots, N - 1 \)

\[
\alpha_n(i, k) \overset{\text{def}}{=} P(s_n = i, \ldots, s_{n+k} = i), \quad s_{n+k+1} \neq i | y_0^n, \Theta^{(m)}), \tag{3}
\]

\[
\alpha_n(i) \overset{\text{def}}{=} P(s_n = i | y_0^n, \Theta^{(m)}). \tag{4}
\]

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

\[
\alpha_n(i, N - n - 1) \overset{\text{def}}{=} P(s_n = i, \ldots, s_{N-1} = i | y_0^n, \Theta^{(m)}). \tag{5}
\]

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 till time \( n \).

**Algorithm 1** Computing forward probabilities \( \alpha_n(i, k) \) and \( \alpha_n(i) \).

1: \textbf{for} \( n = 0 \rightarrow N - 1 \) \textbf{do}

2: \quad \textbf{for} \( k = 0 \rightarrow N - n - 1 \) \textbf{do}

3: \quad \quad \textbf{if} \( n = 0 \) \textbf{then}

4: \quad \quad \quad \alpha_0(i, k) \propto \pi_i \hat{d}_i[k] b_i[y[0]]

5: \quad \quad \textbf{else}

6: \quad \quad \quad \alpha_n(i, k) \propto b_i[y[n]] \left( \sum_{j \neq i} \alpha_{n-1}(j, 0) \pi_j \hat{d}_j[k] \right) + \alpha_{n-1}(i, k + 1)

7: \quad \textbf{end if}

8: \quad \textbf{end for}

9: \quad \alpha_n(i) = \sum_k \alpha_n(i, k)

10: \textbf{end for}

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

\[
\gamma_n(i, k) \overset{\text{def}}{=} P(s_n = i, \ldots, s_{n+k} = i | y_0^{N-1}, \Theta^{(m)}), \tag{5}
\]

\[
n = 0, \ldots, N - 1; \ k < N - n - 1
\]

\[
\xi_n(i, j, k) \overset{\text{def}}{=} P(s_{n-1} = i, s_n = j, \ldots, s_{n+k} = j, s_{n+k+1} \neq j | y_0^{N-1}, \Theta^{(m)}), \tag{6}
\]

\[
n = 1, \ldots, N - 1; \ k < N - n - 1
\]

Note that we resort to a different definition for the backward probabilities with respect to the usual \( \beta \) notation [14]. In [14],

\(^1\) Our definitions differ from [10] in order to avoid numerical instability.

\(^2\) The normalization coefficient for \( \alpha_n(i, k) \) is calculated by summation over \( i \) and \( k \).
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. The Algorithm 2 illustrates the backward probabilities computation, where δ denotes the Kronecker function.

Finally, the state duration is expressed as:

\[ \lambda_i = \left( \sum_{n=1}^{N-1} \sum_{k=0}^{N-2} \sum_{j=1}^{N} \sum_{j\neq i} \xi_n(j, i, k) + \sum_{k=0}^{N-1} k \gamma_0(i, k) \right)^{-1} \cdot \left( \sum_{n=1}^{N-1} \sum_{k=0}^{N-2} \sum_{j=1}^{N} \sum_{j\neq i} \xi_n(j, i, k) + \sum_{k=0}^{N-1} \gamma_0(i, k) \right). \]  

Note that a single iteration of the estimation algorithm consists of calculating first the state sequence probabilities (3)-(6), which is the expectation step. Subsequently, we compute the new parameter estimates by averaging the observations weighted with the state probabilities (7)-(10), which corresponds to the maximization step of an iteration of the algorithm. Convergence is assured by Jensen’s inequality [16].

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, with the statistics evaluated on a composite MVC test sequence. The P-HMM parameters are estimated by 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 ability to represent not only the training sequence, but also other sequences with similar content. A comparison with a well-known single view VBR model is also carried out.

### A. MVC encoder settings

The composite MVC sequences considered in the model assessment are generated by concatenating several test sequences with very different activity levels, as reported in Table I. All the sequences are in CIF format, with a frame rate of 25 fps, and Nview = 4 views. The resulting MVC composite sequence is approximately 6 minutes long. We encode the sequence using the two different GOP structures reported in Figures 1 and 2. Both structures exhibit motion compensation dependencies among the views for anchor and non-anchor frames [17]. 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 Figure 1 is mainly due to the residuals of the motion compensation whereas for sequences encoded using the GOP in Figure 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. We use JMVC v7.0 to encode the sequences.
Then, we use the different MVC bitstreams to build traffic models.

![Table I](https://via.placeholder.com/150)

**TABLE I**

**TEST SEQUENCES USED TO GENERATE THE COMPOUND SEQUENCE.**

As we discussed in the previous section, the number of bins in the pmfs of the model may be different for different frame types in order to have a trade-off between performance and model complexity. In our tests we use the number of bins shown in Table II (a) and (b) for the GOP structures in Figures 1 and 2, respectively. The bins are placed in the interval between the minimum and the maximum observed frame size for each frame type.

![Table II](https://via.placeholder.com/150)

**TABLE II**

**NUMBER OF BINS IN THE PMF FOR EACH FRAME OF THE GOP GIVEN IN DISPLAY ORDER.**

In the simulations, we employ a 3-state model in order to represent low, medium and high level activity. A first coarse estimation of the model parameter 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 likelihood 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 pmfs. The bins are converted to the mean frame size of the interval they represent. The synthetic traffic generation is described in Algorithm 3.

![Algorithm 3](https://via.placeholder.com/150)

**Algorithm 3** ** Synthetic traffic generation.**

1. Initial state \(i\) extracted according to the distribution \(\pi_1, \pi_2, \ldots, \pi_N\).
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 II.
13. **end while**

**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. Figures 3, 4 and 5 show the acf and the Q-Q plot for different compressed MVC sequences. Q-Q plot measures how two distributions are similar 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 are. 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 pmfs to have an explicit probability distribution and that we employ the Poisson distribution for the state durations, thus forcing a non-stationary 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 [19]. Specifically, we focus on the "model A" in [19], 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 7-state Markov chain for the activity level. Moreover, an ad-hoc parameter estimation procedure is used in [19]. We see in the Figures 3, 4 and 5 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. 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. Figures 6 and 7 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 [19] is not able to capture the statistics for each single view. Conversely, our models can characterize efficiently 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 P-HMM-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 Figure 8. In the first one, called “Multiview TV”, the server sends all the MVC content to the user. In the second one,
Fig. 5. Q-Q plot and autocorrelation estimated on the actual MVC sequence and the synthetic sequences (GOP structure in Figure 1, medium quality stream).

Fig. 6. Q-Q plot and autocorrelation estimated on the actual MVC sequence (View #1) and the synthetic sequences (GOP structure in Figure 1, medium quality stream).

Fig. 7. Q-Q plot and autocorrelation estimated on the actual MVC sequence (View #3) and the synthetic sequences (GOP structure in Figure 1, medium quality stream).
denoted as "Interactive TV", the user requests one view at the 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.

Fig. 8. System description for multiview services. The feedback channel is present only in the interactive TV case.

From the point of view of the MVC source model, the above services differ in that the traffic generated during the Multiview TV session depends only on the encoded video, whereas the traffic generated during the Interactive TV session depends also on the view selection process. Thereby, 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 [20]-[23]. 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 describe the foreground 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 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 distribution.

Formally, let $d_i^n[k]$ be the density function for the duration time in state $i$:

$$d_i^n[k] = \frac{\beta_i^n}{\Gamma(\alpha_i^n)} \alpha_i^n e^{-\alpha_i^n k} \beta_i^n$$

(11)

The parameters $\alpha_i$ and $\beta_i$ are derived from the mean and the standard deviation of duration time for the corresponding state, respectively $\mu_i$ and $\sigma_i$, according to the following expressions:

$$\alpha_i = \frac{\mu_i^2}{\sigma_i^2}$$

$$\beta_i = \frac{\mu_i}{\sigma_i^2}$$

(12)

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.

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

TABLE III

View Switching Model (VSM) Parameters.

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 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 Figure 8). 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 [24]. After an initial prefetch delay $D$ (namely $D=2$ sec. 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 bitrate 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 minute long multiview test sequence, by concatenating the streams described in Table I, similarly to the sequences used in Section III-A. A 3-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.

$^4$Specifically, we have adopted the same numerical channel model parameters as in [24], $\alpha = 80$, $n = 4$, $\chi = 0.025$. 
Fig. 9. Frame loss rate in Multiview TV services (low quality stream, $c_f = 2$).

First, we compare the sender buffer frame loss rate for different values of the buffer size. Figures 9(a), 10(a) and Figures 11(a), 12(a) show respectively these results for the Multiview TV and the Interactive TV cases, at different channel rates and stream quality values. 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 non-stationary 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. Figures 9(b), 10(b) and Figures 11(b), 12(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 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 play-out 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 ones used for the parameter estimation, is able to capture relevant traffic features of actual data thus providing a good frame loss rate estimation for both transmission and play-out buffer, as seen from Table V. 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 variable bit rate (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. Ugr, M.M. Hannuksela, J. Lainema, and M. Gabbouj, “The Emerging MVC Standard for 3D Video Services,” EURASIP Journal on Advances in Signal Processing, vol. 2009, Article ID 786015, 13 pages, 2009. doi:10.1155/2009/786015.
[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, “Markov Model of H.264 Video Traffic,” presented at the 4th Int. Sym. on Im./Video Comm. (ISIVC ’08), Bilbao - Spain, July 9-11th, 2008.
(a) High quality sequence

| buffer     | $c_r = 1$ | $c_r = 1.5$ | $c_r = 2$ |
|------------|-----------|-------------|-----------|
| P-HHM-A    | 0.37      | 0.41        | 0.15      |
| P-HHM-B    | 0.56      | 0.7         | 0.63      |
| Model A    | 1.58      | 1.42        | 1.11      |

(b) Low quality sequence

| buffer     | $c_r = 1$ | $c_r = 1.5$ | $c_r = 2$ |
|------------|-----------|-------------|-----------|
| P-HHM-A    | 0.18      | 0.11        | 0.09      |
| P-HHM-B    | 0.46      | 0.61        | 0.62      |
| Model A    | 1.47      | 1.61        | 1.68      |

TABLE IV
AVERAGE FRAME LOSS RATE DIVERGENCE BETWEEN REAL SEQUENCE AND SYNTHETIC SEQUENCES (MULTIVIEW TV CASE).

(a) High quality sequence

| buffer     | $c_r = 1$ | $c_r = 1.5$ | $c_r = 2$ |
|------------|-----------|-------------|-----------|
| P-HHM-A    | 0.29      | 0.32        | 0.12      |
| P-HHM-B    | 1.08      | 1.18        | 0.65      |
| Model A    | 2.43      | 1.69        | 1.05      |

(b) Low quality sequence

| buffer     | $c_r = 1$ | $c_r = 1.5$ | $c_r = 2$ |
|------------|-----------|-------------|-----------|
| P-HHM-A    | 0.39      | 0.34        | 0.34      |
| P-HHM-B    | 0.59      | 0.65        | 0.67      |
| Model A    | 4.02      | 4.22        | 4.27      |

TABLE V
AVERAGE FRAME LOSS RATE DIVERGENCE BETWEEN REAL SEQUENCE AND SYNTHETIC SEQUENCES (INTERACTIVE TV CASE).

Fig. 10. Frame loss rate, Multiview TV services (high quality stream, $c_r = 1$).

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

[7] S. Colonnese, S. Rinauro, L. Rossi, and G. Scarano, “H.264 Video Traffic Modeling Via Hidden Markov Process,” in Proc. EUSPICO-09, 2009, pp.2221-2225, 2009.

[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] L. Rossi, J. Chakareski, P. Frossard, and S. Colonnese, “A non-stationary hidden Markov model for multiview video traffic,” in Proc. 17th IEEE International Conference on Image Processing (ICIP), 2010, pp.2921-
Fig. 11. Frame loss rate in Interactive TV services (low quality stream, $c_r = 2$).

Fig. 12. Frame loss rate in Interactive TV services (high quality stream, $c_r = 1$).

2924.

[10] M. Russell and R. Moore, “Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition,” ICASSP 85, 1985.

[11] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the Scalable Video Coding Extension of the H.264/AVC Standard,” IEEE Trans. on Circuits and System for Video Technology, vol. 17, no. 9, pp. 1103-1120, September 2007.

[12] S. Park and D. Sim, “An efficient rate-control algorithm for multiview video coding,” IEEE 13th International Symposium on Consumer Electronics, 2009.

[13] N.D. Doulamis, A.D. Doulamis, G.E. Konstantoulakis, and G.I. Stassinopoulos, “Efficient modeling of VBR MPEG-1 coded video sources,” IEEE Trans. on Circ. and Sys. for Video Tech., vol.10, no.1, pp.93-112, Feb. 2000.

[14] L.R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” Proceedings of the IEEE, vol. 77, no. 2, pages 257-286, 1989.

[15] 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.

[16] Y. Ephraim and N. Merhav, “Hidden Markov processes,” IEEE Trans. Inform. Theory, vol. 48, pp. 1518-1569, June 2002.

[17] 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 Journal on Advances in Signal Processing, vol. 2009, pp. 1-18, 2009.

[18] Joint multiview coding (JMVC) v7.0, available via CVS at garcon.ient.rwth-aachen.de.

[19] 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, 2003.

[20] 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. on Broadcasting, vol. 56, no. 3, pp. 321-330, Sept. 2010.

[21] A.M. Kermarrec, E. Le Merrer, y. Liu, and G. Simon, “Surfing Peer-to-Peer IPTV: Distributed Channel Switching,” in Proc. of EuroPar, 2009, pp. 574-586.

[22] 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, June 2008.

[23] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain, “Watching Television Over an IP Network,” Proc. of ACM Internet Measurement Conference, 2008.

[24] 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.