An Analytical Framework to Model an H.264 Video Encoder for Transmission on the Internet
FRANCESCO LICANDRO, GIOVANNI SCHEMBRA, VALERIA TOMASELLI
Dipartimento di Ingegneria Informatica e delle Telecomunicazioni - University of Catania – ITALY

Abstract - One of the most widely used encoding standards for video transmission at a low bit rate is H.264. Many techniques have been introduced in the literature to control the output stream of an H.264 encoder, in order to respect the available bandwidth or the declared bandwidth. The majority of them allow users to define a given target for the output bit rate of an H.264 video source; this is achieved by skipping captured frames before encoding. The higher the number of frames the encoder skips, the higher the motion discontinuity perceived by the human eye. The main contribution of the framework is to support designers in the choice of H.264 traffic models for the most common video sources. However, while many analytical models have been defined for MPEG video sources [1], attention regarding H.264 traffic sources has mainly been devoted to the encoder process and to sizing it empirically. No attention has been paid, to the best of our knowledge, to analytically modeling it in order to optimize encoding parameter design. The main contribution of this paper is to introduce a Markov-based analytical framework to model an H.264 video encoder [2]. This framework allows us to calculate the performance of an H.264 encoding system which applies a frame-skipping control policy in order to respect an emission target for each encoded frame. The target represents the maximum number of packets the source can transmit in the network in a frame interval. When the source generates frames with a number of bits greater than the target, these frames are buffered. In order to limit the packetization delay, a threshold is used to skip frames. Moreover, besides the target, the analytical framework modeling the H.264 video source allows us to choose how many P-frames to encode between two I-frames, and the number of B-frames to encode, in order to minimize the skipping probability.
The paper is structured as follows. Section 2 presents the H.264 encoding system we model in the paper. Then, Section 3 introduces a statistical analysis of H.264 traces. In Section 4 the H.264 video encoding system is modeled as a switched batch Bernoulli process (SBBP). Section 5 applies the model to a case study and derives numerical results and important hints for the encoding system designer to efficiently dimension the encoding parameters. Section 6 completes the model with derivation of the main performance parameters. Finally, the authors’ conclusions are drawn in Section 7.

1 Introduction
Taking into account that video is expected to constitute most of the traffic on the network in the immediate future, the challenge today is to define traffic models for the most common video sources. However, while many analytical models have been defined for MPEG video sources [1], attention regarding H.264 traffic sources has mainly been devoted to the encoder process and to sizing it empirically. No attention has been paid, to the best of our knowledge, to analytically modeling it in order to optimize encoding parameter design.
The main contribution of this paper is to introduce a Markov-based analytical framework to model an H.264 video encoder [2]. This framework allows us to calculate the performance of an H.264 encoding system which applies a frame-skipping control policy in order to respect an emission target for each encoded frame. The target represents the maximum number of packets the source can transmit in the network in a frame interval. When the source generates frames with a number of bits greater than the target, these frames are buffered. In order to limit the packetization delay, a threshold is used to skip frames. Moreover, besides the target, the analytical framework modeling the H.264 video source allows us to choose how many P-frames to encode between two I-frames, and the number of B-frames to encode, in order to minimize the skipping probability.

The paper is structured as follows. Section 2 presents the H.264 encoding system we model in the paper. Then, Section 3 introduces a statistical analysis of H.264 traces. In Section 4 the H.264 video encoding system is modeled as a switched batch Bernoulli process (SBBP). Section 5 applies the model to a case study and derives numerical results and important hints for the encoding system designer to efficiently dimension the encoding parameters. Section 6 completes the model with derivation of the main performance parameters. Finally, the authors’ conclusions are drawn in Section 7.

2 H.264 video transmission system
In this section we will describe the H.264 video transmission system we model in the rest of the paper. As represented in Fig. 1, it is made up of a Video Source whose output is sent to the H.264 encoder. The bits emitted by the encoder are sent to the Packetizer device, which prepares the video stream to be sent over the Internet, segmenting the output flow according to the UDP/IP protocol stack; each segment will constitute the payload of a UDP packet. The generated segments are then queued in the Output Buffer, before being sent over the network at a packet rate of \( R_p = R/F \) packets per frame interval, where \( R \) and \( F \) are the channel packet rate and the source frame rate, respectively. The Output Buffer works as a smoother in order to eliminate the burstiness which is typical of any encoded video stream. Of course, if the system is not well dimensioned, the presence of the Output Buffer may cause buffering delay and packet losses. Another fundamental block in the video system is the Encoding Mode Controller device. Monitoring the state of the Output Buffer queue, \( s_0 \), and the activity of the frame to be encoded, \( a \), it controls the encoding process. More specifically, this device:
1. assures that the number of B-frames encoded between two successive reference frames (I- or P-frames) is equal to a given configuration parameter \( b \); in other words, it assures that the minimum distance between two successive reference frames, not considering skipped frames, is equal to \( b + 1 \);
2. imposes the periodical presence of I-frames in the output stream by setting the number of P-frames between two successive I-frames to \( p \); so the distance between two successive I-frames, in terms of encoded frames between them, is equal to \( D_s = (b + 1) \cdot (p + 1) \);
3. skips frames to avoid Output Buffer queue saturation. The third target is achieved by the Encoding Mode Controller, by monitoring the Output Buffer queue. For this purpose, we will distinguish between two different kinds of H.264 video source, according to the behavior of the Encoding Mode Controller:
Open-loop H.264 video source, when the Encoding Mode Controller only meets the first and the second of the above targets, that is, it only works to alternate the encoding mode between I, P and B; Closed-loop H.264 video source, when the Encoding Mode Controller receives feedback from the Output Buffer in order to meet all the above targets. In this case it can alternate the encoding mode between I, P and B, or skip some frames.

3 Statistical analysis of H.264
In this section we will discuss the statistical properties of an H.264 data flow. These properties will be used in Section 4 to find a suitable model for H.264 video sources. An H.264 encoder generates an emission process influenced by both the activity and the frame-encoding mode. According
to what has been said so far, the possible encoding modes are \( I, P, \) or \( B \). However, given the possibility of skipping some frames, we consider an additional coding mode, the \( S \) mode, which relates to skipped frames and is characterized by zero emissions.

The H.264 emission process can be described by the activity process and the activity/emission relationships. The activity process only depends on the characteristics of the frame being encoded; the activity/emission relationships represent the number of bits which the encoder generates during the encoding of a picture, and are characterized by both the activity and the frame-encoding mode used. In the next sections we will first characterize the activity process (Section 3.1) and then the activity/emission relationships (Section 3.2). In order to show these processes, we will consider the trace obtained from one hour of H.264 encoded video sequences of the movie “Evita”, using the open-loop scheme, that is, with no rate control.

### 3.1 Characterization of the activity process

As already said, the activity of a frame only depends on the peculiarities of the picture itself. The activity of the macroblock \( h \) in frame \( n \), \( L_h(n) \), can be derived from the variance of the four luminance blocks within this macroblock, as follows:

\[
L_h(n) = 1 + \min\{\sigma_{i,3}^2(n), \sigma_{i,4}^2(n), \sigma_{i,1}^2(n), \sigma_{i,2}^2(n)\}, \quad i \in \{1,2,3,4\}
\]

where \( \sigma_{i,k}^2(n) \), with \( i \in \{1,2,3,4\} \), represents the variance of luminance of the \( i \)-th block within the macroblock \( h \) in the generic frame \( n \). The discrete-time activity process \( L(n) \) can be defined as the average value of the activities in the macroblocks within the frame \( n \):

\[
L(n) = \frac{1}{D} \sum_{h} L_h(n)
\]

where \( D \) is the number of macroblocks within a frame. In [6][7] it was demonstrated that the activity process can be represented by its first- and second-order statistics. Let us stress that it was demonstrated for MPEG video traffic, but it is valid for H.264 video traffic as well, given that the activity process behavior does not depend on the particular encoding technique. In Section 5 we will use a Markov-modulated process to model the activity of H.264 video traffic.

### 3.2 Characterization of the activity/emission relationships

While the activity process does not depend on the encoding mode, the emission process does. For each encoding mode \( \varsigma \in \{I, P, B, S\} \) we define the activity/emission relationships as the distribution of the sizes of the frames of this encoding mode, expressed in bits, once the activity \( a \) of the same frame is given:

\[
y_c(a | \varsigma) = \Pr[V(n) = r | L(n) = a, J(n) = \varsigma]
\]

\[
\forall \varsigma \in \{I, P, B, S\}
\]

where:

\( V(n) \) is the emission process, representing the number of bits used to encode the generic frame \( n \), \( L(n) \) is the activity process, calculated as in (2) for the generic frame \( n \) and \( J(n) \) is the encoding mode process, representing the encoding mode used to encode the generic frame \( n \). In [6][7] it was demonstrated that all the random values defined in (3), for each value of \( \varsigma \in \{I, P, B\} \) and \( a \), are Gamma-shaped, with a mean value linearly increasing with the activity, and the variance linked to the activity with a parabolic law. So we can say that, for each activity \( a \) and for a given encoding mode \( \varsigma \in \{I, P, B\} \), the activity/emission relationships can be exhaustively described by the mean value and variance of the \( I-, P- \) and \( B\)-frame emission processes, indicated here as \( \mu_{\varsigma|a} \) and \( \sigma_{\varsigma|a}^2 \), respectively.

### 4 System model

In this section we define a Switched Batch Bernoulli Process (SBBP) matching the statistical characteristics of the H.264 video traffic outlined in the previous section. First of all, in Section 4.1 we define an SBBP model. Then, in Section 4.2 we present our methodology to calculate the SBBP process modeling an H.264 source. To this end we will use a discrete-time approach, and the slot duration, \( \Delta \), is set to the frame interval duration, that is, \( \Delta = 1/F \).

#### 4.1 Switched Batch Bernoulli Process (SBBP)

A Switched Batch Bernoulli Process is a discrete-time emission process modulated by an underlying Markov chain, with a finite number of states [4-5]. Each state of the Markov chain is characterized by an emission probability density function (pdf): the SBBP emits data according to the pdf of the current state of the underlying Markov chain. Therefore an SBBP is fully characterized by the state space \( \mathcal{S}^{(I)} \) of the underlying Markov chain, and by two matrices:

1. the transition probability matrix of the underlying Markov chain, \( Q^{(Y)} \), which is a square matrix having a dimension equal to the number of states of the underlying Markov chain. Let \( \mathcal{S}^{(Y)}(n) \) represent the state of the underlying Markov chain in the generic slot \( n \). The generic element of the matrix \( Q^{(Y)} \) is defined as follows:

\[
Q_{[s_i,s_f]}^{(Y)} = \Pr\{S^{(Y)}(n+1) = s_f | S^{(Y)}(n) = s_i \} \quad \forall s_i, s_f \in \mathcal{S}^{(Y)}
\]

2. the emission probability matrix, \( B^{(Y)} \), whose rows contain the emission pdfs for each state of the underlying Markov chain. The generic element of the matrix \( B^{(Y)} \) is defined as follows:
\[ B_{[r,r',s]}^{(Y)} = \text{Prob}\{Y(n) = r \mid S(n) = s_r \} \quad \forall s_r \in S^{(r)}, \forall r \in \left[0, r_{\text{MAX}}^{(Y)} \right] \] (5)

From \( Q^{(r)} \) we can obtain the steady-state probability array of the underlying Markov chain of the SBBP \( Y(n) \), \( \pi^{(r)} \), by solving the following linear system:

\[
\begin{align*}
\pi^{(r)} \cdot Q^{(r)} &= \pi^{(r)} \\
\sum_{s_r} \pi^{(r)}|n\rangle &= 1
\end{align*}
\] (6)

Below we will use an extension of the meaning of the SBBP to model not only an emission process, but also the activity process which characterizes the video sequence.

4.2 Source model

The model we introduce in this section has to capture the statistical behavior of the closed-loop H.264 encoding system, that is, both the activity process behavior and the activity/emission relationships. More specifically, after identifying a set of possible activity states, \( \mathcal{S}^{(a)} \), each representing one possible level of scene activity, both transitions between activity states and transitions between one frame and the successive one have to be modeled simultaneously.

We will obtain the model of the closed-loop H.264 encoding system in two steps:

1. derivation of an activity SBBP, \( A(n) \), modeling the activity process \( L(n) \) of the movie;
2. derivation of the SBBP \( Y(n) \), modeling the emission process of the closed-loop H.264 system; it will be obtained by calculating its underlying Markov chain from the underlying Markov chain of the activity SBBP \( A(n) \), and its emission process from the activity/emission relationships.

1) First step: modeling video sequence activity

As discussed in Section 3.1, the activity process, \( L(n) \), of an H.264 video source can be characterized by its first- and second-order statistics, \( f_1(a) \) and \( C_2(m) \). Therefore, the target is to calculate the activity SBBP, \( A(n) \), defined by the parameter set \( (Q^{(a)}, B^{(a)}) \), which is able to fit these statistics. As discussed in Section 3, this process does not depend on the particular encoding technique used to encode the movie, but only on the intrinsic characteristics of the movie. It is based on the solution of the so-called inverse eigenvalue problem in the discrete-time domain. Let us indicate the state space of the underlying Markov chain of the activity process as \( \mathcal{S}^{(a)} \). It represents the set of activity levels to be captured.

The activity SBBP \( A(n) \) is defined by the parameter set \( (Q^{(a)}, B^{(a)}) \), where \( Q^{(a)} \) is the transition matrix among the states in \( \mathcal{S}^{(a)} \), as is customary, whereas \( B^{(a)} \) is an “activity probability matrix”, which is defined as follows: when the underlying Markov chain of \( A(n) \) is in the state representing the activity level \( i \), the activity process takes values according to the probabilities in the \( i \)-th row of the activity probability matrix \( B^{(a)} \).

2) Second step: modeling H.264 emission process

Once the activity SBBP \( A(n) \) modeling the activity process has been obtained, we can derive the SBBP \( Y(n) \) modeling the H.264 emission process. Let us define the state of the underlying Markov chain of \( Y(n) \) as follows:

\[
S^{(r)}(n) = (S^{(a)}(n), S^{(b)}(n), S^{(r')}(n))
\] (7)

where:

- \( S^{(a)}(n) \in \mathcal{S}^{(a)} \) is the state of the underlying Markov chain of \( A(n) \);
- \( S^{(b)}(n) \in \{0, \ldots, K \} \) is the output buffer queue state, defined as the number of packets in the queue and in the server at the generic slot \( n \);
- \( S^{(r')}(n) \in \{0, \ldots, D_r - 1\} \) is the frame position in the GoP (Group of Pictures) at slot \( n \), that is, the number of frames encoded since the last \( I \)-frame; \( D_r \) is the distance between two successive \( I \)-frames, in terms of the encoded frames between them. Let \( s^{(a)}_r, s^{(b)}_r, s^{(r')}_r \) represent the states of the processes \( S^{(a)}(n), S^{(b)}(n), S^{(r')}(n) \) in the generic slot \( n \), and let \( s^{(a)}_{r'}, s^{(b)}_{r'}, s^{(r')}_{r'} \) represent the states of the same processes in the successive slot \( n + 1 \).

The order of events in each slot is assumed to be as follows:

1. the new state of the activity process, \( s^{(a)}_{r'} \), is calculated according to the transition probability matrix of the activity SBBP \( A(n) \), \( Q^{(a)} \), derived in the first step;
2. the new frame encoding mode, \( j^{*} \in \{I,P,B,S\} \), is calculated according to the values of \( s^{(a)}_{r'}, s^{(b)}_{r'} \), according to the rules which will be shown in (9);
3. the state of the process \( S^{(r')}(n) \) is updated according to its previous state and the frame encoding mode \( j^{*} \), according to (11); the new value of the emission process is calculated according to the new state of the activity process, \( s^{(a)}_{r'} \), and the frame encoding mode selected, \( j^{*} \), according to the emission probability matrix in (10);
4. finally, the new output buffer state, \( s^{(b)}_{r'} \), is calculated according to (12).

With this in mind, the transition probability matrix of the underlying Markov chain, \( Q^{(r')} \), can be obtained as follows:

\[
Q^{(r')}_{[r,r',s]} = \begin{cases} 
\sum_{j^{*}} Q^{(a)}_{[s^{(a)}_{r'},j^{*}]} B^{(a)}_{[j^{*},j^{*}]} & \text{if } C^{(\text{FramePosition})}_{[r,r',j^{*}]} \text{ and } C^{(\text{Buffer})}_{[s^{(b)}_{r'},s^{(b)}_{r'}]} \\
0 & \text{otherwise}
\end{cases}
\] (8)

where:

- \( j^{*} \in \{I,P,B,S\} \) represents the frame encoding mode in the slot \( n + 1 \), and depends on \( s^{(a)}_{r'} \) and \( s^{(b)}_{r} \) as follows:
The state of the underlying activity process Markov chain is

\begin{align*}
  \text{Controller algorithm described in Section 2, taking into}
  \text{account. These conditions can be expressed as}
  \begin{cases}
    S & \text{if } b \leq s'_r \leq D_2 -1 \text{ and } s'_r - b \times (b + 1) \text{ and } s'_q > M \\
    I & \text{if } s'_r = (p + 1) \cdot b + p \text{ and } s'_q \leq M \\
    P & \text{if } b \leq s'_r \leq D_1 - (b + 2) \text{ and } s'_r - b \times (b + 1) \text{ and } s'_q \leq M \\
    B & \text{otherwise}
  \end{cases}
\end{align*}

(9)

\( r \) is the number of emitted packets;
\( Q_{[a',\infty)} \) is the transition probability matrix of the underlying Markov chain of the activity process;

Equation (9) was derived according to the Encoding Mode Controller algorithm described in Section 2, taking into account:

a) a frame is an S-frame if and only if the next frame to be encoded is a reference frame and the variable \( s'_r \) is greater than the skip threshold, \( M \);

b) a frame is encoded as an I-frame if and only if \( p \) P-frames and \( (p + 1) \cdot b \) B-frames have already been encoded after the last I-frame, and the buffer queue length is less than or equal to the skip threshold;

c) a frame is encoded as a B-frame if and only if the frame to be encoded is a reference frame but not all the \( p \) P-frames have already been encoded since the last I-frame, and the variable \( s'_r \) is less than or equal to the skip threshold;

d) a frame is encoded as a B-frame if and only if all the above conditions are false;

The probability of using \( r \) packets to encode the frame \( j^* \) when the state of the underlying activity process Markov chain is \( s''_a \in \mathcal{Z}^{(a)} \), is:

\begin{align*}
  B_{[a',\infty)}(r | a) = \begin{cases}
    \sum_{a=0}^{a'} y^{(I)}(r | a) \cdot B^{(a)}_{[a',a]} & \text{if } j^* = I \\
    \sum_{a=0}^{a'} y^{(P)}(r | a) \cdot B^{(a)}_{[a',a]} & \text{if } j^* = P \\
    \sum_{a=0}^{a'} y^{(B)}(r | a) \cdot B^{(a)}_{[a',a]} & \text{if } j^* = B \\
    \sum_{a=0}^{a'} y^{(S)}(r | a) \cdot B^{(a)}_{[a',a]} & \text{if } j^* = S
  \end{cases}
\end{align*}

(10)

where \( y^{(I)}(r | a) \), \( y^{(P)}(r | a) \), \( y^{(B)}(r | a) \) and \( y^{(S)}(r | a) \) are the functions introduced in (3) to characterize the activity/emission relationships, while \( B^{(a)}_{[a',a]} \) is the probability that the activity is

\( a \in [0, a_{\text{max}}^{(c)}] \) when the activity level is \( s''_a \in \mathcal{Z}^{(a)} \), and can be obtained as discussed in the first step.

Moreover, in (10) we have indicated:

- \( C^{(\text{FramePosition})}_{[a',\infty)} \): the Boolean evolution condition for the process \( S^{(a)}(n) \) representing the frame position in the GoP;
- \( C^{(\text{Buffering})}_{[a',\infty)} \): the Boolean evolution condition for the buffer queue length process \( S^{(0)}(n) \).

Let us note that the above conditions are deterministic, given that the only processes in our model which present stochastic behavior are the activity process and emission process, while the processes \( S^{(a)}(n) \) and \( S^{(0)}(n) \) evolve deterministically once the new values \( s''_a \), \( j^* \) and \( r \) are known. In the following subsections we will analyze each of these terms in detail.

3) \( C^{(\text{FramePosition})}_{[a',\infty)} \): evolution for the frame position process

The evolution of the process \( S^{(r)}(n) \), representing the frame position in a GoP, is deterministic. In fact, its state in a generic slot depends only on its previous state and on the frame-encoding mode, \( j^* \). The state of this process is incremented each time a frame is encoded, while it is not changed when a frame is skipped. Therefore we have:

\begin{align*}
  C^{(\text{FramePosition})}_{[a',\infty)} = \begin{cases}
    \text{TRUE} & \text{if } \begin{cases}
      (j^* = I \text{ or } j^* = P \text{ or } j^* = B) \\
      \text{and } s''_r = (s'_r + 1) \text{mod } D_2
    \end{cases} \\
    \text{FALSE} & \text{otherwise}
  \end{cases}
\end{align*}

(11)

4) \( C^{(\text{Buffering})}_{[a',\infty)} \): the evolution condition for the buffer queue length process

The new buffer queue length state is calculated each time a new frame is encoded. Therefore the state of the process \( S^{(0)}(n) \) depends on its previous state and on the number of emissions. The state of the process \( S^{(0)}(n) \) is simply updated by adding to its previous state the number of emissions, \( r \), and subtracting the maximum number of packets which can be served by the channel in a slot, \( R_s \). In addition, the fact that it cannot assume values less than 0 or greater than \( K \) has to be taken into account. These conditions can be expressed as follows:

The emission pdf for I-frames, P-frames and B-frames is shown in Fig. 2. The emission pdf for I-frames is given by a normal distribution centered at 0 and with a standard deviation of 10. The emission pdf for P-frames is given by a normal distribution centered at 20 and with a standard deviation of 5. The emission pdf for B-frames is given by a normal distribution centered at 30 and with a standard deviation of 7.
5 Performance evaluation

In this section we show how the model defined above can be used to evaluate the performance of the H.264 encoder system illustrated in Section 2. It is very important to study how the skipping probability of the H.264 encoder system is influenced by system parameters such as the output buffer service rate $R_s$, the number of successive $B$-frames, $b$, and the number of $P$-frames, $p$, between two successive $I$-frames and how the output buffer queue length is influenced by these system parameters. All these statistics can be calculated from the transition probability matrix, $Q^{(r)}$, or from the steady-state probability array $\pi^{(s)}$, whose generic element is defined as follows:

$$
\pi^{(s)}_{i,j} = \text{Prob}\{S^{(s)}(n) = j \mid s_i \}
$$

In Section 5.1 we will present the expressions which allow us to calculate the buffer queue length statistics, such as its pdf and its mean value and variance. Through these statistics, in fact, it is possible to understand how the configuration parameters, $R_s$, $p$ and $b$, influence the buffer queue length, which is responsible for an increase or decrease in the transmission delay and the number of skipped frames.

In Section 5.2 we will show how to calculate the pdf of the number of successive skipped frames and the skipping probability.

5.1 Output buffer queue length statistics

In this section we calculate the statistics of the output buffer state. To this end, we derive the pdf of the output buffer queue length as follows:

$$
f_{\sigma}(s_o) = \text{Prob}\{S^{(o)}(n) = s_o \} = \sum_{v_{s_i,S^{(o)}(n)}=s_o} \sum_{t_{s_i,S^{(o)}(n)}=t_o} \pi^{(e)}_{s_i,S^{(o)}(n),t_{s_i,S^{(o)}(n)}}
$$

From this pdf it is possible to compute the mean value and variance of the buffer queue length, by using the following well-known expressions:

$$
E[s_o] = \sum_{s_o} s_o \cdot f_{\sigma}(s_o)
$$

$$
Var[s_o] = \sum_{s_o} s_o \cdot f_{\sigma}(s_o) - \left( \sum_{s_o} s_o \cdot f_{\sigma}(s_o) \right)^2
$$

5.2 Frame skipping statistics

The pdf of the number of successive skipped frames between two reference frames can be calculated as follows:

$$
f_{\sigma_{sk}}(s_{sk}) = \text{Prob}\{\sigma_{sk} = s_{sk} \} = \text{Prob}\{J(n+1) = S, J(n) = S, \ldots, J(n + s_{sk} - 1) = S\}
$$

where $\sigma_{sk}$ is the random variable which counts the number of successive skipped frames, and can take values in the range $[0, \text{MaxSkip}]$, where

$$
\text{MaxSkip} = \left[ \frac{K - M}{R_s} \right]
$$

$\left[ x \right]$ being the minimum integer no lower than $x$.

Let $J(n)$ be the frame encoding mode process whose value depends on the frame position in the GoP, $s_r$, and on the output buffer queue state, $s_o$, according to (9).

Let us indicate the state space, made up of the states of $\mathcal{Z}^{(r)}$ causing frame skipping, as $\mathcal{Z}^{(s)}$. Let

$$
\mathcal{Z}^{(s)} = \{s_r, s_o, s_{sk}\} such that \lambda(s_r, s_{sk}) = S
$$

The complementary state space to $\mathcal{Z}^{(s)}$, with respect to the space state $\mathcal{Z}^{(r)}$, will be indicated as $\mathcal{Z}^{(enc)}$, representing the set of states of $\mathcal{Z}^{(r)}$ which determine the encoding of a frame:

$$
\mathcal{Z}^{(enc)} = \mathcal{Z}^{(r)} \setminus \mathcal{Z}^{(s)}
$$

Let us denote the matrix of system transitions towards a skipping state as $\mathcal{Q}^{(r)}_{\text{sk}}$; it can be calculated from $\mathcal{Q}^{(r)}$ by setting all the probabilities of transition towards a non-skipping state to zero. Likewise, let us denote the matrix of system transitions towards an encoding state as $\mathcal{Q}^{(r)}_{\text{enc}}$; it can be obtained from $\mathcal{Q}^{(r)}$ by setting the probabilities of transition toward a skipping state to zero.

We have:

$$
\mathcal{Q}^{(r)}_{\text{sk}}(s_{sk}) = \begin{cases} 
\mathcal{Q}^{(r)}(s_{sk}) & \text{if } s_{sk} \in \mathcal{Z}^{(s)} \\
0 & \text{otherwise}
\end{cases}
$$

and $\mathcal{Q}^{(r)}_{\text{enc}} = \mathcal{Q}^{(r)} - \mathcal{Q}^{(r)}_{\text{sk}}$. So equation (19) becomes:

$$
f_{\sigma_{sk}}(s_{sk}) = \pi^{(s)}_{(r,s)} \cdot \mathcal{Q}^{(r)}_{\text{sk}}(s_{sk}) = \pi^{(s)}_{(r,s)} \cdot \mathcal{Q}^{(r)}_{\text{enc}}(s_{sk}) \cdot \mathcal{P}^{(s)}_{(r,s)}
$$

where $\mathcal{P}^{(s)}_{(r,s)}$ is the steady-state probability array in the first slot of a skipping period. It can be calculated as follows:
The denominator in (24) is needed to make the sum of the resulting array elements equal to 1, transforming the conditional probability into a joint probability. At this point we have calculated the pdf of the random variable \( \sigma_{x_k} \). Now it is easy to calculate its mean value. We have:

\[
E[\sigma_{x_k}] = \hat{\pi}^{(r,\text{ENC})} \cdot Q^{(I)}_{\rightarrow 2} \cdot 1 \quad (25)
\]

The variance of the number of skipped frames is computed using the following well-known expression:

\[
\text{Var}[\sigma_{x_k}] = \sum_{s_{x_k}=0}^{\text{MaxSkip}} s_{x_k} \cdot f_{x_k}(s_{x_k}) \left( \sum_{s_{x_k}=0}^{\text{MaxSkip}} s_{x_k} \cdot f_{x_k}(s_{x_k}) \right)^2 \quad (26)
\]

Finally, another important statistic is the skipping frame probability, which can be calculated by using the following equation:

\[
\text{Prob}[S] = \text{Prob}[J(n) = S] = \sum_{\forall i,j \in \{0,1,2,4\}} \sum_{\forall \ell} \pi_{(r)}^{(I)} \cdot Q^{(I)}_{\rightarrow 2} \quad (27)
\]

### 6 Numerical results

In this section we apply the analytical paradigm introduced in the paper to the encoding and transmission of the movie “Evita”, chosen as a case study. Its encoding characteristics were described at the beginning of Section 3. The size of the Output Buffer was set to 60 packets. For transmission, we used a packet size of 604 bytes, where 28 bytes constitute the packet header, and 576 bytes the payload.

In order to study how the the project parameters, \( R_v \), \( p \) and \( b \), influence the skipping process, these three configuration parameters were varied in the ranges \([6, 21], [1, 4]\) and \([0, 4]\), respectively. Fig. 2 shows the pdf of the number of emissions for each kind of frame. In Section 6.1 we will apply the proposed model to analyze how the most important configuration parameters \( p \), \( b \) and \( R_v \) influence the performance of the encoding system.

#### 6.1 Model applied to video encoder performance analysis

Figs. 3.a and 3.b represent the skipping probability vs. \( R_v \), calculated according to (27), with two different values of \( p \): \( p = 1 \) and \( p = 4 \). Each curve in these figures differs from the others in the number \( b \) of encoded B-frames between two successive reference frames. Although the cases \( p = 2 \) and \( p = 3 \) were also studied, the figures obtained were very similar to the ones presented here, and hence they are not shown in this paper. With low \( R_v \) values (less than 9 packets/sec in this case), the skipping probability is greater for higher values of the project parameter \( b \) than for lower ones; instead, we obtain the opposite result with higher \( R_v \) values. This happens because with low \( R_v \) values, B-frames are encoded with a number of packets greater than \( R_v \), leading to an increase in the buffer queue length, as is possible to note from (2), and hence in the skipping probability. On the contrary, with greater \( R_v \) values, B-frames are encoded with a number of packets less than \( R_v \); in this case the number of packets which leave the output buffer in each slot is greater than the number of packets emitted in the same slot, and the skipping probability decreases with the increase in the number of successive B-frames encoded between two reference frames. With very large \( R_v \) values, all these curves tend to zero.

### 7 Conclusions

The paper deals with the problem of transmitting H.264 video traffic on the Internet. To this end, after a statistical analysis of H.264 video flows, a methodology to construct a Markov-based analytical framework to model an H.264 video encoder was introduced. The proposed paradigm is applied to a case study to demonstrate how it can be used to analyze the performance of the proposed encoding system and to achieve some guidelines to design its main parameters. The proposed paradigm allows the designer to choose the best service rate to be negotiated with the network, and the best GoP structure for an H.264 video stream in order to minimize both the number of frames the encoder has to skip, and the mean Output buffer delay.

### References


