University of Padova

Department of Information Engineering

Master Degree in Telecommunications Engineering

Source video rate allocation and scheduling policy design for wireless networks

Candidate

Massimiliano Pesce

Supervisor

Prof. Michele Zorzi

Co-supervisor

Dr. Daniele Munaretto

Academic Year 2013/2014

Alla mia famiglia

Prediction is very difficult,
especially about the future.
Niels Bohr (attributed)

Contents

List of Figures

Abstract

The increasing availability of smart devices that allow the user to instantaneously share contents in the Internet, leads to an ever growing amount of data produced and consumed by end users. With the advent of new content sharing social platforms, i.e., Facebook, Twitter, Instagram, Youtube, etc., telecommunications researchers are asked to find new techniques to satisfy the content delivery requests in an efficient manner. In this work we address the problem of designing an efficient algorithm that allows high quality real time video streaming, from the source coding and scheduling perspectives. We take into account some key quantities like frame correlation, channel constraints and *Quality of Service* (QoS) metrics such as distortion and *Peak Signal-to-Noise Ratio* (PSNR). We first formalize the problem in a mathematical fashion and then propose a Markov-chain based solution that applies to quantized values of the metrics involved in the framework, achieving good performance while maintaining the complexity of the framework low. We compare the simulation results for different scenarios and we conclude the thesis by proposing a trade-off solution between performance and complexity.

Sommario

La crescente diffusione di dispositivi mobili atti alla condivisione di contenuti multimediali sul web, ha portato ad un importante incremento del traffico dati. Con l'avvento di nuove piattaforme di condivisione come Facebook, Twitter, Instagram, Youtube, ecc., il settore delle telecomunicazioni è alla ricerca di nuove tecniche in grado di soddisfare efficientemente le richieste degli utenti. In questo lavoro siamo alla ricerca di un algoritmo efficiente capace di garantire buone prestazioni nella trasmissione video in tempo reale, agendo ai livelli di codifica e scheduling. Nel fare ciò, saranno tenuti in considerazione parametri tecnici rilevanti come la correlazione tra immagini del video, i vincoli di capacità imposti dal canale e metriche di valutazione della *Qualità del Servizio* (dall'inglese Quality of Service QoS) come distorsione e *rapporto picco segnale-rumore* (dall'inglese PSNR). Inizialmente formalizzeremo il problema a livello matematico e in seguito proporremo una soluzione semplificata basata sui processi Markoviani in grado di ottenere buone prestazioni e complessità ridotte. Confronteremo i risultati per scenari diversi e infine proporremo un compromesso tra prestazioni e complessità.

Ringraziamenti

Desidero ringraziare tutti coloro che mi hanno aiutato nella stesura della tesi con suggerimenti, critiche ed osservazioni: a loro va la mia gratitudine, anche se a me spetta la responsabilità per ogni errore contenuto in questa tesi. Ringrazio anzitutto il prof. Michele Zorzi, relatore, e il dott. Daniele Munaretto, co-relatore: senza il loro supporto e la loro guida sapiente questa tesi non esisterebbe. Un ringraziamento va anche alla dott.ssa Toni Laura per il prezioso contributo.

Chapter 1

Introduction

"Big brother is watching you" [1] is the famous phrase reminded to people living in the dystopic George Orwell's novel, *1984*, where one person was able to observe everyone using cameras placed everywhere. Maybe in 1948, when the novel was written, nobody could imagine that this fantastic world, not many years later, will have been easily feasible. Obviously, at that time, this was only product of fantasy, but let us pretend to be in that world and raise a question: how can the Big Brother manage all data that come from cameras in real-time? How should its coding and scheduling algorithms work? Nowadays this situation is fairly practical, people are used to shoot videos with their portable devices and upload them on their favourite social network, thus generating huge quantities of data flowing through the Internet. In the past years, mobile networks have seen an exponential increase in video traffic which has grown till exceeding 50% of the total shared data [2], as shown in Figure 1.1, and it seems to continue with the same trend or even faster. It is estimated that the mobile network capacity will need to be increased 65 times to scale with the expected traffic, between 2013 and 2018.

Video Aware Wireless Networks (VAWN) [3] is a multy-year project that supports our claims. Wireless networks must keep pace with these requests by exploiting new technologies and algorithms that have to be designed to fit the specific characteristics of the video stream in order to be able to better deal with this huge amount of data. Current mobile networks face issues when providing a reliable service to video streaming [4]. It requires a

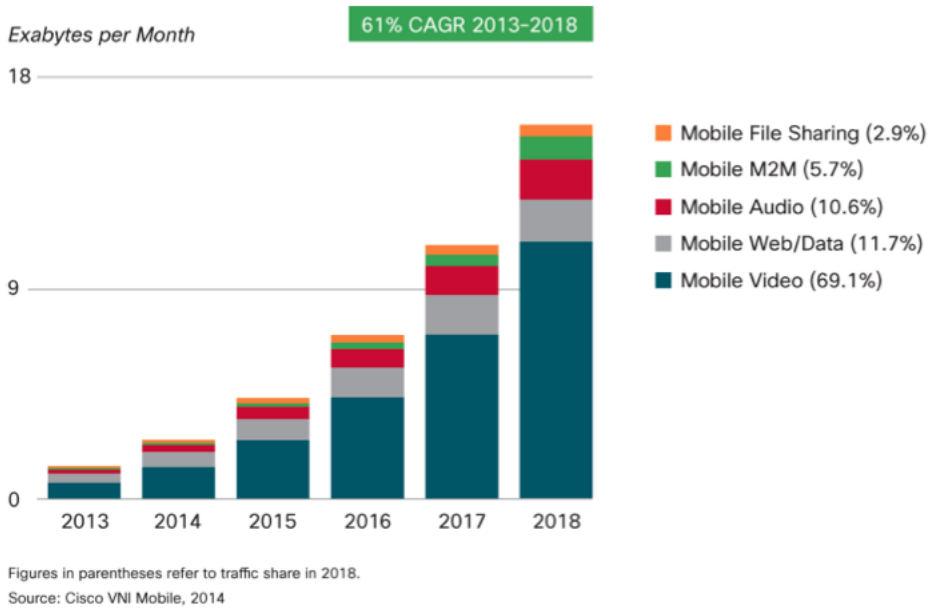

Figure 1.1: Cisco mobile traffic forecast.

steady flow of information and delivery of packets within a target deadline to fulfill quality and delay contraints [5]. Wireless networks have to deal with a multipath fading channel which is variable in time and must be shared among multiple users that interfere with each other. Shadowing and pathloss might further increase link variability leading to unfavorable conditions for video streaming. The problem of finding a solution to send efficiently, i.e., with an acceptable QoS, multiple video flows over a wireless channel in a scenario where user requests grow exponentially is a challenging topic. In this work[1] we focus our study on the real time upstreaming scenarios, where a mobile user shoots a video clip from a camera, e.g., on his helmet when skiing, or from the smartphone when traveling (as in Figure 1.2), and upstreams the video to the Internet to share the content with friends on social networks, e.g., Facebook, Twitter, and so on. The encoding rate to generate the video frames and the scheduling policy for the transmission at the wireless interface of the mobile device should be decided in order to guarantee a target expected video quality at the consumer side. The complexity of making such decisions

[1]Part of this work was published in [6]

increases when jointly taking into account the mobility of the source, the dynamics of the scene and the channel variations. In this thesis, we consider the inherent frame correlation of the video stream, due to the dynamics of both source and scene, to properly select the encoding rate of the video frames generated and the associated scheduling policy, under the constraint of the time-varying channel conditions. We design and implement a *Markov Decision Process* (MDP) model where the decisions on the encoding rate of the video frames and on the scheduling policy are made with the goal of minimizing the distortion of the video upstreamed to the Internet, and there made available for public video consumption.

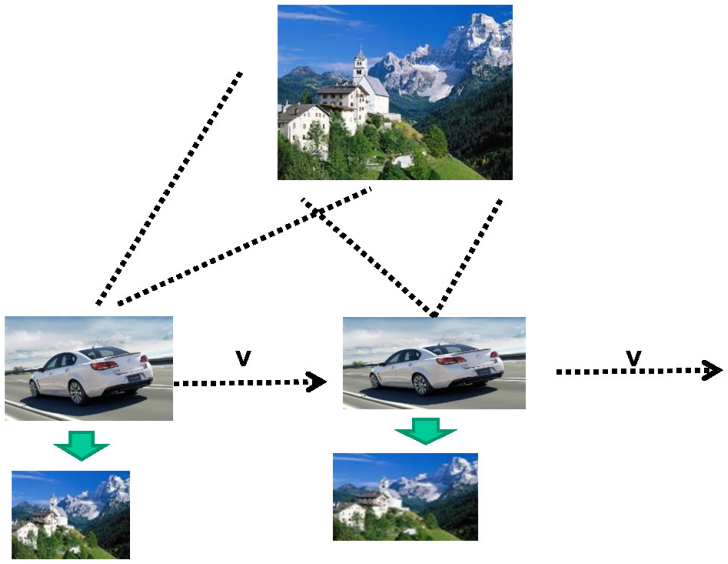

Figure 1.2: Reference scenario: mobile video upstreaming.

Mapping the mobility of the user and the dynamics of the scene to the frame correlation information is challenging due to the independent and simultaneous effect of the two sources of mobility on the video frames, which has not been yet jointly addressed. However, assuming to know the resulting frame correlation, a predictive video encoder uses the information about the correlation between consecutive frames to select the necessary video encoding rate, say, within a given *Group of Picture* (GoP), to keep the video distortion to a target level. Furthermore, the optimal choice of such rate has an

impact on the packet scheduling scheme to be adopted and is constrained by the channel state conditions and service requirements in terms of expected provisioned quality. Thus, the purpose of this work is to design an MDP model to foresee the encoding rate and scheduling policy at the video producer side that minimizes the average long-term distortion of a sequence of video frames, e.g., a GoP, given the frame correlation information and the wireless channel conditions. The thesis is structured as follows:

- Chapter 2: a technical overview of mobile video streaming is provided with focus on standard video encoding schemes and video quality metrics;
- Chapter 3: we discuss some prior work related to this thesis;
- Chapter 4: we formulate a mathematical model and provide an analytical solution;
- Chapter 5: we propose a Markov-based solution and report simulation results;
- Chapter 6: we draw our conclusions and we discuss some future directions.

Chapter 2

Mobile video streaming

Video streaming is the traffic source that generates the largest amount of data in the Internet [7]. Differently from audio and image signals that require hundreds of bytes, one second of *Comité Consultatif International pour la Radio* (CCIR) 601 [8] video signal at a rate of 30 frames per second is about 20 Mbytes. Since videos can be seen as sequences of images in a temporal coherent order, one approach to compression is to compress video frames like images. However, there are limitations to this approach: human beings do not perceive motion video in the same way they perceive still images. The motion in videos can mask coding artifacts that would be visible in images and, therefore, compression techniques may exploit these features as advantages to reduce the bitrate. Thanks to its network-friendly inclination, H.264 *Advanced Video Coding* (AVC) is the intended standar for high-definition video coding over mobile networks and it is considered in this work. In particular we consider real-time video upstreaming applications. The mobility of the source with the dynamic of the shot scene are foundamental to estimate the correlation between frames. In order to improve the efficiency of the coding process, a study of the correlation model is required. The better the accuracy of the correlation model, the more the insights we gain into the impact of the correlation on the rate-to-quality behavior of videos.

2.1 Video coding: H.264/AVC standard

H.264/AVC is the result of continuous enhancements of a series of video standards. Formalized with the final draft in December of 2001 by the *Joint Video Team* (JVT)[1] and then approved in 2003 by ITU-T as Recommendation H.264 and by ISO/IEC as International Standard 14496–10 (MPEG-4 part10) AVC, it has evolved through the development of the ITU-T H.261, H.262(MPEG-2), and H.263 video coding standards [9]. The standard is designed for technical solutions including video-on-demand or multimedia streaming services over wireless networks. The basic block diagram is reported in Figure 2.2 [9].

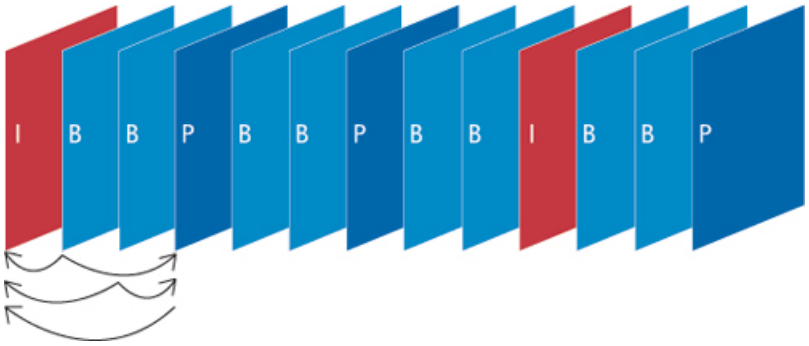

Figure 2.1: A possible arrangement of a typical GoP structure.

The Video Coding Layer (VCL) [9] works dividing the frames into inter and intra (I) pictures. The former are frames obtained using some prediction on the neighbor frames, while the latter are coded independently of the other frames. The inter pictures frames can be divided into two sets: the *predictive coded* (P) and the *bidirectional predictive coded* (B). The inter pictures are obtained by subtracting a motion-compensated prediction [7] from the original source as described in Section 2.1.1; then the residuals are transformed in into the frequency domain by a *Discrete Cosine Transform* (DCT). The transform coefficients are, therefore, scanned, quantized and coded using varible-length codes. Moreover, a local decoder is employed to reconstruct

[1] *Video Coding Expert Group* and *Motion Video Expert Group* joined together in December 2001 to form the *Joint Video Team* (JVT)

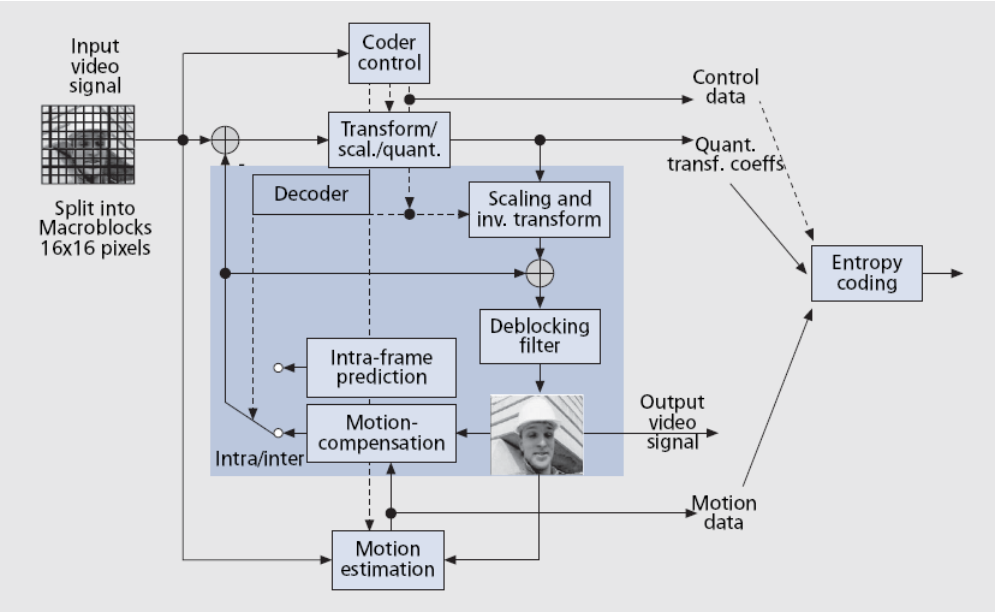

Figure 2.2: Block diagram of the ITU-T H.264/AVC encoder.

the frame for use in future predictions. Intra pictures, instead, are coded without reference to past pictures. Details differentiate H.264/AVC from its predecessors. The decorrelation process includes motion-compensated prediction and transformation of the prediction error for the inter frames, while, for the intra frames, it includes intra prediction modes and transforms used in this mode. The macroblock of the encoder, as aforementioned, is the same used in the other standards. It consists of four 8×8 luminance blocks and two chrominance blocks. Differently from previous standards, in H.264 it is possible a further subdivision of the 8×8 macroblock into 8×4, 4×8 and 4×4 sub-macroblock like in Figure 2.3. These smaller blocks are useful for tracking much finer details of the scene in the motion-compensated prediction. Along with the 8×8 partition, the macroblock can also be partitioned into two 8×16 or 16×8 blocks. Field mode of the H.264 standard uses 16×16 macroblocks.

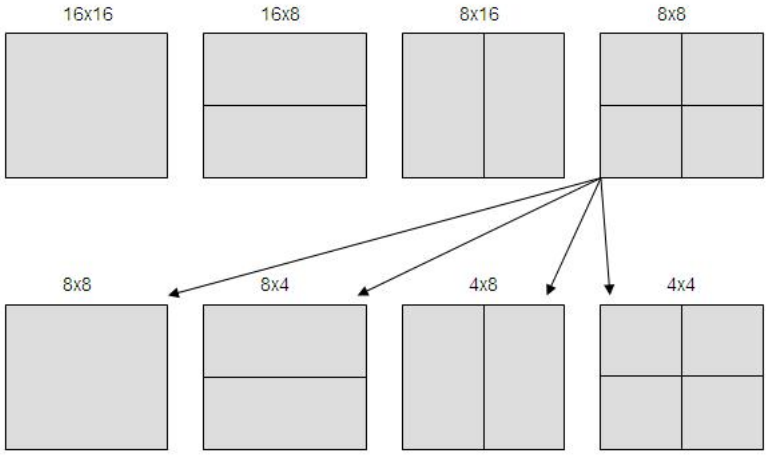

Figure 2.3: Macrobocks in standard H:264/AVC.

2.1.1 Motion Compensated Prediction

In order to predict the pixel values of a frame given the previous frame, the coder has to take into account the motion of the objects in the image. The approach that has worked best in practice is the *block-based motion compensation* [7]. The frame being encoded is divided into blocks of size $M \times M$ and for each block we search for that one most closely matching the encoding block. We measure the distance between two blocks as the sum of the squared differences of the pixels in the two blocks. In the case it is impossible to find a block in the image that is nearer to the encoding block at least for a given threshold, the block is declared uncompensable and it is encoded without prediction. On the other hand, if it is possible for a block to find a valid matching, the block is encoded using a *motion vector*. The motion vector is the relative location of the block to be used for prediction obtained by subtracting the coordinates of the upper-left corner pixel of the block being used for prediction. Therefore, a motion vector is a pointer to the matching block of the previous frame. The H.264/AVC standard exploits motion compensation using quarter-pixel accuracy. The reference image is expanded interpolating twice between two adjacent pixels. This operation results in a smoother residual. The prediction is done by searching among up to 32 pictures to find the best matching. Different predictions are employed

according to the type of the frame [9].
Motion compensated prediction performs the scene dynamic part for the correlation model considered in Chapter 4. The distance in terms of *Mean Squared Error* (MSE) considers, infact, the intrinsic variability of the scene that is reported in the coding of subsequent images.

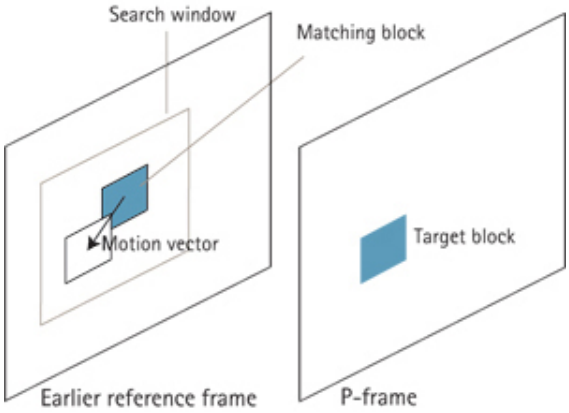

Figure 2.4: Example of motion vector searching.

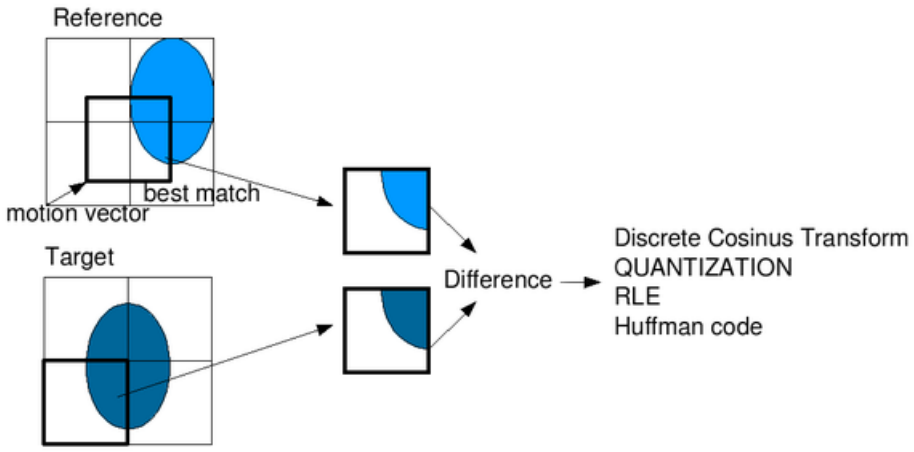

Figure 2.5: Motion compensation in coding procedure.

2.1.2 Transform

The transform used in the coding procedure is a separable integer 4×4 DCT-like transform [7]. The transform matrix and its inverse are given by

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & 2 \\ 1 & -1 & -1 & 1 \\ 1 & 2 & 2 & -1 \end{bmatrix} \quad H^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix}$$

As coefficients are integer, the implementation is simple and avoids error accumulation in the transform procedure, while the small size of the blocks allows a better representation of small stationary regions of the image.

2.1.3 Intra frame prediction

As most of the bits in video coding are expended in encoding I frames, the standard looks for improving their compression in order to substantially reduce the bitrate [7]. The H.264 standard contains several prediction modes. For 4×4 block there are nine prediction modes that are reported in Figure 2.6.

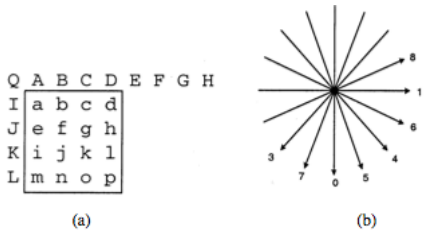

Figure 2.6: Prediction modes for 4×4 intra prediction.

The 16 samples that are labeled as a - p are predicted using prior decoded samples in adjacent blocks labeled as A - Q. The samples on the boundaries of the 4×4 block are copied into the block as indicated by the arrows [9]. For each mode number it is associated a corresponding arrow which shows the direction of the prediction. For example, with mode 1, pixel I is used to predict pixels a, b, c and d. Mode 2 is called DC mode and it is not associated to an arrow: the average of the left and top boundary pixels is

used as the prediction for all 16 pixels. For blocks of different size, similar modes are applied.

2.1.4 Quantization

The standard uses a uniform scalar quantizer for the transform coefficients in H.264/AVC. The value of the parameter can be chosen among a set of 52 values and an increase of 1 means an increse of the quantization step size by about 12%. Note that a change of the step size by 12% also means roughly a reduction of 12% of bit rate. The quantized transform coefficients of a block generally are scanned in a zig-zag fashion and transmitted using entropy coding methods. Say Q_{step} the size of one of the 52 possible quantizers; the step size doubles for every sixth Q_{step}. To make the transform simple, the quantization incorporates scaling process [7].

2.1.5 Network Abstraction Layer (NAL)

The H.264/AVC standard is designed for use in a large variety of applications: video-on-demand, multimedia streaming, data storage are only some of them, while, every day, a lot of new applications enter the market. To address the need for flexibility and customizability, the standard introduces the NAL that formats the *Video Coding Layer* (VCL) representation of the video and provides header information compatible with a variety of transport layers or storage media. The NAL is thus designed to provide "network friendliness", enabling simple and effective customization of the use of VCL for a broad variety of systems. For more information we refer the interested reader to [9].

2.2 Realtime video streaming

A typical video streaming session is composed of two phases: an initial *buffering* phase followed by a *steady state* phase [10]. During the buffering phase, the end-to-end available bandwidth limits the transfer rate and, when a sufficient amount of data is available in the receiver buffer, the player starts

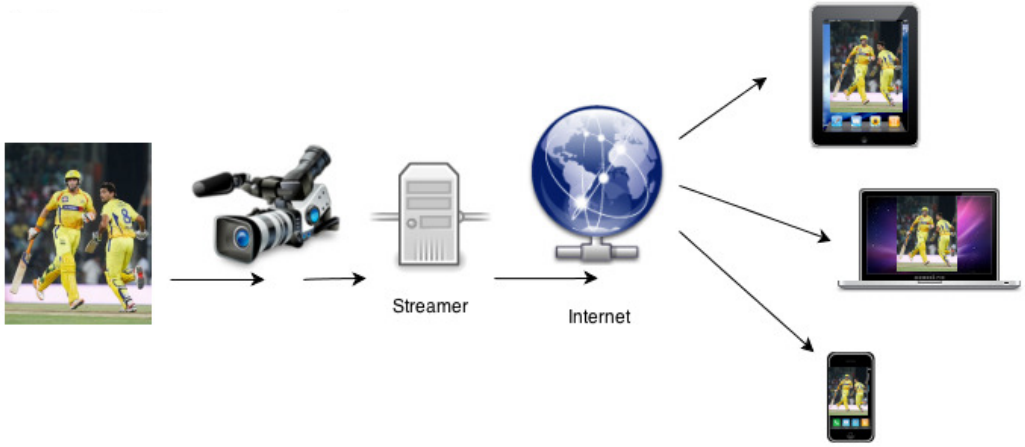

Figure 2.7: Live streaming structure.

the playback. Video playback does not wait for the buffering phase to end. The average download rate in the steady state phase is, instead, slightly larger than the video encoding rate. To evaluate the streaming performance, we consider the ratio between the average download rate during the steady state phase and the video encoding rate. Call it accumlation ratio. When this value goes lower than one, the video playback is interrupted due to the empty buffer, while, a value larger than one means that the amount of video content in the player buffer increases during the steady state phase, improving the resilience to transient network congestion. In the steady state phase, an average download rate is maintained by periodically transferring one block of video content producing a kind of ON-OFF cycles. The buffering phase ensures that the player has a sufficient amount of data to compensate for the variance in the end-to-end available bandwidth during video playback. The reduced transfer rate in the steady state phase ensures that the amount of video content does not overwhelm the video player while keeping constant the amount of buffered data during the buffering phase. The reduced data transfer rate is important for mobile devices which may not be able to store the entire video and the reduced load can increase the number of videos that can be streamed in parallel. Whereas, usually, the downlink channel provides better perfromance than that of the uplink channel, in the second case, the

transmission setting has to be designed considering the lower available bandwidth. A certain data compression technique is necessary in order to obtain good results. This is the reason to adopt H.264/AVC and H.264 *Scalable Video Coding* (SVC) [11].

2.2.1 Streaming protocols

Designing a network protocol that support video streaming should consider a set of aspects [12][13], among which there is the transport protocol.

- **UDP:** this protocol send data into a flow of small packets. There is no guarantee of delivery, but it is simple and efficient. Retransmission and error correction techniques should be implemented by the application layer;
- **TCP:** it is more complex than UDP to implement. It guarantees the correct delivery of the data implementing mechanisms of timeouts and retransmissions. However, when the protocol detects a loss, the stream stalls and retransmits the data incrementing the delay. While this aspect is tolerable in video-on-demand scenario, it is not acceptable in realtime applications like video conferencing;
- **RTSP:** the Real-time Streaming Protocol was specifically designed to stream media over networks. RTSP runs over a variety of transport protocols;
- **RTP and RTCP:** similar to RSTP but they both run over UDP;
- **Unicast:** separate copies of the media stream are sent to each recipient. It is not scalable;
- **Multicast/Broadcast:** it sends a single stream from the source to a group of recipients (1-to-many networks). Some prior work on broadcast video streaming rate allocation can be found in [14] and [15];
- **Peer-to-peer:** protocols arrange for prerecorded streams to be sent among computers. This prevents the server and its network connections

from congestion. However, it raises technical, performance, security, quality, and business issues.

2.2.2 Video quality metrics

Quality metrics able to fairly represent the video stream nature hardly characterize the human video perception. However, some different approaches were found: QoS metrics are related to the technical aspects of the network [16], while QoE metrics are based on perceptual evaluation of the video stream [17]. The main video quality metrics in use are: PSNR, MSE, *Structural Similarity* index (SSIM) and *Mean Opinion Score* (MOS). PSNR and MSE are QoS-based video metrics whereas SSIM and MOS are an objective and subjective representations of the QoE, respectively.

- **PSNR:** [18] is defined as the ratio between the maximum power of a signal and the power of the noise that corrupts it. Because of the high variance of the signals, usually, PSNR is expressed in decibel. It is used for estimating the quality of the reconstruction of a lossy coding, but it can be used also to evaluate the quality of a video frame given its distorsion introduced by the channel through which it has been transmitted. PSNR is a metric intended to be an approximation to human perception of the signal. It is defined via the MSE. Given an $M \times N$ image I and its noisy approximation K, we have:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i,j) - K(i,j)]^2 \tag{2.1}$$

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned} \tag{2.2}$$

where MAX_I is the maximum possible pixel value of the image.

Although high values of PSNR indicates high quality, sometimes it may not be the case. For this reason other indices have been introduced.

- **SSIM:** [19] measures the structure similarity between two images. It is designed to improve PSNR and MSE metrics which have proven to be inconsistent with the human eye perception. Differently from the aforementioned techniques, SSIM considers image degradation as changes in the image structural information that is based on the strong correlation between neighboring pixels. SSIM is computed between serial windows of two images x and y of dimensions $N \times N$:

$$SSIM(x, y) = \frac{2(\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2.3}$$

  where

  - μ_x is the average of x;
  - μ_y is the average of y;
  - σ_x is the standard deviation of x;
  - σ_y is the standard deviation of y;
  - σ_{xy} is the covariance of x and y;
  - $c_1 = (0.01L)^2$, $c_2 = (0.03L)^2$;
  - L is the dynamic range of the pixel values;

  This formula is computed only on the luminance component of the images. Examples of SSIM-based resource allocation algorithms for realtime streaming are given in [20].

- **MOS:** used for decades with the intent of formalizing in a mathematical way the user's perception of the quality of the network. Historically used for audio quality classification, it was computed by some people listening in a quiet room to some audio signals and scoring according to the quality they perceived. ITU-T recommendation P.800 [21] explains the environmental conditions needed for a good estimation of

the MOS index for audio calls. Like most standards, the implementation is somewhat open to interpretation by the equipment or software manufacturer. Moreover, due to technological progress of phone manufacturers, a calculated MOS of 3.9 in a VoIP network may actually sound better than the formerly subjective score of greater than 4.0. The Mean Opinion Score for multimedia, i.e., audio, voice, video, provides a numerical indication of the quality perceived by the user after compression and transmission. MOS is expressed as a single number in the range from 1 to 5, where 1 is the lowest perceived video quality, and 5 is the highest perceived video quality measurement. MOS is computed taking the average of the scores of a number of listeners that experience the signal under study. In Table 2.1 we report the meaning of the MOS values in terms of quality and impairment perceived by the end users.

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
4	Poor	Annoying
5	Bad	Very annoying

Table 2.1: Mean opinion score MOS.

Chapter 3

Related work

Prior work is focused on the design of models which study the correlation among images and the associated theoretical rate-distortion bounds, as in [22]. In this work, we propose a correlation model for pixels belonging to two consecutive frames as the product of a spatial and temporal factors. However, the spatial correlation is computed based on the local texture of a specific video, which makes the method infeasible in practice, in terms of computational complexity, for user-generated content real-time applications and services. Strong temporal correlation models between adjacent frames of video signals have been successfully exploited in standard video compression algorithms, such as the MPEG [23] codec family and in particular the H.264/AVC [24] and SVC [11], which are block-oriented motion-compensation-based video compression techniques. The tight relation between the dynamics of the source and the scene with the frame correlation is still an open research item which needs to be modeled to best adapt in real-time the video encoding process to the mobility pattern and preferences of the video upstreaming producers.

Other related work mainly focuses on the design of MDPs to model the scheduling of data packets over time-varying shared channels. For instance, in [25], the authors model the one-user scheduling problem as an MDP and derive from it the structural properties of the optimal solutions in order to design online learning algorithms that preserve such properties and achieve nearly optimal performance. The video encoding rate selection is not taken

into account before the scheduling decisions since the study is limited to the wireless scheduling transmission and does not consider the impact of the mobility pattern of the user. The extension to the multi-user case [26] takes into account the users' heterogeneous video traffic characteristics, the time-varying network conditions and the dynamic coupling among the users' resource allocations across time, but the role played by the mobility of the user and the dynamics of the scene in the system are not considered, nor is the frame correlation information.

In [27], the authors propose a systematic solution to the problem of scheduling delay-sensitive transmissions over time-varying wireless channels. A dynamic scheduling scheme is designed as an MDP that explicitly considers the users' heterogeneous characteristics. Foresighted decisions are made to schedule multiple data units with different priorities based on the users' requirements. An online learning algorithm is developed to capture the impact of the current decision with scope limited to the packet scheduling problem.

In [28] the authors design two scheduling algorithms that optimize the quality of scalably coded videos. The first scheduling scheme is derived from an MDP and models the dynamics of the channel. Based on this, a near-optimal scheduling policy is computed that minimizes the distortion of the video. Upon the insights taken from this model, an online scheduling algorithm is designed. This work, based on the dynamics of the channels and on prioritized video queues to be scheduled, does not take into account the information about the frame correlation of consecutive video frames to minimize the video distortion already at the encoding phase.

Contrary to this related work, in this work we take into account the frame correlation between consecutive video frames, for instance, due to the user mobility or the dynamics of the scene, in the rate-distortion formulation, and use this information to optimally select the video encoding rate and the scheduling transmission policy under the constraint of the wireless channel conditions. The optimal decision is made so as to minimize the impact of the selected actions on the perceived video quality at the consumer side in terms of average long-term distortion.

Chapter 4

Analytical study

In this chapter we perform the analytical study of the proposed delivery system. The introduction of the use case scenario is followed by the modeling and the formalization of the problem. Finally, an analytical solution is provided.

4.1 System description

In our delivery system we envision a mobile video source generating a sequence of video frames in real-time. This sequence is coded at the source and transmitted via a wireless connection (either Wi-Fi or 3-4 G) in order to be shared in the Internet (e.g., social networks), such that video consumers can watch the real-time user-generated content.

We assume a dynamic user shooting or displaying a dynamic scene that is characterized by a frame correlation model. The information provided by the correlation model can be used by the encoding process to properly set the encoding rate, which is defined in our system as the number of bits used to encode the video frame. Highly correlated frames will require few bits to be represented and played at the receiver (most of the information is contained in the previous frame), whereas uncorrelated frames will need high encoding rates to supply the lack of correlation with previous frames. Moreover, the selection of the encoding rate has an impact on the scheduling policy of the transmitter, since it has to best schedule its packets to cope with

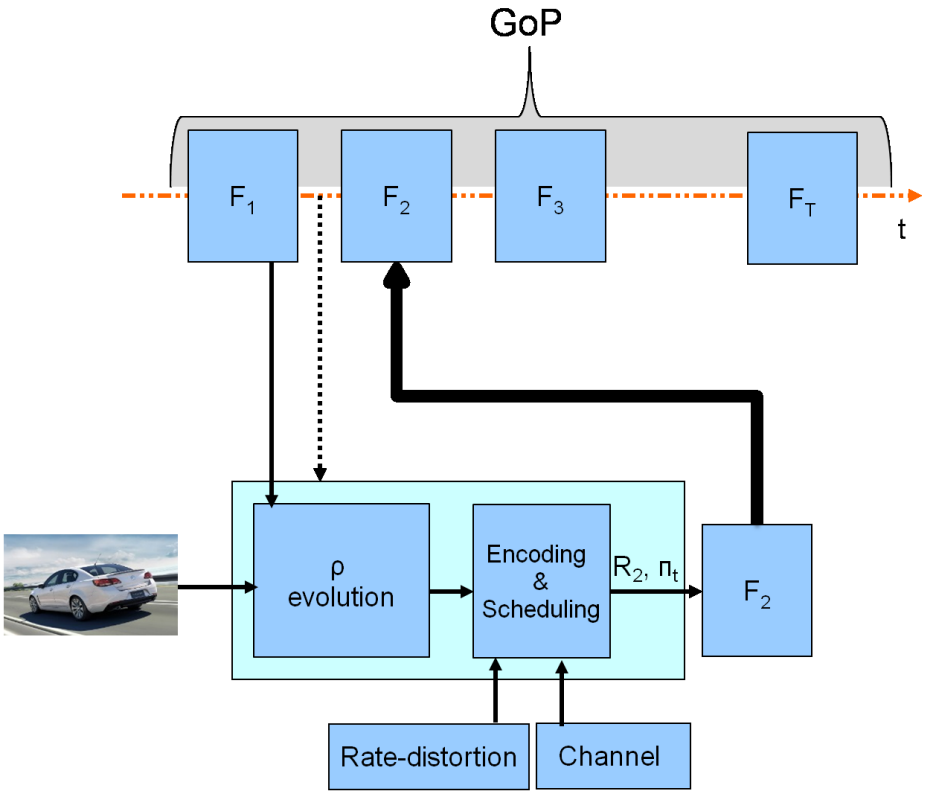

Figure 4.1: Framework overview.

the available channel conditions. We assume that a video frame contributes in terms of perceived quality when the frame is fully received. Frames that are partially transmitted are discarded and, hence, negatively impact the perceived quality of the uploaded video.

We design a delivery framework for real-time mobile video upstreaming which takes into account the video frame correlation to opportunistically encode and transmit video frames on a per-GoP basis, whose sequence starts with an I frame followed by P frames. The goal of the system is to select the optimal encoding rate and the scheduling policy at the source to achieve a target perceived distortion (or, viceversa, quality) of the video produced.

In Figure 4.1, our proposed framework comes to play right after the latest video frame F_1 has been encoded. We consider a first module which foresees the evolution of the video flow in terms of correlation between video frames.

The module computes the frame correlation ρ between the last video frame F_1 and the current frame to be encoded, F_2, based on the mobility of the camera (source), and on the changes of the video content due to the dynamics of the scene. The outcome of this computation is fed into a second module, the optimizer, which is in charge of making the decision on whether or not to encode the video frame in this time instant, at which encoding rate R_2 and how/when to schedule the frames queued at the transmitter, i.e., it selects a scheduling policy π_t. This module makes the decision based on a rate vs. distortion mapping function, which takes into account the frame correlation, and on the wireless channel conditions. That is, the encoding rate of each video frame is selected with the goal of minimizing the overall perceived distortion of the video, or viceversa, to enhance the overall delivered quality, and to keep it steady through the variations of the wireless channel conditions, Γ_t.

4.2 System model

We assume that the mobile user acquires a video frame in each time slot. Thus, we assume that at the current time slot t the system acquires frame F_t. Each video frame is correlated by a factor $\rho^{(t)} = \rho(F_{t-1}, F_t)$ with the previous video frame acquired. Once the current video frame is acquired by the camera, the encoding rate for such frame is selected based on the correlation between consecutive frames and based on the target video distortion (or, conversely, the target video quality) to be achieved by the system. Assuming that the frame will be entirely transmitted, the distortion can be written as a function of the encoding rate R_t and the frame correlation $\rho^{(t)}$:

$$D_t = f(R_t, \rho^{(t)}) \tag{4.1}$$

Assuming that the video frame generation is a Gaussian process with mean μ and variance σ^2 and that the samples are uncorrelated, we find the following analytical expression [29] for the rate-distortion function:

$$D_t = \mu \cdot \sigma^2 \cdot 2^{-2R_t} \tag{4.2}$$

As a general rule, the distortion values computed with the above formula can only be attained by increasing the coding block length. Nevertheless, even at unit block lengths one can often find good quantizers which operate at reasonable distances from the rate-distortion function.

When the frame correlation is non-zero, subsequent frames can be partially represented by the initial frame of the sequence. Thus, given the correlation information, the encoding rate of the video frames can be reduced. In the case of maximum correlation, i.e., $\rho^{(t)} = 1$, no encoding is necessary. Conversely, when the correlation $\rho^{(t)} = 0$, the encoder will have to use the maximum rate to best represent the completely new video frame. Hence, we modify Equation (4.2) as follows:

$$D_t = \mu \cdot (\sigma^2_{MAX} - \Delta\sigma(\rho^{(t)})) \cdot 2^{-2R_t} \tag{4.3}$$

where σ^2_{MAX} is the maximum variance of the encoding residual, that is obtained without knowing the frame correlation, and $\Delta\sigma(\rho^{(t)})$ is a discount factor on the variance, and thus on the distortion, due to the correlation information as discussed above. The resulting variance, $\sigma^2_{MAX} - \Delta\sigma(\rho^{(t)})$, which is modeled so as to fulfill the condition of having an uncorrelated signal [29], can be used by a predictive encoding process to encode consecutive video frames. From Equation (4.3), we note that the source can decide to tune the encoding rate given the frame correlation as input to best meet the target video distortion of the service and, at the same time, to meet the channel capacity constraints.

As a general observation, the higher the frame correlation, the lower the encoding rate required to achieve the same level of distortion, and viceversa. Existing rate control schemes incorporate spatio-temporal correlations to improve the accuracy of rate-distortion models, by using statistical regress analysis for dynamical model parameters update. Representative of this approach is the linear mean absolute difference model in [30], where model parameters are updated by linear regression. Thus, we assume a linear approximation of the relation between frame correlation and video distortion as follows:

$$D_t = \mu \cdot \sigma^2_{MAX}(1 - \rho^{(t)}) \cdot 2^{-2R_t} \tag{4.4}$$

The extreme cases of Equation (4.4) are given by setting $\rho^{(t)} = 0$, i.e., no correlation information can be used in the encoding process and the distortion function becomes as in Equation (4.2), while with $\rho^{(t)} = 1$ neither video encoding nor frame transmission is required. The latter case can be seen as a static scenario (both source and scene are fixed), while the first case above is the most dynamic scenario that will be considered.

4.3 Problem formulation

The general formulation of the problem provides a mathematical representation of the system. The model is developed considering a video frame F_t generated at time t and a frame transmission deadline of N consecutive time slots. A video frame that can not be transmitted within a deadline of N slots from the instant in which it has been acquired will be discarded. The set of video frames candidate for transmission at time t is $\underline{C}_t$, each one with its respective deadline counter. This last variable is assigned to every frame in $\underline{C}_t$ and accounts for the available slots for transmission, for a given frame, before expiring. Thus, in the following we indicate with $F_i \in \underline{C}_t$ the couple *frame-deadline counter*. We assume that each frame is acquired at time $T_A(t)$ and encoded with a given rate specified in the decision $\underline{\pi}_t$ that we need to compute for each time slot t. The decision $\underline{\pi}_t$ is a vector of lenght $|\underline{C}_t|$ with entries equal to the transmission rates scheduled for time slot t for all the frames in the set $\underline{C}_t$, which is given by Equation (4.5).

$$\underline{C}_t : \{l \text{ s.t. } T_A(l) \leq t \leq T_A(l) + N - 1\} \tag{4.5}$$

Moreover, a mobile user experiences variable channel conditions defined by the variable $h_t \in \mathcal{H}$ and, therefore, a certain bandwidth $B(h_t)$ constraint. The state s_t of the Markov chain is defined by the tuple $(\underline{\rho}_t, h_t, \underline{C}_t, \underline{R}_{left})$ where $\underline{R}_{left}$ is a vector containing the rate to be transmitted for the frames in vector $\underline{C}_t$, and $\underline{\rho}_t$ are the respective correlations. For example, if a given frame i has been coded with rate $2R$ and transmitted with rate R, vector $\underline{R}_{left}$ will have the value R at index i. Moreover, for the frames that have been

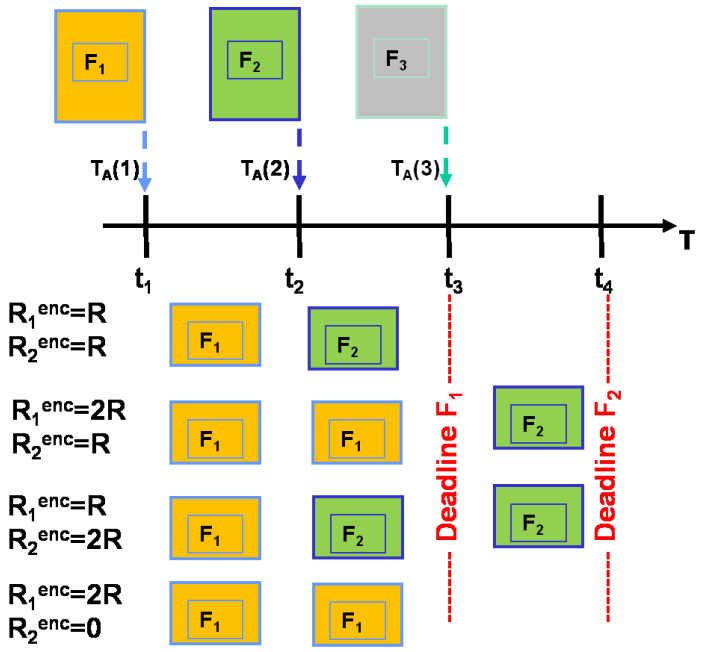

Figure 4.2: Example of possible encoding and scheduling decisions for $N = 2$.

acquired but not yet coded, the vector values will be temporarily undefined[1].

The solution to the problem should provide the choice of $\underline{\pi}_t$ for t within a given prediction horizon T in order to maximize an expected total reward of the frames. To do so, we introduce the action matrix $\underline{\Pi}_t = [\underline{\pi}_t, \underline{\pi}_{t+1}, \ldots, \underline{\pi}_{t+T-1}]$ that considers the decisions for the entire time orizon T in which the optimization is performed. Moreover, $u(s_t, \underline{\pi}_t)$ is the reward achieved by taking action $\underline{\pi}_t$ at time t knowing that we were in state s_t. In order to give less relevance to frames that are farther in the optimization range T than those which are nearer to the actual time slot t, we introduce a discount factor γ^{t+k} for the foresighted optimization. In this scenario, the problem formulation is as follows:

$$\max_{\underline{\mathbf{\Pi}}_t} \ \mathbb{E}\left\{\sum_{k=0}^{T-1} \gamma^{t+k} u(s_{t+k}, \underline{\pi}_{t+k})\right\} \tag{4.6}$$

$$\text{s.t.} \ \sum_i \underline{\pi}_{t+k,i} \leq B(h_{t+k}) \ \ \forall k \in [0, T] \tag{4.7}$$

[1]Enqueued packets are not yet coded, thus their conding rate still has to be decided.

Assuming in our model that the frame correlation ρ is known, the only random variable that we have is the channel. Then, the expectation is done over all possible channel values obtaining the expression in Equation (4.8).

$$\mathbb{E}\left\{\sum_{k=0}^{T-1}\gamma^{t+k}u(s_{t+k},\underline{\pi}_{t+k})\right\} = \gamma^t u(s_t,\underline{\pi}_t) + \sum_{k=1}^{T-1}\gamma^{t+k}\left\{\sum_{s_{t+k}:h_{t+k}\in\mathcal{H}} p(h_{t+k}|h_{t+k-1})u(s_{t+k},\underline{\pi}_{t+k}|\underline{\pi}_{t+k-1})\right\} \quad (4.8)$$

4.4 Analytical solution

Based on the general model of the previous section, we assume a simplified scenario for the sake of tractability. We set $\gamma = 1$[2] and we leave as future work the extension to other values of γ. We divide the channel conditions in two cases: good channel and bad channel, $h_t \in \{h_g, h_b\}$. With h_g channel, we are allowed to transmit at rate R while, with h_b channel, only $R/2$ can be sent. Moreover, in order to simplify the framework, we assume that when a new frame is coded all the previous frames not already transmitted are discarded. In this way, if we are in a given state s_t and we acquire a new frame F, we can decide to either put it in a queue and continue to transmit the current frame or encode the new frame neglecting all of the pending frames, if any. In the latter case the distortion increases. Obviously this framework provides a suboptimal solution as it may waste resources because it does not transmit parts of different frames in the same time slot. For instance, when the channel is good and the last part of the current frame is only $R/2$, we miss the remaining available channel capacity. However, these assumptions make it possible to design a framework which is tractable from the analytical point of view. As in this model we want to maximize the average total reward associated to the transitions, we foresee two actions:

1. we assign to each transition its associated reward, gained by the (even partial) transmission of the frame. In case the frame is not completely

[2]With a finite time horizon T, the sum is always convergent.

transmitted, once we delete the frame, we also delete the partial rewards previously gained. In other words, if we transmit R bits of a given frame i, encoded with a rate $R_i = 2R$ along a good channel at a certain time instant, then, with this transition we obtain an estimated reward of $D_i^{max} - D_i(R)$ where D_i^{max} is the maximum distortion of the frame, i.e., with $\rho^{(i)} = 0$ and $R_i = 0$, and $D_i(x)$ is the distorion of the frame i with transmission rate x. Moreover, if a second transition allows the transmission of the remaining R part of the frame, it brings one more improvement of $D_i(R) - D_i(2R)$, for a total reward of $D_i^{max} - D_i(R) + D_i(R) - D_i(2R) = D_i^{max} - D_i(2R)$. Differently, if we are not able, in the second transition, to complete the transmission and we decide to encode a new frame, the reward relative to this decision is $D_i(R) - D_i(0)$ with a total reward of $D_i^{max} - D_i(0)$ for that frame. However, we need to distinguish the case when a frame is placed in the queue and its coding procedure is forwarded from the case in which we encode a frame with rate $R = 0$. In the former we get a reward of $D_i^{max} - D_i^{max} = 0$ while, in the latter, a reward of $D_i^{max} - D_i(0)$.

2. we can assign rewards to frames only once their transmissions are completed. When we decide, for example, to encode a certain frame with rate $R_i = 2R$ but the channel is, in order, bad for two slots and good for one, we can transmit only $R/2$ for two time slots and R for the last slot. Using this second method, the rewards associated with the three transitions are, respectively, $[0, 0, D_i^{max} - D_i(2R)]$, for a total reward of $D_i^{max} - D_i(2R)$. Also here, the case of coding with rate zero must be treated differently from the case of enqueued frame. The first case provides a reward of $D_i^{max} - D_i(0)$, whereas the second is temporarily undefined.

In this work we take the second approach in the list and we represent in Figure 4.3 a simplified version of its use case. We impose that the initial frame should be always encoded at a non-zero rate. Thus, from the initial state we can choose rate R or $2R$ and, given that we are in good channel, we transmit R bits of the actual frame. The next state can be one of the

four successive states. Considering, for instance, the first state of the second column reached with probability p and with the decision labeling the arrow, the slot in which we receive frame F_2 has bad channel and, thus, allows for the transmission of only $R/2$ bits. From this state we have three choices related to different coding decisions. We can select the encoding rate in the set $\{0, R, 2R\}$ and, according to the channel probabilities, we will have good or bad channel. Making the choice of not encoding frame F_2, we do not need to send it and, thus, in the next slot, we are ready to code the new received frame F_3. This decision brings a partial reward of $D_2^{max} - D_2(0)$. Otherwise, encoding F_2 with rate R, will mean that in the next slot $R/2$ rate is left out because of the bad channel condition. In this successive state, we receive frame F_3 that is appended in the queue $\underline{C}_t$ and we need to decide whether to encode it or not. The reward for this decision is zero as we do not terminate F_2 transmission. Using, instead, $2R$ bits to encode F_2, we obtain similar conditions to the previous choice, but with $3R/2$ bits left. In the rightmost part of Figure 4.3 we show some total rewards weighted with channel probabilities and related to their respective paths. Our optimal solution consists in computing the total reward for all the possible paths and taking that one with the highest reward. Once we know the optimal path, we select its first transition as the next scheduling-coding action to be taken, and we repeat the optimal prediction for the next time slot. Note that the depth of the tree in Figure 4.3 is the length of the prediction window. The size of the vector $\underline{C}_t$, that contains, in a time-increasing order, the pending frames for which the deadline is not yet reached, is at most N. When a frame i is not transmitted within the deadline, that frame is dropped and the transition in the Markov chain will count the distortion with a reward of $D_i^{max} - D_i(0)$. For each frame, the deadline counter decreases by one for every time slot the frame is in queue or not completely transmitted. When it reaches zero the frame is dropped.

The development of a complete framework is left for future work. We foresee to adopt further assumptions in order to keep the complexity low for future practical implementations.

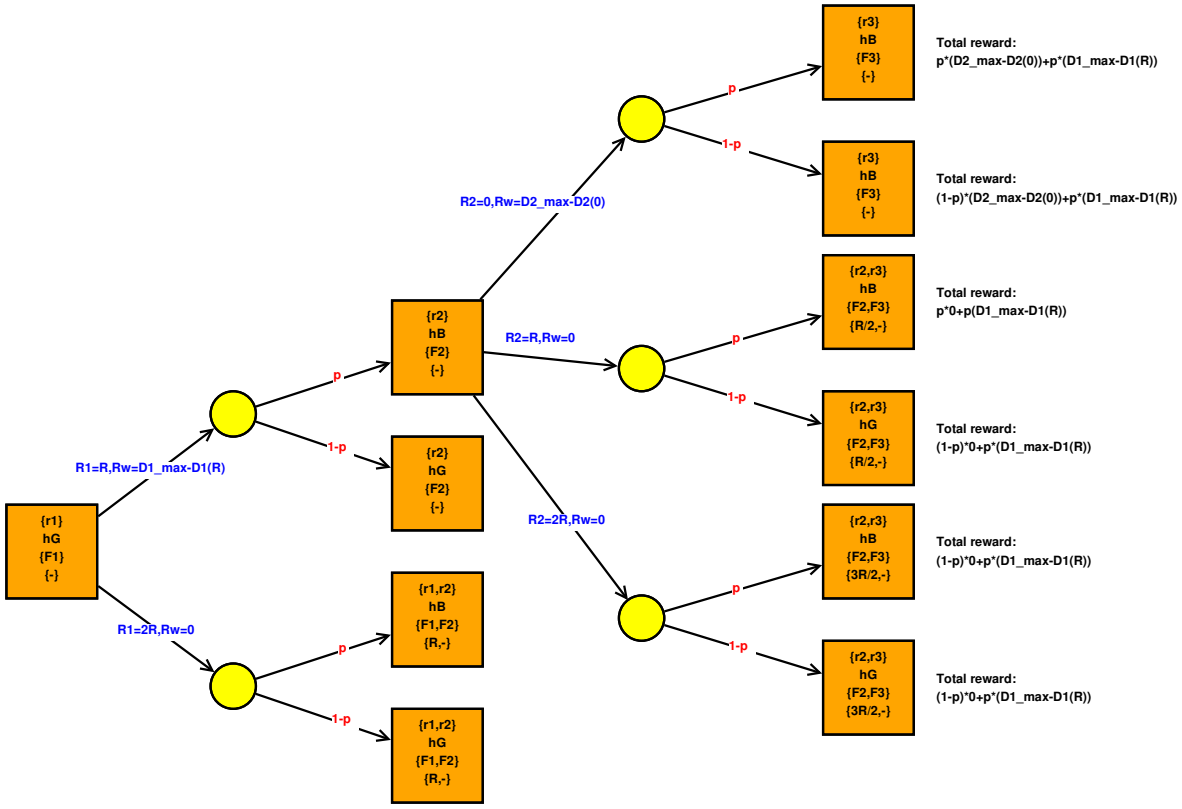

Figure 4.3: Analytical solution.

Chapter 5

Markov decision model

Starting from the general analysis in Chapter 4, we build a Markov model according to a simplified but practical implementation of the framework. We extend the framework by considering the scheduling of the frames to be transmitted, thus we omit the requirement of ordered delivery as for the analytical framework in Chapter 4. Moreover, we assume to know the channel capacity for the entire prediction window and we inspect the impact of the window size on the performance of the framework. We also analyze the impact of the correlation by assessing the framework via simulations for a set of scenarios. Then, we compare our solution obtained using quantized values of the frame correlation information with respect to that using real values of it.

5.1 Problem formulation

We consider a video frame generated at time t, say F_t, and a frame transmission deadline of two consecutive slots ($N = 2$), i.e., t and $t+1$. Hence, a frame that cannot be transmitted within two time slots will be discarded, which increases the video distortion. We call this encoder-decoder scheme "basic," where once the video frame is encoded, it must be completely received at the decoder, otherwise the frame is dropped. Based on the correlation between frames, we can assume that in case a frame is not received, the frame correlation information can still be used to partially reconstruct the missing frame to

keep the distortion low (similarly to common error concealment techniques). Extension of this framework to a "scalable" encoder-decoder scheme [11] is left for future work. In this case, a frame encoded at a certain rate contributes to decrease the distortion at the receiver even if only partially received. The more the bits received per frame, the lower the distortion.

As a first step towards a more general model for the study of joint encoding-scheduling strategies, we consider the use case scenario depicted in Figure 4.2. Each video frame is acquired at time $T_A(t)$ and encoded with rate picked from the set $\{0, R, 2R\}$, with channel capacity fixed and set to R (bits per slot). Based on the selected encoding rate, there are several scenarios for the scheduling policy to be adopted for a sequence of 2 video frames. Note that, based on the frame correlation between consecutive frames and on the channel capacity, the encoding rate varies along with the acquisition of the video frames, which significantly impacts the evolution of the video scheduling strategy in the time domain. For instance, in the first row the decision made by the system is to encode both frames F_1 and F_2 with rate R, thus each frame will be encoded and sent in the corresponding acquisition slot. However, it may happen that the first frame, F_1, contains more information or is weakly correlated to the previous frame so that it requires a higher rate in the encoding process to keep the distortion low. In this case the frame will be encoded at rate $2R$ and consequently sent over two slots. Hence, the system is dictated to generate frame F_2 at rate R as in the second row of the example or rate 0 as in the last row. The choice depends on the frame correlation information, in fact the system might prefer not to encode frame F_2 in case of high correlation with the previous frame, thus making room for the next video frame to be encoded (which is then prioritized), or it can decide to encode frame F_2 in case it foresees that this will improve the performance independently of which encoding rate will be decided for F_3. Thus, the frame correlation is used to prioritize the video frame encoding and scheduling decisions throughout the whole sequence of frames, e.g., a GoP.

In our analysis we first assume a static channel, i.e., the channel capacity is constant and thus known in advance. This assumption makes it possible

to isolate the impact of the frame correlation on the system performance. Once we identify the behavior of the system when the decisions are based on the frame correlation information and the target video distortion to be achieved, we add another degree of complexity to the problem by including time-varying channel conditions into the problem formulation.

We define the distortion D_t of frame F_t as a function of its encoding rate, i.e., R_t and of the frame correlation $\rho^{(t)}$ as in Equation (4.4). Once frame F_t is encoded at rate R_t, it can be sent at time t with transmission rate $R_t^{(1)}$, and at time $t+1$ with rate $R_t^{(2)}$, where $R_t = R_t^{(1)} + R_t^{(2)}$. Moreover, since each frame can be sent within two slots with a given channel capacity C_t, it holds that $R_t^{(1)} + R_{t-1}^{(2)} \leq C_t$.

We note that the distortion is an additive metric which can be used over a GoP to measure the overall distortion of the sequence of video frames produced [31][32]. As future work we will consider a QoE-based rate-distortion function, thus we can map the distortion, i.e., the MSE, using Equation (2.2) and then we map the PSNR to SSIM [33] or Video SSIM [34], which can be assessed in terms of MOS [19].

The goal of our work is to select the encoding rate and transmission policy, i.e., the tuple of possible actions $\underline{\pi}^*$, that minimize the average long-term distortion of a sequence of video frames, e.g., a GoP of time window of T slots, as follows:

$$\underline{\pi}^* = \arg \min_{\underline{\pi}} \frac{1}{T} \sum_{\tau=t}^{t+T-1} D_\tau(R_\tau, \rho^{(\tau)}) \tag{5.1}$$

under the channel capacity constraint C_τ.

5.2 Solving method

We solve our optimization problem via dynamic programming [35]. Given the distribution of the frame correlation values as input, we compute the optimal combination of video coding and transmission scheduling policy over a sequence of video frames (GoP) which minimizes the average long-term distortion of such video sequence. Once the distribution of $\rho^{(t)}$ is known,

the problem becomes to find the path with the minimum value of distortion in the corresponding Markov chain. Since we assume that each video frame must be delivered within the deadline of two time slots, the states of the MDP have to keep the information of the transmission rate, frame correlation and distortion of up to 2 slots before. Our method predicts the optimal

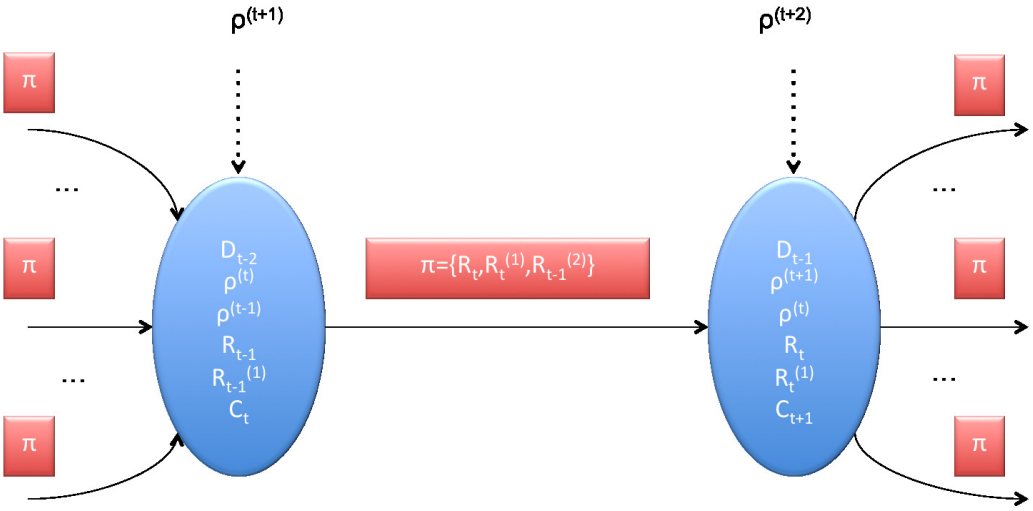

Figure 5.1: Transition from state at time t to $t+1$.

policy based on the frame correlation information between video frames, the current channel capacity and the target video distortion to be delivered. In our framework each state of the chain includes five quantities. The first component is the distortion of the second last frame, D_{t-2}, since it has been already sent out and thus it can be computed. Then the state contains the frame correlation between the last and second last frames, $\rho^{(t-1)}$, and the frame correlation between the current and last frame, $\rho^{(t)}$, which are used in slots $t+1$ and $t+2$ to compute D_{t-1} and D_t, respectively. Furthermore, the state contains the encoding rate of the video frame acquired in the previous slot $t-1$, R_{t-1}, and its transmission rate in the same slot, $R_{t-1}^{(1)}$

Thus, the state of the system S_t at time t can be written as the following tuple:

$$S_t = \{D_{t-2}, \rho^{(t-1)}, \rho^{(t)}, R_{t-1}^{(1)}, R_{t-1}\} \tag{5.2}$$

Among all possible combinations of those quantities, which represent the

entire state space, a subset of them are feasible and build up our solution space. In particular, the solution space is given by the states fulfilling the condition that the transmission rate in the first available slot is not greater than the encoding rate of a frame, $R_t^{(1)} \leq R_t$. The decision to be made is given by $\pi_t = \{R_t\}$, i.e., the encoding rate of frame F_t, R_t. In the static channel scenario, thus $C_t = C$, the second part of the previous frame, $R_{t-1}^{(2)}$, is forced by the current state to assume the value of the remaining unsent part of R_{t-1}, whereas the capacity constraint dictates that $R_t^{(1)} = C - R_{t-1}^{(2)}$. Once the action is selected, D_{t-1} can be computed and used in the following state. At time $t+1$, the state of the system will be $S_{t+1} = \{D_{t-1}, \rho^{(t)}, \rho^{(t+1)}, R_t^{(1)}, R_t\}$ and the decision will be given by $\pi_{t+1} = \{R_{t+1}\}$, which makes it possible to eventually compute D_t. Each transition in the chain, as shown in Figure 5.1, is associated to decision π_t. We build the chain considering only those transitions that guarantee lossless transmissions (basic encoder-decoder scheme as aforementioned), i.e., $R_t^{(1)} + R_t^{(2)} = R_t$, which implies $R_t^{(1)} \leq R_t$, and meet the channel capacity constraint $R_t^{(1)} + R_{t-1}^{(2)} \leq C$. Among all admissible paths in the solution space of our MDP, we select the optimal transition that fulfills Equation (5.1).

5.2.1 Dynamic channel case

We introduce in the optimization problem statement a Markov-based time-varying channel quality, Γ_t, and the respective channel capacity as $C_t = C(\Gamma_t)$, which is the number of bits that can be reliably transmitted in slot t. Note that the decision for the dynamic channel scenario is now given by the tuple $\pi_t = \{R_t, R_t^{(1)}, R_{t-1}^{(2)}\}$, with $R_{t-1}^{(2)}$ set to $R_{t-1} - R_{t-1}^{(1)}$ if $R_{t-1} - R_{t-1}^{(1)} \leq C_t$ and to 0 otherwise. The state of the system contains also the channel information compared to Equation (5.2):

$$S_t = \{D_{t-2}, \rho^{(t-1)}, \rho^{(t)}, R_{t-1}^{(1)}, R_{t-1}, C(\Gamma_t)\} \tag{5.3}$$

The information of the channel state adds a degree of complexity to the problem, and will be evaluated in the simulation results once the impact of the frame correlation on the system is separately assessed by using a static channel.

5.2.2 Setup

We implement in Matlab an MDP with a set of available encoding rates, i.e., $\{0, R, 2R\}$, and a set of channel capacity values, i.e., $\{0, 0.5R, R, 1.5R, 2R\}$. The set of available transmit rates is the same as that of the encoding rates. For the static channel scenario, we let the channel pick one fixed value for the whole session, whereas in the channel dynamic scenario the channel capacity varies following a given distribution, but we reduce the set of possible values to $\{0, R, 2R\}$ due to the additional computational complexity. We set $\mu = 1$ and $\sigma^2_{MAX} = 200$. To assess the impact of the frame correlation on the performance of the system, we consider a set of ranges of slots that can be foresighted, called prediction time window, that includes the values $\{1, 2, 4\}$. A unit time window corresponds to the case of no prediction, i.e., the encoder generates the frame with rate set to the current channel capacity. The frame correlation values are picked from the set $\{0, 0.3, 0.6, 1\}$ using different statistical distributions, as in Figure 5.2, where a correlation value of 0 corresponds to the case of uncorrelated video frames, i.e., a very high mobility scenario, whereas a correlation value of 1 refers to a static scenario (perfectly correlated frames). In order to have some variability in the correlation, distributions in Figure 5.2 are obtained by sampling uniform, gaussian and exponential distributions with the following thresholds: $thr_0 = -\infty$, $thr_1 = 0$, $thr_2 = 0.15$, $thr_3 = 0.45$, $thr_4 = 0.8$, $thr_5 = 1$, $thr_6 = +\infty$.

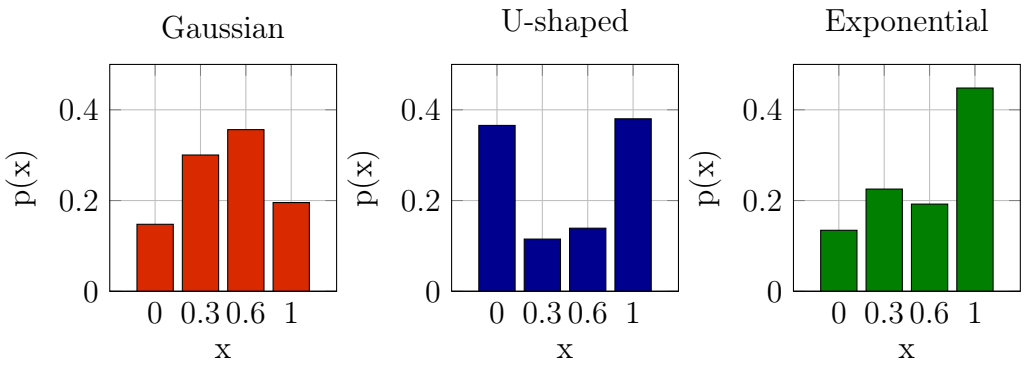

Figure 5.2: Different statistical frame correlation distributions.

Due to the heterogeneous intervals defined by those thresholds, the contin-

uous uniform (0,1) distribution turns into a Gaussian sampled distribution, the continuous gaussian ($\mu = 0.5$, $\sigma = 1$) into a U-shaped distribution and the continuous exponential ($\lambda = 1$) into a sampled exponential distribution. The choice of defining small sets of values for all the metrics involved in the framework is due to the complexity associated to the corresponding MDP. Note that the scope of this work is to qualitatively assess the performance of an MDP which jointly selects encoding rate and scheduling policy in real-time based on the dynamics of the environment and on the channel conditions.

5.2.3 Discussion

First of all, in Figure 5.3, we isolate the impact of the frame correlation when varying its statistical distribution and the available channel capacity. We pick ρ in $\{0, 0.3, 0.6, 1\}$ according to a discrete gaussian, exponential, or U-shaped distribution (i.e., the extreme values 0 and 1 have higher probability of being chosen) like in Figure 5.2, and measure the performance of the MDP in terms of average distortion over a short-term horizon of 20 slots, e.g., a possible GoP size. We implement the upper and lower bounds which correspond to setting ρ to 0 and 1, i.e., the highly dynamic and static scenarios, respectively.

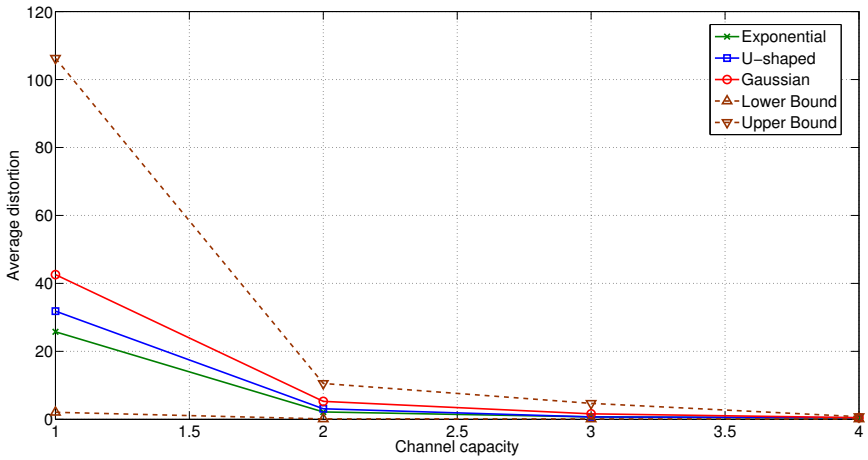

Figure 5.3: Average distortion for exponential, U-shaped and gaussian distributions of ρ with upper and lower bounds.

Another practical meaning of the upper bound is that it represents the case in which the encoder is not aware of the frame correlation and the decoder does not use this information to decode the frame. As expected, the frame correlation is a useful information to keep the video distortion low, and the gain slightly changes from one distribution of ρ to another. In fact, what matters is the available channel capacity which dictates the degrees of freedom of the system: the more constrained the channel capacity, the lower the average distortion measured and the higher the gain compared to frame correlation-agnostic encoder-decoder schemes.

In Figure 5.4 we show the impact of the prediction window size on the system performance. A prediction window size of 2 achieves almost the maximum gain of the system in the static scenario, and larger window sizes perform exactly the same. The small gain achieved in terms of overall distortion is due to the limited range of options available in the system, but still gives some insights into the problem of dimensioning the window size. Specifically, the rather small set of available frame correlation values could be extended to fully exploit the foresighted correlation, at the cost of a higher computational complexity associated to the MDP.

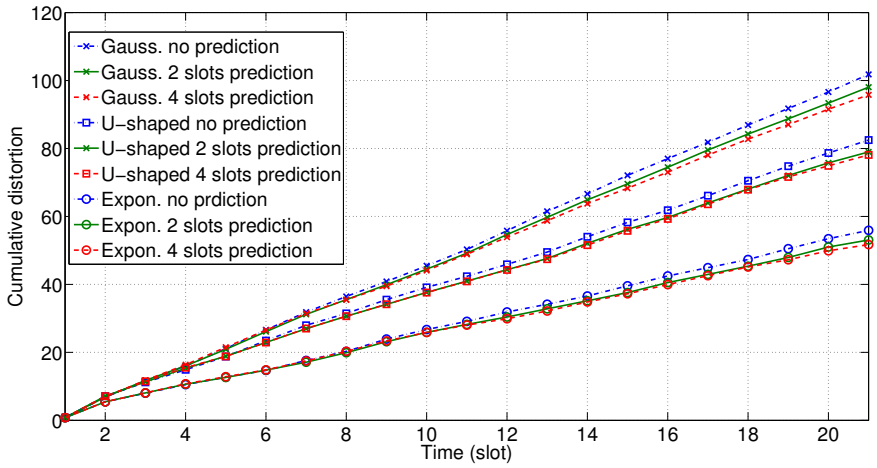

Figure 5.4: Impact of the prediction window size on the distortion.

In Figures 5.5 and 5.6 we report the number of paths computed by the system when varying the prediction window size and the channel capacity,

and the number of links of the optimal path when varying the channel capacity and the size of the simulation horizon, i.e., the number of available states in the corresponding MDP. Note that the computational complexity exponentially increases with the window size, as expected. For the static channel scenario the performance achieved when using a prediction window larger than one is not worth the associated complexity.

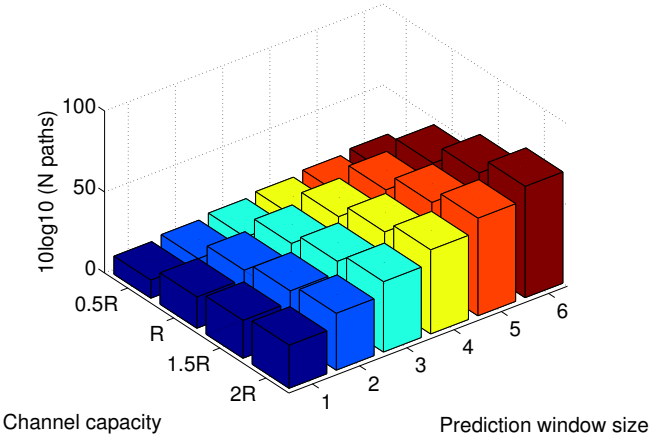

Figure 5.5: Complexity of the MDP in terms of number of paths when varying the prediction time window and the channel capacity.

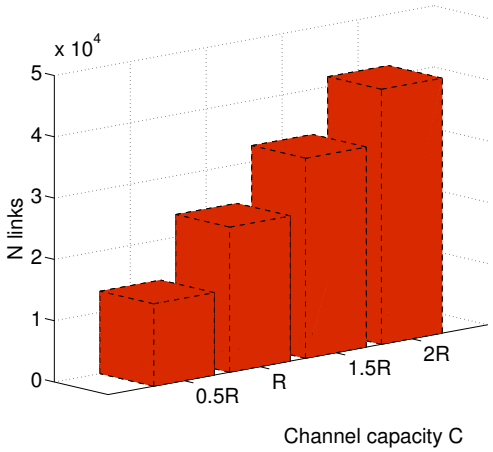

Figure 5.6: Complexity of the MDP in terms of number of links.

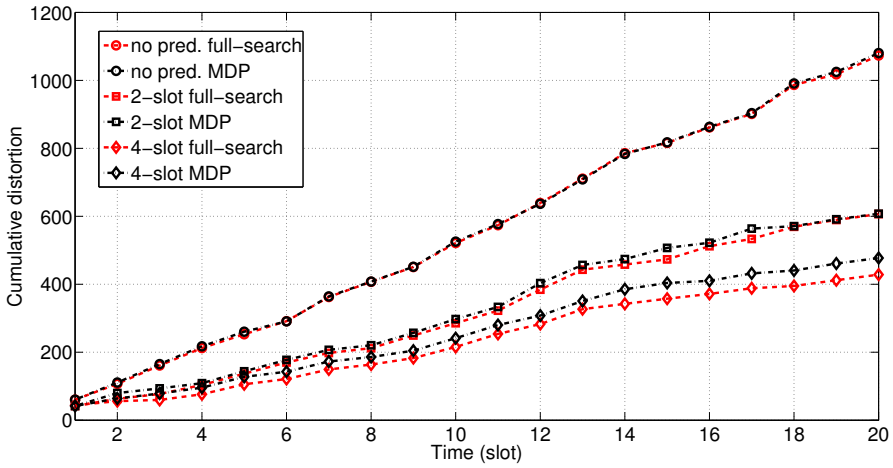

Figure 5.7: Dynamic channel scenario: impact of the prediction window size on the distortion for a GoP of 20 and comparison with the full-search algorithm.

In Figure 5.7 we show the impact of the prediction window size on the distortion in a dynamic channel scenario. We pick ρ normally distributed, as in Figure 5.2, in the set $\{0, 0.3, 0.6, 1\}$ and model the channel conditions as a random variable uniformly distributed in the set $\{0, R, 2R\}$. The results are averaged over 100 runs. Opposite to the static channel scenario, the joint foresighted information on frame correlation and channel conditions makes it possible to reduce the distortion, for a GoP of size 20 (the length of the simulated horizon), by at least half that obtained without prediction. Moreover, a window size of 4 further reduces the distortion by up to $\sim 65\%$ compared to the no prediction case due to the fact that the encoding and scheduling choices at time t have an impact on the next two following slots, whereas a window size of 2 simply accounts for the very next slot impacted by the action taken in the current slot. We compare the performance of our framework with that of a full-search algorithm which computes the optimal decisions for continuous values of ρ, whereas the MDP, which is solved via dynamic programming, samples those values into the aforementioned set of discrete values of ρ to find the optimal path. However, for the sake of comparison, the MDP computes the distortion based on the continuous values

of ρ. We observe in Figure 5.7 that the larger the prediction window size, the finer the estimation of the MDP, and the lower the cumulative distortion over the GoP. Note that the gaps between the simulation results using the MDP and the full-search algorithm are negligible on average, which confirms the reasonable level of accuracy of the MDP.

Chapter 6

Conclusions

In this work we formalized the problem of designing an efficient coding and scheduling procedure for video streaming, assessed in terms of distortion and delay, from a mathematical perspective and we proposed an analytical solution using simplified conditions. From this model we designed a simulation framework that computes the optimal encoding rate and scheduling decisions based on the frame correlation via an MDP. The goal of the optimization is to minimize the average long-term distortion of the video session under the wireless channel constraints. Our simulation results show that a simple delivery framework as the one we proposed, with limited frame correlation and channel information, is beneficial at the encoder for real-time applications as long as the complexity of the MDP can be kept at reasonable levels. When considering dynamic channel scenarios, the larger the prediction window size, the better the quality of the GoP delivered, but the worse the complexity. As future work, we plan to pursue analytical evaluations based on the proposed MDP framework in Chapter 4. Thus, we will implement the simplified proposed analytical solution considering different values for the deadline and the prediction window, inspecting the system behavior. Then, we foresee as future work the analysis of the trade-off between quality delivered and computational complexity of this solution with respect to that in Chapter 5. We will map the distortion of the GoP delivered to some QoE-based metric for a range of practical settings. We further plan to investigate the practical impact of the framework on upstreaming video services over

wireless networks by taking into account realistic channel traces. Moreover, we intend to accurately model the frame correlation information, that will be properly integrated into a more complex extension of the rate-distortion function, based on actual video encoding techniques, mobility of the users, and dynamics of the scene.

Appendix A

Markov chains

A stochastic process $\{X_t\}$ with the property that, for any t, given the value of X_t, the successive values of X_s for $s > t$ are independent of the values of X_u for $u < t$, is a *Markov process*. The probability of any future behavior of the process is not influenced by its past behavior, given the present state. In the case of a process which takes values on a finite or countable space and whose time is indexed, we have a *discrete-time Markov process*. In a mathematical perspective the Markov property can be formalized as follows.

$$P[X_{n+1} = j | X_0 = i_0, \ldots, X_n = i] = P[X_{n+1} = j | X_n = i] \tag{A.1}$$

$\forall n$ and $\forall i_0, \ldots, i_{n-1}, i, j$. A *Markov chain* is, therefore, defined by a (possible infinite) set of states linked by transition probabilities and refers to a sequence of random variables of the Markov process. In general, these probabilities are functions not only of the initial and final states, but also of the time of the transition as well. In the case of transition probabilities independent of the time variable, we say that the Markov chain has stationary transition probabilities. The *one-step transition probability* is defined as the probability that the chain is in state j at time $n + 1$ given that it was in state i at time n and is given in Equation (A.2).

$$P_{ij}^{n,n+1} = P[X_{n+1} = j | X_n = i] \tag{A.2}$$

Most of the chains have stationary probabilities and, thus, we let $P_{ij}^{n,n+1} = P_{ij}$,$\forall n$.

Usually these transition probabilities are grouped in a matrix called *transition probabilities matrix* $\mathbf{P}$:

$$\mathbf{P} = \begin{bmatrix} P_{00} & P_{01} & P_{02} & \cdots \\ P_{10} & P_{11} & P_{12} & \cdots \\ P_{20} & P_{21} & P_{22} & \cdots \\ \vdots & \vdots & \vdots & \\ P_{i0} & P_{i1} & P_{i2} & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix}$$

As we are dealing with probabilities, we have that

$$P_{ij} \geq 0 \quad \forall i, j = 0, 1, 2, \ldots \tag{A.3}$$

$$\sum_{j=0}^{\infty} P_{ij} = 1 \quad \forall i, j = 0, 1, 2, \ldots \tag{A.4}$$

Moreover it can be proven that a Markov chain is completely defined by its transition matrix and initial state X_0 [36]. In Figure A.1 we show an example of a finite-state Markov chain with five states and with respective transition probabilities.

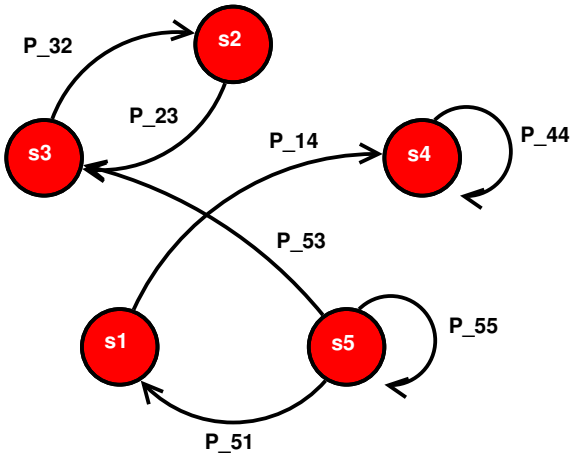

Figure A.1: Example of Markov chain.

Appendix B

Matlab code

In this appendix we report some parts of Matlab code used to implement the algorithms described in the thesis.

B.1 Markov chain - static channel

```
% Piece of Matlab code for building the chain;
% D = set of all possible distortions;
% rho_range = set of all possible correlations;
% rate_range = set of all possible encoding rates;
% rate_range_tx = set of all possible transmission rates;

...

% Create all possible states
for d=1:length(D)
    for rh=1:length(rho_range)
        for r1=1:length(rate_range_tx)
            for r=1:length(rate_range)
                for rh_1=1:length(rho_range)
                    if (rate_range_tx(r1)<=rate_range(r))
                        states(i).D=D(d);
                        states(i).rho_k=rho_range(rh);
                        states(i).rho_k_1=rho_range(rh_1);
                        states(i).R1_prev=rate_range_tx(r1);
                        states(i).R_prev=rate_range(r);
                        i=i+1;
                    end
                end
            end
        end
    end
```

```
end
states=states(1,1:i-1);

% Create all possible transitions

j=1;
for r1=1:length(rate_range)
    for r2=1:length(rate_range_tx)
        for r3=1:length(rate_range_tx)
            if (rate_range(r1)>=rate_range_tx(r2) ...
                    && (rate_range_tx(r2)+rate_range_tx(r3))<=C)
                Pi(j).R=rate_range(r1);
                Pi(j).R1=rate_range_tx(r2);
                Pi(j).R2_prev=rate_range_tx(r3);
                j=j+1;
            end
        end
    end
end
Pi=Pi(1,1:j-1);

% Determine feasible transitions

for from_index=1:i-1
    %disp(from_index);
    from=states(from_index);
    for pi_index=1:j-1
        pi=Pi(pi_index);
        if (pi.R2_prev==(from.R_prev-from.R1_prev) ...
                && pi.R1<=(C-from.R_prev+from.R1_prev))
            for rho_k_current=rho_range
                next_st.D=distortion(from.rho_k_1,from.R_prev);
                next_st.rho_k=rho_k_current;
                next_st.rho_k_1=from.rho_k;
                next_st.R1_prev=pi.R1;
                next_st.R_prev=pi.R;
                to_index=getStateIndex(next_st,states);
                transMtx(from_index,to_index)=1;
                PiMtx(from_index,to_index)=pi;
            end
        end
    end
end
```

Listing B.1: Markov chain - static channel

B.2 Markov chain - fading channel

```
% Piece of Matlab code for building the chain;
% D = set of all possible distortions;
% rho_range = set of all possible correlations;
% rate_range = set of all possible encoding rates;
% rate_range_tx = set of all possible transmission rates;

...

% Create all possible states
i=1;
for d=1:length(D)
    for rh=1:length(rho_range)
        for r1=1:length(rate_range_tx)
            for r=1:length(rate_range)
                for rh_1=1:length(rho_range)
                    for C=1:length(C_range)
                        if (rate_range_tx(r1)<=rate_range(r))
                            states(i)=setState(D(d), ...
                                      rho_range(rh), ...
                                      rho_range(rh_1),...
                                      rate_range_tx(r1), ...
                                      rate_range(r),C_range(C));
                            i=i+1;
                        end
                    end
                end
            end
        end
    end
end
states=states(1,1:i-1);

% Create all possible transitions
j=1;
for r1=1:length(rate_range)
    for r2=1:length(rate_range_tx)
        for r3=1:length(rate_range_tx)
            if (rate_range(r1)>=rate_range_tx(r2))
                Pi(j).R=rate_range(r1);
                Pi(j).R1=rate_range_tx(r2);
                Pi(j).R2_prev=rate_range_tx(r3);
                j=j+1;
            end
        end
    end
end
```

```
Pi=Pi(1,1:j-1);

% Determine feasible transitions
for from_index=1:i-1
    from=states(from_index);
    for pi_index=1:j-1
        pi=Pi(pi_index);
        if (pi.R2_prev<=from.C_k && pi.R1<=(from.C_k-pi.R2_prev))
            for rho_k_current=rho_range
                for c=C_range
                    if (pi.R2_prev~=(from.R_prev-from.R1_prev))
                        next_st=setState(distortion(0,0), ...
                                rho_k_current,from.rho_k, ...
                                pi.R1,pi.R,c);
                    else
                        next_st=setState(distortion( ...
                                from.rho_k_1, ...
                                from.R_prev),rho_k_current, ...
                                from.rho_k,pi.R1,pi.R,c);
                    end
                    to_index=getStateIndex(next_st,states);
                    transMtx(from_index,to_index)=1;
                    PiMtx(from_index,to_index)=getPiIndex(pi,Pi);
                end
            end
        end
    end
end
```

Listing B.2: Markov chain - fading channel

B.3 Minimum distortion path

```
function [ pi_opt,min_dist,paths ] = getMinDist( curr_state, ...
                    states,transMtx,PiMtx, ...
                    Pi,rho_input,C_input )

% pi_opt = minimum distortion path;
% min_dist = distortion value with pi_opt;
% paths = number of different path explored.

curr_state_index = getStateIndex(curr_state,states);
T_win = length(rho_input);
next_states = find(transMtx(curr_state_index,:));
paths=0;

% Determine next admissible states
j=0;
next_allowed=zeros(1,length(next_states));
for i=1:length(next_states)
    next_index=next_states(i);
    next = states(next_index);
    if (next.rho_k == rho_input(1) && next.C_k==C_input(1))
        j=j+1;
        next_allowed(j)=next_index;
    end
end
next_allowed =next_allowed(1:j);

if (isempty(next_allowed))
    min_dist=200;
    pi_opt=getPiMtx(1,1);
    disp('Error: Unable to find the next state.');
    return
end

pi_min_next=getPiMtx(1,1);
min=Inf;
pi_opt=Pi(1,PiMtx(curr_state_index,next_allowed(1)));
next_opt=0;
pi_min=pi_opt;
if (T_win==1)
    for next_index=next_allowed
        pi=Pi(1,PiMtx(curr_state_index,next_index));
        D_k_1 = states(next_index).D;
        if (D_k_1 < min)
                min = D_k_1;
                pi_opt = pi;
                next_opt=next_index;
```

```
        elseif (D_k_1==min)
            if ((curr_state.rho_k==1 && pi.R<pi_opt.R) || ...
                (curr_state.rho_k~=1 && curr_state.C_k~=0 ...
                && pi.R1>=pi_opt.R1 && pi.R>=pi_opt.R) || ...
                (curr_state.rho_k~=1 && curr_state.C_k==0 ...
                && pi.R<pi_opt.R))
                min = D_k_1;
                pi_opt = pi;
                next_opt=next_index;
            end
        end
    end
    min_dist=curr_state.D+states(next_opt).D;
    paths=length(next_allowed);
else
    for next_index=next_allowed
        pi=Pi(1,PiMtx(curr_state_index,next_index));
        [pi_next,min_next,paths_next]=getMinDist( ...
                        states(next_index), ...
                        states,transMtx,...
                        PiMtx,Pi, ...
                        rho_input(1,2:length(rho_input)), ...
                        C_input(1,2:length(C_input)));
        paths=paths+paths_next;
        if (min_next < min)
                min=min_next;
                pi_min_next=pi_next;
                pi_min=pi;
        elseif (min_next==min)
            if ((curr_state.rho_k==1 && pi.R<pi_opt.R) || ...
                (curr_state.rho_k~=1 && curr_state.C_k~=0 ...
                && pi.R1>=pi_opt.R1 && pi.R>=pi_opt.R) || ...
                (curr_state.rho_k~=1 && curr_state.C_k==0 ...
                && pi.R<pi_opt.R))
                min=min_next;
                pi_min_next=pi_next;
                pi_min=pi;
            end
        end
    end
    pi_opt=[pi_min,pi_min_next];
    min_dist=curr_state.D+min;
end
end
```

Listing B.3: Recursive function for the minimum distortion

Bibliography

[1] G. Orwell, *Nineteen Eighty-Four*. London: Secker and Warburg, 1949.

[2] Cisco, "Visual networking index: Global mobile data traffic forecast update," February 2014.

[3] ——, "Video aware wireless networks (VAWN) research program." [Online]. Available: http://www.cisco.com/web/about/ac50/ac207/crc_new/vawn/index.html

[4] X. Zhu and B. Girod, "Video streaming over wireless networks," in *Proceedings of European Signal Processing Conference,(Poznan, Poland*, 2007, pp. 1462–1466.

[5] D. Munaretto, T. Melia, S. Randriamasy, and M. Zorzi, "Online path selection for video delivery over cellular networks," in *Globecom Workshops (GC Wkshps), 2012 IEEE.* IEEE, 2012, pp. 1367–1372. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6477782

[6] M. Pesce, D. Munaretto, and M. Zorzi, "A Markov decision model for source video rate allocation and scheduling policies in mobile networks," in *2014 13th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net'14)*, Piran, Slovenia, june 2014, pp. 119–125.

[7] K. Sayood, *Introduction to data compression.* Morgan Kaufmann, 2012.

[8] ITU-R RECOMMENDATION, BT.601-7, "Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios," Tech. Rep., 2011.

[9] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," vol. 13, no. 7, pp. 560–576, 2003. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1218189

[10] A. Rao, A. Legout, Y.-s. Lim, D. Towsley, C. Barakat, and W. Dabbous, "Network characteristics of video streaming traffic," in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*. ACM, 2011, p. 25. [Online]. Available: http://dl.acm.org/citation.cfm?id=2079321

[11] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of H.264/AVC," *IEEE Trans. Circuits Syst. Video Technology*, vol. 17, pp. 560–576, 2003.

[12] C. Krasic, K. Li, and J. Walpole, "The case for streaming multimedia with TCP," in *Interactive Distributed Multimedia Systems*. Springer, 2001, pp. 213–218. [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-44763-6_22

[13] C. Patrikakis, N. Papaoulakis, C. Stefanoudaki, and M. Nunes, "Streaming content wars: Download and play strikes back," in *User Centric Media*. Springer, 2010, pp. 218–226. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-12630-7_26

[14] D. Munaretto, D. Jurca, and J. Widmer, "Broadcast video streaming in cellular networks: An adaptation framework for channel, video and AL-FEC rates allocation," in *Wireless internet conference (WICON), 2010 the 5th annual ICST*. IEEE, 2010, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5452675

[15] D. Munaretto, L. Scalia, T. Soni, and J. Widmer, "Reliable broadcast streaming over 802.11 WLANs with minimum channel usage," in *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*. IEEE, 2009, pp. 30–35. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5202387

[16] ITU-T RECOMMENDATION, E.800, "Terms and definitions related to quality of service and network performance including dependability," Tech. Rep., 1994.

[17] R. Jain, "Quality of experience," *MultiMedia, IEEE*, vol. 11, no. 1, pp. 96–95, Jan 2004.

[18] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/el_20080522

[19] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hossfeld, "Impact of frame rate and resolution on objective QoE metrics," in *Workshop on Quality of Multimedia Experience (QoMEX)*, Trondheim, Norway, June 2010.

[20] M. Zanforlin, D. Munaretto, A. Zanella, M. Zorzi, "SSIM-based video admission control and resource allocation algorithms," *IEEE WiOpt (WiVid) 2014*, May 2014.

[21] ITU-T RECOMMENDATION, P.800, "Subjective video quality assessment methods for multimedia applications," Tech. Rep., 1999.

[22] J. Hu and J. D. Gibson, "New rate distortion bounds for natural videos based on a texture dependent correlation model in the spatial-temporal domain," in *46th Annual Allerton Conference on Communication, Control, and Computing*, Urbana-Champaign, IL, Sept. 2008.

[23] "The moving picture experts group." [Online]. Available: http://mpeg.chiariglione.org/

[24] "Advanced Video Coding for Generic Audiovisual Services," *ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC*.

[25] F. Fu and M. van der Schaar, "Structure-aware stochastic control for transmission scheduling," *IEEE Trans. on Vehicular Technology*, vol. 61, pp. 3931 – 3945, Nov. 2012.

[26] ——, "A Systematic Framework for Dynamically Optimizing Multi-User Wireless Video Transmission," *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 308 – 320, Apr. 2010.

[27] ——, "Structural Solutions for Dynamic Scheduling in Wireless Multimedia Transmission," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 22, pp. 727 – 739, May 2012.

[28] C. Chen, R. W. Heath, A. C. Bovik, and G. de Veciana, "A Markov Decision Model for Adaptive Scheduling of Stored Scalable Videos," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 23, pp. 1081 – 1095, June 2013.

[29] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression.* Prentice Hall, 1971.

[30] H.-J. Lee, T. Chiang, and Y.-Q. Zhang, "Scalable rate control for MPEG-4 video," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, pp. 878 – 894, Sep. 2000.

[31] D. Munaretto, D. Jurca, and J. Widmer, "A Fast Rate-Adaptation Algorithm for Robust Wireless Scalable Streaming Applications," in *IEEE WiMob 2009*, Marrakech, Morocco, Oct. 2009.

[32] ——, "A resource allocation framework for scalable video broadcast in cellular networks," *Mobile Networks and Applications*, vol. 16, no. 6, pp. 794–806, 2011. [Online]. Available: http://link.springer.com/article/10.1007/s11036-010-0270-6

[33] Z.Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, pp. 600 – 612, Apr. 2004.

[34] Z.Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Sig. Proc.: Image Comm.*, vol. 19, pp. 121 – 132, Apr. 2004.

[35] D. Bertsekas, *Dynamic Programming and Optimal Control.* Athena Scientific, 2007.

[36] M. Pinsky and S. Karlin, *An introduction to stochastic modeling.* Academic press, 2010.