University of Bologna

Department of Electrical, Electronic, and Information Engineering Guglielmo Marconi

XXV PhD. Course in Electronics, Computer Science, and Telecommunications

Reliable Broadcasting and Streaming of Multimedia Content

by

Valentina Pullano

Coordinator:
Prof. **A. Vanelli-Coralli**

Supervisors:
Prof. **G.E. Corazza**
Prof. **A. Vanelli-Coralli**

March 2013

To my family,
undisputed guide
of my existence....
and to Laura,
which recently left this world
leaving a gap
that will never be filled...

"The only limit to our
realization of tomorrow
will be our doubts of today."

Franklin D. Roosevelt

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ABEL	Average Burst Error Length
ACK	Acknowledgment
ADSL	Asymmetric Digital Subscriber Loop
ARQ	Automatic Repeat reQuest
BEC	Binary Erasure Channel
BL	Base Layer
CGS	Coarse Grain Scalability
CRC	Cyclic Redundancy Check
CS	Check Sum
DSCQS	Double Stimulus Continuous Quality Scale
DP	Dependency Path
DSL	Digital Subscriber Loop
DTM	Discrete Multi-tone Modulation
DVB-H	Digital Video Broadcasting- Handheld
DVB-SH	Digital Video Broadcast ing- Satellite to Handheld
DVB-T	Digital Video Broadcasting- Terrestrial

EC	Error Concealment
EEP	Equal Error Protection
EL	Enhancement Layer
ER	Error Resilience
FEC	Forward Error Correction
FEC DB	Forward Error Correction Data Block
FEC SB	Forward Error Correction Source Block
FER	Frame Error Rate
FR	Full Reference
FTTC	Fiber To The Curb
FTTH	Fiber To The Home
GEC	Gilbert-Elliott Channel
GF	Galois Field
GPS	Global Positioning System
H.264/AVC	H.264/Advanced Video Coding
HARQ	Hybrid Automatic rePeat Request
HEVC	High Efficiency Video Coding
HDTV	High Definition Television
HFC	Hybrid Fiber-Coax
HVS	Human Visual System
IAT	Inter Arrival Time
IL	Interleaver Length
IPTV	Internet Protocol Television
ISO	International Standardization Organization

ITU	International Telecommunication Union
JND	Just Noticeable Difference
JVT	Joint Video Team
JSCC	Joint Source and Channel Coding
LA-FEC	Layer Aware - Forward Error Correction
LA-FEC UI	Layer Aware - Forward Error Correction with Unequal Time Interleaver
LDPC	Low Density Parity Check
LDS	Lower Dependency Set
LL-FEC	Link Layer - Forward Error Correction
LMS-ITS	Land Mobile Satellite - Intermediate Tree Shadow
LOS	Line of Sight
LT	Luby Transform
LTE	Long Term Evolution
MB	Macro Block
MDLA-FEC	Multi-Dimensional/Multi-Layer Aware Forward Error Correction
MGS	Medium Grain Scalability
MOS	Mean Opinion Score
MPEG	Moving Picture Expert Group
MSE	Mean Square Error
MTU	Maximum Transmission Unit
MVC	Multiview Video Coding
NACK	Negative Acknowledgment
NALU	Network Abstraction Unit

NR	No Reference
OP	Operation Point
PEIN	Prolonged Electrical Impulse Noise
PER	Packet Error Rate
PHY	Physical layer
PSNR	Peak Signal to Noise Ratio
PSTN	Public Switched Telephone Network
QoE	Quality of Experience
QoS	Quality of Service
QP	Quantization Parameter
RAP	Random Access Point
REIN	Repetitive Electrical Impulse Noise
RR	Reduced Reference
RS	Reed Solomon
RTCP	Real-Time Control Protocol
RTP	Real-Time Transmission Protocol
RTSP	Real-Time Streaming Protocol
SDTV	Standard Television
SHINE	Single Isolated Impulse Noise
SL	Single Layer
SNR	Signal to Noise Ratio
ST-FEC	Standard- Forward Error Correction
SVC	Scalable Video Coding
TCP	Transmission Control Protocol

TS	Transport Stream
UDP	User Datagram Protocol
UDS	Upper Dependency Set
UEP	Unequal Error Protection
UI	Unequal Interleaver
UL	Upper Layer
UL-FEC	Upper Layer- Forward Error Correction
VCEG	Video Coding Expert Group
VLC	Variable Length Coding
VoD	Video on Demand
VQEG	Video Quality Expert Group
WPSNR	Windowed Peak Signal to Noise Ratio
Y-PSNR	Luminance - Peak Signal to Noise Ratio

INTRODUCTION

Motivation and Goals

Year 2000: television is analogue, Internet connection speed is very low, cell phones are just phones and the biggest innovation is represented by the introduction of SMS messages. Video cameras use analogue tape media, VHS video tapes are still largely employed for recording video contents.

Year 2012: digital television diffusion is growing and coverage expanding, bringing about hundreds of channels with high definition quality and interactive services as benefits. 3D movies are appreciated worldwide and 3DTVs are appearing in many houses. Internet TVs are entering the market and so-called ”smart” devices are changing the way in which TV sets are used. Users do no stand in front of the TV as passive viewers, they can play, interact, buy and pay by credit cards all what they need. Internet connection speed is consistent and it enables the use of video-based web applications. Cell phones are multitasking devices no longer used only for calling, they can be used to send emails, for instantaneous messaging, for video recorder and sharing, and are largely employed for enjoying video contents. DVDs and Blue-Ray discs are used for recording/storing movies and any other kind of data, a huge amount of data (GBytes) can be stored on few millimeters of thickness.

In only 12 years, a multitude of multimedia applications have entered our daily life determining a significant change in the way in which we communicate, we keep ourselves up to date, we keep in touch, we spend our free time. Skype video calls, for example, are a clear representative of this trend. They give people the feeling of being always together even if thousands of kilometers apart.

Considering all the above, it is possible to affirm that, in this period of significant technology shifts, the emergence of digital video represents one of the most prominent revolutions. Broadcasting and streaming applications diffusion has been massive. The number of services based on video communications has been growing on a daily basis and final users have not been passive subjects anymore, they can select in a conscious way what to watch and which is the quality they expect to enjoy.

However, when video transmission communications are targeted, the aspects to be taken into account are manifold. Nonetheless, the reliability of the transmission is without any doubt one of the most important.

The term **reliable**, in the context of video transmission, addresses the robustness of the transmission against channel imperfections, that generally lead to errors and losses and may result in undecodable video sequences at the decoder side or in unacceptable quality at the receiver end. The reliability of a video stream is function of several aspects, ranging from the actual state of the underlying transmission channel, to the prediction mechanisms employed in the compression stage, to the kind of error protection mechanisms enabled, and not only.

Reliable video transmission techniques, also known as *error control mechanisms*, are thus fundamental for high quality video service provisioning and, at the same time, extremely challenging. Error control mechanisms can act on multiple stages of a typical communication system: at the source encoder (error refinement), at the channel encoder (forward error correction), at the source decoder (error concealment), cross-layers (hybrid design). In addition, also techniques acting during the transmission scheduling phase, such as time interleaver schemes, may fall in this category.

Two kinds of errors typically affect a transmission system: *Bit Errors* and *Packet Erasures*. The former are generally ascribable to physical channel's imperfections and may consist of bits insertion/deletion and bits inversion. Their effect is function of both the compression scheme employed and the real visual content of transmitted data and it can range from negligible to objectionable. The latter can be caused by packet loss/dropp in packet networks and/or generated as a consequence of physical layer errors, since single bit errors may led to a group of consecutive undecodable bits. Their effect is much more destructive than bit errors and for these reasons ad-hoc mechanisms have to be designed to cope with them.

This thesis presents the outcomes of the studies carried on during my PhD, focused on the reliable transmission of multimedia contents in streaming and broadcasting applications, tailoring especially video contents.

The design of efficient error-control mechanisms, able to enhance video transmission systems' reliability is targeted. Cross-layers channel coding techniques able to cope with bit errors as well as packet erasures, are considered. Unequal time interleaver mechanisms are another viable solution for controlling the effects of errors and erasures. They act on the time diversity of the data flow, enhancing the robustness against the typical kinds of channel impairments. In this context, it is not possible to leave out of consideration the nature of the factors which affect the physical layer channel. Ad-hoc noise modeling can be extremely suitable in the evaluation of FEC schemes performances. In addition, it is also fundamental to have an insight into the quality perceived by the real consumers of video service applications.

During my PhD. I have dealt with all these issues, proposing techniques and novel ideas which have contributed to the definition of viable solutions in the field of video broadcasting and streaming applications. The applicability and the value of these techniques will be proved considering practical constraints and requirements of real system implementations.

Thesis Outline

The dissertation is organized in five chapters and two appendixes.

In Chapter 1, the basis of coding theory are given. A classification of the mechanisms for error protection acting at different stages of the transmission chain is provided along with an overview of channel codes. Finally, a classification of the quality evaluation techniques is given, along with the introduction of the Peak Signal to Noise Ratio (PSNR) quality metric.

Chapter 2 tackles the performance analysis of the Layer-Aware Forward Error Correction with Unequal Time Interleaver (LA-FEC UI) scheme in broadcasting and streaming applications, targeting mobile receivers. This activity has been carried out during my internship period in collaboration with Dr. Hellge Cornelius and the other designers of the theoretical concept at the basis of this study. The effectiveness of the solution is proved by means of graphical and tabular results. The analysis presented in this chapter were partially presented in [1], [2] and are going to appear in [3].

In Chapter 3, the Multi-Dimensional/Multi- Layer Aware Forward Error Correction (MDLA-FEC) scheme is introduced as an extension of the LA-FEC approach to the multidimensional case. A mathematical description of the resulting decoding structure is provided, referring to a specific view setup based on a 9-layer multidimensional media flow. The conducted theoretical analysis entails both broadcasting TV and IPTV applications. The evaluation presented in this chapter is going to appear in [6].

In Chapter 4, a mathematical model for deriving the correction probability of an UL-FEC scheme after transmission over a channel characterized by randomly distributed error bursts with fixed length, is presented. The transmission channel affected by fixed-length randomly distributed error bursts is modeled as an B_L+1 state Markov chain. The combinatorial analysis, that has been conducted in order to find all the error blocks distribution combinations over the delivered data block still leading to decodable data, is described. The resulting decoding probability is calculated applying the provided mathematical model. The analysis presented in this chapter is going to appear in [4]. Also this activity has been carried on during internship.

In Chapter 5, a method able to detect the position of lost frames within a received uncompressed video sequence is provided. The presented methodology, named Windowed PSNR (W-PSNR), is additionally able to detect the sequence alignment of the received sequence with the original unimpaired one and to evaluate the PSNR values of both the shorter-received sequence and of the post-alignment one, thus giving an insight into the incurred quality decrease. The reported analysis were presented in [5].

Finally, in Appendix 1 the most important video parameters and characteristics are described and an overview of the basic techniques used for video communication over different communication networks and video-oriented Internet protocols is provided. Appendix 2 gives the video compression essentials. The state-of-the-art H.264/AVC video compression standard is introduced along with its main extensions.

Original Contributions

The activities performed during the three years of this doctorate study led me to obtain original scientific contributions in several fields.
Regarding techniques of reliable error corrections, the main contributions are the following:

- Multi-platform implementation of the Unequal Time Interleaver mechanism on the top of the LA-FEC /SVC structure.

- Analysis of the transmission scheduling to be applied to layered video flows for guaranteeing fast access to the video services while enhancing the corresponding robustness.
- Evaluation of the benefits of UL-FEC mechanisms in terms of failure decoding probability and measured video quality.
- Analysis of physical layer impulsive noise impact to higher layers performance, expressed in terms of packet erasure rate.
- Design of a novel channel model for emulating fixed length error bursts.
- Design of a mathematical model for deriving the correction probability of an UL-FEC scheme after transmission over a channel characterized by randomly distributed error bursts with fixed length.
- Design of the MDLA-FEC mechanism for protecting multi-layer/multi-dimensional media stream.
- Mathematical representation of the decoding structure of the MDLA-FEC approach for broadcasting and IPTV applications.

Regarding techniques of video quality evaluations, the main contributions are the following:

- Design of a novel mechanism for objectively evaluating the quality of a loss-affected video sequence based on a sliding window mechanisms.

CHAPTER 1

TECHNIQUES FOR RELIABLE VIDEO TRANSMISSION AND QUALITY EVALUATION

1.1 Introduction

The last decades have been characterized by several technology shifts, among which the most significant one is represented by the emergence of multimedia contents and applications as a fundamental aspect of daily life. Thanks to the development of new technologies as well as the employment of new infrastructures it is nowadays possible to enjoy multimedia contents everywhere and in every moment. Indeed, on one side the Internet has changed the way in which we can obtain information, and on the other, the evolution of video communications have made contents much more attractive and impressive. With the explosion of digital video applications, a very productive industry has developed and expanded, several new companies and niche markets have emerged as well, making the fruition of multimedia contents much easier, more attractive and interactive. As a consequence, digital video technologies are nowadays involved in several fields, ranging from applications for video telephony and conferences, to applications for cultural heritage, education, medicine, etc.

High-definition television (HDTV), three-dimensional television (3DTV) and HD DVD standard (blue-ray) are now becoming widely available and in the next few years the age of analog television will become just a far memory. One of the most critical aspects of video broadcasting and streaming is the ability to protect the transmitted information from errors/losses due to propagation. To this end, different channel coding techniques have been developed and in the following their main characteristics will be detailed. In addition, other possibilities exist for limiting the effect of errors and losses, i.e. *error resilience* and *error concealment* techniques, applied at the source encoder and decoder respectively. Therefore, with the expression *reliable video transmission techniques*, the set of techniques able to control the presence and to act on the effect of transmission errors and losses are addressed. These techniques may act at different stages of the transmission chain, often in combination to each other. Channel coding techniques provide methods for protecting the information flow over its transmission path and enable mechanisms for correcting/recovering information errors/losses, by means of data retransmission or by adding structured redundancy. Channel coding methods and techniques are an important field of the information theory. These techniques can be classified following different criteria, a first important classification is in function of the ISO/OSI pipeline level where the channel code is applied (e.g. physical layer (PHY) or upper layers (UL)) and, as a consequence, of the elementary units processed by the channel encoder/decoder. Techniques of error correction performed at the physical layer comprise the classical channel coding techniques, where the elementary units to be processed are binary digits. Albeit some extensions to non-binary schemes exist (i.e. Reed Solomon codes), physical layer codes are typically **bit-oriented**. Upper layer codes, instead, use packets (groups of bits /symbols) as elementary units and for this reason they are commonly referred as **packet-oriented** channel codes. UL codes are applied to upper layers of the protocol stack and do not exclude the physical layer protection, but act as a form of additional protection designed to tackle with all the situations in which the employed physical layer protection is not enough. Accordingly with the given classification, if PHY layer codes are considered, bit/symbol insetion/inversion/deletion are the errors to tackle, while in case of UL codes, packet erasures have to be addressed. For this reason, UL codes can be also referred as *erasure codes.*

Being the communication in real system environments unavoidably affected by errors/interferences/losses, the benefits introduced by the use of UL codes is easy to understand. In fact, UL codes are suitable for protection against long erasure bursts, as the ones which characterize mobile communication links. The techniques analyzed in the rest of this dissertation, largely employ UL codes, albeit some of them have been proved to be suitable for both physical and upper layer protection. This choice has been driven by the large flexibility introduced by UL solutions - fundamental requirements of emerging systems. In fact, packet codes are implemented in software, therefore easy to reconfigure. Furthermore, UL codes allow an easy upgrade of network terminals and receivers because they can be used in integrating different standards.

Considering all the above, it is clear how essential video compression and reliable video transmission techniques are to all of those video-based applications and associated markets. The state-of-the-art video compression standard, the H.264/AVC, and its extensions are described in Appendix B, while in this chapter, techniques for reliable transmission of multimedia contents are addressed. In order to allow the reader to familiarize with the most significant concepts at the basis of this PhD thesis, an overview of the possible techniques able to improve the robustness of a multimedia flow against communication channels' imperfections are provided. For a complete analysis of the topic, both error correction and error concealment/resilience techniques are described, even though the research has been mainly focused on channel coding techniques. In addition, an overview of the main classes of methodologies currently available to evaluate video quality is provided.

1.2 Coding theory essentials

In 1948, Claude Elwood Shannon wrote a seminal paper [12] that can be considered as a milestone of telecommunication studies and that assured him the paternity of the Information Theory.

As a general concept, a communication system has the aim of transmitting data from a source to a destination by means of a communication medium, generally referred as channel. Shannon did several steps in describing mathematically a general communication system. Firstly, in the *source-channel separation theorem*, he shown that the two operations of data compression and protection can be performed separately without any loss of optimality. Afterwards, with his *channel coding theorem*, Shannon affirmed that reliable transmission is possible as long as the transmission rate is less than the channel capacity, which means that the channel capacity represents the maximum rate (bits per channel use) at which information can be transmitted reliably over a given channel. This outcome has been a great motivator for scientist of all around the world to investigate how to design new efficient coding schemes able to reach information rates close to the capacity, while assuring reliable transmission. Unfortunately, Shannon did not provide a constructive proof of the channel coding theorem, indeed he just affirmed that by adding redundancy and transmitting to a rate lower then the channel capacity reliable transmissions can be performed, but he did not clarify how the redundancy has to be constructed.

In relation to error-control techniques, there are basically two mechanisms for adding redundancy: block coding and convolutional coding. Although recently a new class of codes, known as rateless codes, is widely emerging. The cardinal difference among block and convolutional codes is that block codes are memoryless - the current output of the channel encoder (symbol or packet) only depends on the current input - while convolutional codes have memory - the current output relies also on previous inputs.

In coding theory, **block codes** refers to a large and important class of error-correcting codes. Their name stems from the fact that they encode data in blocks. Examples of block codes are Reed-Solomon codes, Hamming codes, Golay codes, and many others. These examples also belong to the class of linear codes, and hence they are called *linear block codes.* In block coding, the information flow is rearranged into pieces - **messages** - of fixed length. Each message is encoded into a **codeword**, also known as **block**. At the receiver side, the decoder will employ some mechanisms in order to recover the original messages from the possibly corrupted received blocks. The performance and success of the overall transmission depends on the parameters of the channel and of the block code. Formally, a block code C of length n over a finite field F (known as Galois Field - GF) is any subset of F_n. In the following, F_q and $GF(q)$ will indistinctly indicate a Galois field of cardinality q. In the binary case, q = 2.

In general, a block code is indicated with $C(k, n)$, where k is the message length, n is the codeword length and $r = k/n$ is the code rate. A binary linear block code is encoded by means of a **generator matrix** (G) - of size (kxn). The encoding procedure consists of the product of the k-binary digits source blocks for the generator matrix and generates codewords of n-binary digits in length. When information bits are part of the codeword, either at the beginning or at the end, the code is **systematic**. **Non-systematic** in all other cases. Another alternative is to use the (mxn) parity check matrix (H). C is in this case defined as the set of n-tuples satisfying the linear system of parity check equations $c \cdot H^T = 0$. If H has full rank, all the parity check equations are linearly independent and $m = n - k$. Block codes are memoryless, this means that codewords are generated as a function of the current input only. The strength of a block code is measured as a function of its power in revealing errors (erasures in case of UL codes) and in correcting them. To this extent, an important parameter is the minimum distance (d_{min}) among two generic codewords belonging to a same code C. Indicating with w_H the Hamming weight of a vector x (number of non-zero elements), the Hamming distance of two codewords belonging to the same code C is the Hamming weight of their sum (or difference) mod 2. The minimum distance d_{min} of a block code C is the minimum of the Hamming distances between any two codewords belonging to C. The minimum distance plays a key role in determining the error detection and correction capability of the code, function of the considered underlying channel. A larger distance allows for more error correction and detection. A code with distance d_{min} allows the receiver to detect up to $d_{min} - 1$ transmission errors and is able to correct up to $(d_{min} - 1)/2$ errors. If more than $(d_{min} - 1)/2$ transmission errors occur, the receiver cannot uniquely decode the received word in general as there might be several possible codewords.

Convolution codes were first introduced by Elias [13] in 1955 as an alternative to block codes from which they differ in that the encoder contains memory and the n encoder outputs, at any given time instant, depends on the m previous input blocks. An (n, k, m) convolutional code can be implemented with a k-input, n-output linear sequential circuit with input memory m. Typically, n and k are small integers with k<n, while the memory order m should be big enough to guarantee low error/erasure probabilities. As for block codes, the code rate is defined as $r = k/n$ and it gives an insight into the potential code efficiency. Another important parameter is the *constraint length*, defined as $L = k * (m - 1)$, which indicates the number of bits in the encoding memory which affect the generation of the current output. The convolutional code structure is fully identified by its parameters. For example, an (n,k,m) code has m boxes representing the memory register, k input bits and n modulo-2 adders - one for each output bit - and some connection between the memory register and the adders - defined by the generator polynomials. The generation polynomial have to be carefully selected, because they do not always result in having good error protection properties.

If $k = 1$ we will talk about *mother codes* and the code structure is exactly like the one just presented. When k>1, the procedure for drawing the code structure changes. k sets of m boxes and n adders have to be drawn, then the adders can be connected to the memory registers using the coefficients of the generator polynomial of degree $k \cdot m$. However, this structure even if easy to understand and to draw, is not suitable for real system applications, for which other techniques, such as the

look up tables, are typically employed.
As for the block-code case, the convolutional code is **systematic** when the output bits contain an easily recognizable sequence of the input bits, **non-systematic** otherwise. For decoding a convolutional code several different approaches exist and they can be grouped into two main categories:

1. Sequential decoding, e.g. Fano algorithm [14];

2. Maximum Likelyhood decoding, e.g. the Viterbi algorithm [15].

The description of these decoding algorithm goes out of the scope of this dissertation, the interested reader is thus advised to relative references.

1.2.1 ARQ & FEC

In function of the targeted application, error detection and correction can be generally performed in two ways: retransmitting lost or corrupted information or adding redundancy by using a channel code. In the first case, an **Automatic Repeat reQuest** (ARQ) mechanism, known also as backward error correction, is employed. In the second case, a **Forward Error Correction** (FEC) code is used.

Automatic repeat reQuest communication systems are based on the detection of errors in coded blocks (frames) and on their retransmission when errors have been detected. In this case a two-way channel is needed in order to request retransmissions. For a given code, the error-detection capability possible in an ARQ system is higher than the error-correction capability of its FEC counterpart, because the error-control capability of the code is spent only on detection, while the correction requires not only the detection but also the localization of the errors. On the other hand, there is an additional cost in an ARQ system, which is the need for a retransmission link. There are also additional operations for the acknowledgement and repetition processes that reduce the transmission rate of the communication system. Each codeword (block) is stored in the transmitter buffer and then transmitted. This codeword can be affected by noise, so that at the receiving end the decoder evaluates if it belongs or not to the code. In the affirmative case, a positive acknowledgement (ACK) is transmitted by the receiver. Otherwise a negative acknowledgement (NACK) is sent for asking the retransmission of the corresponding block in the transmitter buffer. An ARQ system may then have a reduced transmission rate with respect to a FEC system as a result of the retransmission process. Three well known ARQ protocols are Stop-and-Wait ARQ, Go-Back-N ARQ, and

Selective Repeat ARQ. ARQ is suitable whenever the communication channel has varying or unknown capacity and it requires a mandatory return channel. This last constraint makes ARQ a valid solution for a limited number of services and applications. Therefore, the maintenance of buffers and timers for retransmissions makes this solution not valid for real-time and conversational services.

In the *Forward Error Correction* (FEC) case, data are encoded with an error-correction code prior to transmission. The receiver can employ the redundancy data sent along with real data to recover the original information. Receivers do not have to ask the sender for data retransmission, hence a return channel is not required. As FEC code, both block and convolutional codes can be used. FEC codes can be performed at the physical layer only or at both the physical and upper layers.

In addition, ARQ and FEC may be combined, generating the Hybrid Automatic Repeat-Request(HARQ). This scheme is able to correct minor errors without retransmission, and major errors via a request for retransmission. There are two basic approaches:

1. Messages are always transmitted with FEC parity data. A receiver decodes a message using the parity information (redundancy), and requests retransmission using ARQ only when parity data is not sufficient for successful decoding (identified through a failed integrity check).

2. Messages are transmitted without parity data. If a receiver detects an error, it requests FEC information using ARQ, and it reconstructs the original message exploiting the so-collected parity data.

In general, FEC is used for applications that require low latency, applications where the transmitter immediately forgets the information as soon as it is sent (when an error occurs, the original data is no longer available) and applications which do not have a return channel. Instead, applications that require extremely low error rates (such as digital money transfers) must use ARQ. A deeper analysis of this topic can be found in [16].

1.3 Upper Layer Channel codes

As already introduced, channel coding operations can be performed at both the physical layer (PHY) and upper layers of the protocol stack. Upper Layer Forward Error Correction (UL-FEC) codes can enhance the robustness of a data flow against errors and losses by providing additional protection on top of the physical layer. The elementary units to be processed are now packet of bits. For this reason, UL-FEC codes are commonly referred as *packet layer coding* or *packet level coding.*

Both binary and non-binary UL codes exist. In the non-binary case, the packet code is a linear block code $C(n,k)$ belonging to F_q and the channel seen by the packet level decoder is a packet erasure channel with packet size q. All the operations belong to F_q.
In case of binary codes, the ex-or operations are performed packet-wise rather than block/symbol-wise.
Although some classes of channel codes can be adopted as both PHY and UL codes, there are important differences to be aware of for an efficient design of UL codes. Encoded packets composing UL codewords are equipped with a Cyclic Redundancy Check (CRC) or a Check Sum (CS) which allow the receiver to detect erroneous symbols/packets. These packets are discarded by the UL decoder. Fig. 1.1 shows intuitively the packet level coding concept. k input packets are provided as input to the packet level encoder which returns $n > k$ packets, each of which is equipped with a CRC (or CS). During the transmission some of them get lost (erasures), hence the receiver got a number of packets $r \leq n$. The decoder acts on this subset of packets only to recover original data.

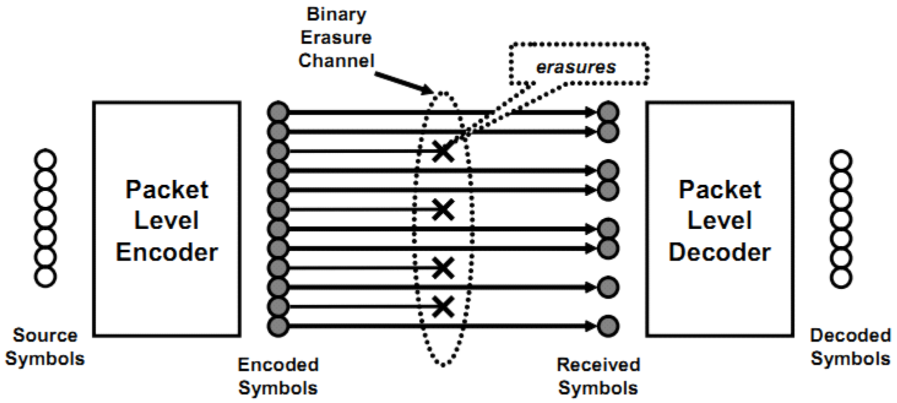

Figure 1.1: UL coding concept.

In the rest of this section, the most effective packet level codes are briefly presented.

1.3.1 Reed Solomon Codes

In 1960, Irving S. Reed and Gustave Solomon published a paper [17] describing a new class of error-correcting codes, known as Reed-Solomon (R-S) codes.
Reed-Solomon codes are non-binary cyclic codes, which generate sequences of m-bits symbols, where m is any positive integer greater than 2. Indicating with k the information block length and with n the codeword length, $RS(n, k)$ codes on m-bit symbols exist for all n and k such that $0 < k < n < 2^m + 2$.
The most conventional Reed Solomon code has $(n, k) = (2^m - 1, 2^m - 1 - 2t)$ where t represents the error correcting capability of the code. For instance, the parameters adopted by DVB-H technical specifications [18] are $m = 8$, $t = 32$, $n = 255$, and $k = 191$. As discussed before, the correction capability of a code is function of the minimum distance among its codewords. For non-binary codes, the distance between two codewords is defined as the number of symbols in which the sequences differ. More in detail, in Reed Solomon codes, the minimum distance is given by $d_{min} = n - k + 1$, as a consequence any combination of errors up to $t = \lfloor (d_{min} - 1)/2 \rfloor = \lfloor (n - k)/2 \rfloor$ can be corrected. This means that the decoder has $(n - k)$ redundant symbols to employ, which is twice the amount of correctable errors. For each error, one redundant symbol is used to locate the error, and another redundant symbol is used to find its correct value. When dealing with non-binary m-bits symbols codes, only a small fraction of possible n-tuples are codewords (i.e. 2^{km} of the large number 2^{nm}). This fraction decreases with increasing values of m. This means that when a small fraction of the n-tuple space is used for codewords, a large d_{min} can be achieved, upgrading code performance. Reed Solomon codes can be designed to have any redundancy and they are able to correct any set of $(n - k)$ symbol erasures within the block. A detailed analysis of encoding and decoding algorithms for R-S codes can be found in [16]. In particular, an efficient R-S decoding algorithm for erasure channels is provided in [19].

1.3.2 Fountain Codes

Fountain codes are a class of erasure codes which do not exhibit a fixed code rate and for this reason are also known as *rateless code.* They represent a big improvement for communications over the Internet, since they have been primarily designed for transmission over the Binary Erasure Channel (BEC), a well-established model for the Internet. The main positive aspect of fountain codes is that they are able to generate a potentially infinite number of encoding symbols starting from a given set of source symbols. In other words, a fountain code produces for a given set of input symbols $(x_1; ..; ..; ..; x_k)$ a potentially limitless stream of output symbols $(z_1; z_2; ..; ..; ...)$. Input and output symbols can be binary vectors of arbitrary length. Output symbols are generated by summing up a subset of input symbols randomly chosen. In order to enable the decoding, receivers are informed about the symbols generation path through packet headers or other application-dependant synchronization means between sender and receivers. In addition, the original source symbols k can ideally be recovered from any subset of the encoding symbols m of size equal to or only slightly larger than the number of source symbols, with error probability at most inversely polynomial in k. The ratio between the number of encoding symbols needed for successful decoding, m, and the number of source symbols, k, is the *code overhead*. The expected number of encoding operations sufficient to generate each output symbol is the *encoding cost*. When the code allows to decode the original k source symbols from any subset of k encoded symbols, it is *optimum* or *ideal*.

The Reed Solomon codes, introduced in the previous subsection, are the first example of fountain-like codes because a message of k symbols can be recovered from any subset of k encoding symbols. However, R-S codes require quadratic decoding time and are limited to a smaller block length n. Low-density parity-check (LDPC) codes [20] reduce the decoding complexity by use of the sum-product algorithm and iterative decoding techniques. They come closer to the fountain code ideal. However, early LDPC codes are restricted to fixed-degree regular graphs due to which significantly more than k encoding symbols are needed to successfully decode the transmitted signal. For the first practical implementation of fountain codes, we should wait for the Luby Transform (LT) codes [21]. Moreover, the most powerful and sophisticated fountain codes are Raptor codes [22], characterized by linear time encoding and decoding, thanks to the introduction of a pre-coding phase, and a small constant number of XOR operations per generated symbol.

1.3.3 Raptor Codes

Raptor codes, which stands for RAPid TORnado, are the first known class of fountain codes with *linear time encoding and decoding*. Invented by Amin Shokrollahi, Raptor codes represent a significant theoretical and practical improvement over LT codes. Similarly to all fountain codes, Raptor codes encode a given message of k symbols into a potentially limitless sequence of encoding symbols such that the knowledge of any k or more encoding symbols allows the message to be recovered with non-zero probability. A symbol can be any size, from a single byte to hundreds or thousands of bytes. The probability that the message can be recovered increases with the number of symbols received above k, becoming very close to 1 once the number of received symbols is only slightly larger than k. Raptor codes may be systematic or non-systematic.

A Raptor code is specified by parameters $(k; C; \Omega(x))$, where C is the (k, n) erasure correcting block code (*pre-code*), and $\Omega(x)$ is the generator polynomial of the degree distribution of the LT code. The definitions of *code overhead* and *decoding cost* for Raptor codes are compliant with the definitions given for the general fountain codes case. Indeed, *encoding cost* of Raptor code is defined as the sum of the pre-code encoding cost divided by the number of source symbols and the encoding cost of the LT code. Further, also memory requirements have to be taken into account when dealing with Raptor codes, since they require storage for intermediate symbols.

The basic idea of Raptor codes is to go over the limitations of preceding fountain coding schemes by concatenating some of them. The encoding phase is then expressed in three steps: 1) by means of an LDPC generation matrix (G_{LDPC}) a number of encoded symbols s are produced from k source symbols and organized into a vector (D_s); 2) the so produced s symbols and the source symbols k, are encoded generating h symbols; 3) the resulting h symbols, plus the s symbols of step 1) plus the k original symbols compose the intermediate symbol vector F. The F vector is encoded with an LT code providing a potentially unlimited sequence of encoded symbols. A schematic overview of these steps is reported in Fig. 1.2. The

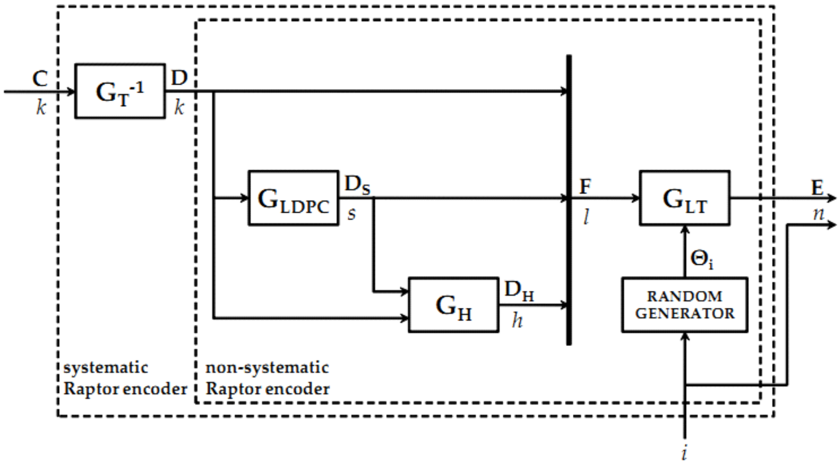

Figure 1.2: Raptor Encoder Structure.

inner part of Raptor encoding structure of Fig.1.2 gives an overview of a generic case in which the code is non-systematic. In case a systematic encoder is targeted (outer bounding box), source symbols are pre-processed by multiplying them by the matrix G_T^{-1}. The real core of Raptor encoding is the LT code, which takes in input $l = k + s + h$ intermediate symbols with their own identifier, called Encoding Symbol Identifier (ESI), and produces the encoding symbols E by xoring a different subset of source symbols of size d for each encoded symbol. That it like using a random generator for producing each encoding symbol which will have a certain degree d in the range [1,l] following a specific degree distribution. The chosen degree distribution will then strongly influence code performance. At the decoding side, receivers must know for each received encoded symbol which is its associate degree and the subset of source symbols used for its generation. This is generally achieved by using an equivalent pseudo-random generator. As already highlighted before, some information are provided to receivers in packet headers of by means of specific applications. This will unavoidably generate overhead.

A systematic Raptor code has been detailed in IETF RFC 5053 [23] and has been employed in multiple standards such as the 3GPP MBMS [24] - for broadcasting and streaming services - and the DVB-IPTV for commercial TV services delivery over IPs. Recently, the RaptorQ code having greater flexibility and improved reception overhead has been defined in the IETF RFC 6330 [25]. This code is able to recover with high probability a source block from any set of encoded symbols equal to the number of source symbols, and in rare cases from slightly more than that. RaptorQ codes provide superior flexibility, support for larger source block sizes, and better coding efficiency than Raptor codes in RFC 5053. They still belong to Fountain codes class. An exhaustive explanation of Raptor encoding and decoding can be found in [22]-[26].

1.4 Joint Source and Channel Coding

The classical Shannon information theory states that one can separately design the source and channel coders, to achieve error-free delivery of a compressed bit stream, as long as the source is represented by a rate below the channel capacity. Therefore, the source coder should compress a source as much as possible for a specified distortion, and then the channel coder can add redundancy through FEC to the compressed stream to enable the correction of transmission errors. This theory, from one side promises that the separate design of source and channel coding does not introduce any performance decrease, from the other side grants a complexity reduction of practical system design. The separation theory, however, is based on some assumptions (infinite delay, code length, etc.) which are no longer valid in practical systems. In fact, to make the compressed bitstream resilient to transmission errors/losses, redundancy must be added into the stream either by the source or the channel coder. Therefore, Joint Source and Channel Coding (JSCC) is often a more suitable scheme, since it allocates the total amount of redundancy between the source and channel coding in an optimized way. In video communication specific applications JSCC is generally in charge of three tasks:

- finding an optimal bit allocation between source coding and channel coding for given channel loss characteristics;
- designing the source coding to achieve the target source rate;
- designing the channel coding to achieve the required robustness.

A graphical illustration of the JSCC idea is provided in Fig. 1.3. In error-free channel condition, the straightforward solution for decreasing the information distortion is to increase data rate. This means that on a Rate-Distortion plan, the lowest distortion is achieved with the maximum available data rate (cf. point (R1,D1), Fig. 1.3). If the error-free condition is not valid anymore, this relationship is no longer true since the overall distortion is now function of both source and channel distortions. For a given channel rate, a compromise between data compression and data protection has to be reached. An optimal point exists for a given channel distortion. Clearly, different channel error rates result in different optimal allocations. Points (R2,D2) and (R3,D3) give an insight of this compromise.

Therefore, JSCC consists of finding the optimal source and channel coding allocation in order to achieve the lowest possible distortion for a given channel rate. There is a substantial number of research results in this area. A comprehensive review can be found in [27], while an interesting overview of specific applications of JSCC for video and image communications is provided in [7].

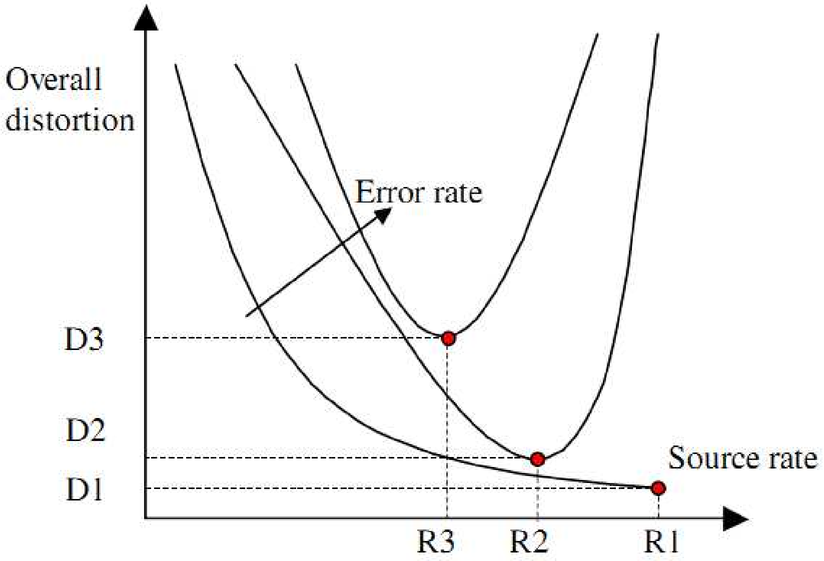

Figure 1.3: JSCC idea, Rate vs. Distortion illustrative example [7].

1.5 Error resilience and concealment techniques

As introduced before, channel coding error control techniques are just a sub-set of a bigger group of techniques whose aim is guaranteeing video flows a sufficient level of robustness against channel imperfections and other phenomena which can potentially deteriorate transmission performance, e.g. network congestion.
All the considered methods act on encoded video streams and try to detect and correct damaged and missing data. However, there are situations in which these methods are not sufficient for a full reconstruction at the decoder side, i.e. the received bitstream still contain errors (bit errors as well as packet losses), and additional protection mechanisms could be fundamental. To this extent, error resilience and concealment techniques can be employed.

Error Resilience (ER) techniques are generally applied at the source encoder. Their aim is to generate bit streams robust to transmission errors, so that an error/loss will not overly influence decoding operation and will not lead to unacceptable distortion. The design goal, in error-resilient coding, is to achieve the best decoded video quality for a given amount of redundancy, or minimize the redundancy while assuring a predefined level of quality, under an assumed channel environment. Compared to source coders that are optimized for coding efficiency, such coders typically are less efficient due to the introduction of additional redundancy bits, commonly referred as *overhead*, structured to enhance the video quality when the bitstream is corrupted by transmission errors. In other words, all the error-resilient encoding techniques work under the same assumption as JSCC techniques: the source coder works in a less efficient way, in order to ensure that the erroneous or missing bits in a compressed stream will not have a disastrous effect in the reconstructed video quality. Error resilient source encoders should be designed for minimizing the *error propagation effect*, which can degenerate in function of the prediction mechanisms adopted during the compression stage.

At the decoder end, some additional operations for recovering missing or disrupted data blocks can be performed. These operations are generally based on estimation techniques which exploit inherent correlation among spatially and temporally adjacent samples and are commonly referred as **error concealment**. The main difference between error concealment methods with respect to the other error-control techniques listed above, is that they bring the advantage of not employing additional data, however at the cost of an increased decoder computational complexity.

In addition, there are some other techniques which work in between the two classes, able to exploit embedded redundancy at the source coder, and to facilitate error concealment at the decoder. For this hybrid class of error control methods, the codec and the network transmission protocol must cooperate with each other. An exemplary case is represented by the assignment of different sets of Quality of Service (QoE) parameters function of the importance of data. For further information on the topic, the reader is kindly advised to [28].

1.6 Video Quality Evaluation

Another fundamental aspect of video communication systems is the quality they are able to provide to final users, or viewers. In order to avoid misunderstanding, it is useful to distinguish between the Quality of Service (QoS) and the Quality of Experience (QoE).

- The Quality of Service (QoS) is a well-established concept, mainly focused on network performance and data transmission. Quality of service was first defined by the ITU in 1994 and it encompasses requirements on connection-related aspects such as loss, time of response, SNR, etc. Lately, with the emergence of new telecommunication networks the term Quality of Service refers to the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance (i.e. to guarantee a certain bit error rate or packet losses probability, etc).

- The Quality of Experience (QoE) is an open research area and many standardization activities are still ongoing. The term QoE refers to the quality as experienced from viewers' perspective, with special attention to the effectively "perceived" quality, also addressed as user experience.

In the literature these two terms are often used in a confusing way, since sometimes the term Quality Of Service is used as a synonym of Quality of Experience. In the rest of this dissertation the two terms will be used following the definition given above. The interest of the scientific community and industries for the specification of well-defined, human-compliant, fast and reliable video quality evaluation procedure or metrics is considerable. The literature in the field is consistent. Providing a full overview of existing techniques is out of the scope of this dissertation. Notwithstanding that, the classification of video quality metrics in function of the subject who rates the quality and of the data needed to perform the evaluation will be supplied.

Using as a criterium the subject called to rate the quality of the video sequence, two main categories exist:

- **Subjective Video Quality Metric**
  The subjective quality assessment is the most reliable way for evaluating video quality. The name "subjective" is descriptive of its main characteristic: the quality is rated by human observers. Subjective tests are the reference for multimedia quality evaluation experiments, since they are the most accurate method. A number of subjects are asked to watch a set of video clips and to rate their quality within a predefined range of values. The most widely known subjective metric is the Mean Opinion Score (MOS) which is calculated by averaging rates over all viewers for a given clip. Albeit very accurate and coherent with the subjective experience, this class of metric is inconvenient for most applications since it is highly time consuming and has strict rules for the environment and the subjects employed during the test. Each human being has different interests and expectations while watching a video and this unavoidably affects test results, the way and the place in which subjective tests are performed attempt to limit these factors through well defined rules

and conditions. The ITU has formalized direct scaling methods in various recommendations [29]-[30], which suggest standard viewing conditions, criteria for the selection of observers and test material, assessment procedures, and data analysis methods. There are a wide variety of subjective testing methods, such as Just Noticeable Differences (JND) - suitable for small impairments, Double Stimulus Continuous Quality Scale (DSCQS) - implicit comparison, Double Stimulus Impairment Scale - explicit comparison - and many others. For further reference in the topic see [31].

- **Objective Video Quality Metrics**
  Objective quality assessment techniques are algorithms designed to characterize the video quality by means of numerical computations and predict viewers' mean opinion score. Objective quality metrics are of fundamental importance for standard organization since they provide means for evaluating the perceived quality without the need of time-consuming viewer panels. Objective metrics can be classified following different criteria:

  1. the amount of reference information needed for the computations:
  2. the domain in which the evaluation is performed: compressed or uncompressed.

When the chosen criteria is the amount of reference information, it is possible to distinguish three classes of metrics:

- **Full Reference (FR) Metrics**
  This class of metrics requires the entire reference video to be available since it is based on a frame-by-frame comparison between the test video and its reference version. FR metrics generally entail a pre-processing phase for reaching the spatio-temporal alignment of the two sequences. Due to their strict constraints they are not widely used in practical applications. The PSNR (Peak Signal to Noise Ratio) metric belongs to this class.

- **Reduced Reference (RR) Metrics**
  This class of metrics does not require the entire reference video but only some features of it. The same features are extracted from both the test and the reference video clips, and the quality evaluation is performed comparing them. As for the FR case, some alignment requirements, albeit not so stringent as for the FR case, need to be satisfied.

- **No Reference (NR) Metrics**
  This class of metrics does not require the reference video at all and it is completely free from alignment issues. As a result, NR metrics are much more flexible then FR and RR metrics at the cost of less accuracy. NR are based on assumptions about the content of the video sequence considered and they are able to distinguish content from distortions. Generally they are based on blockiness estimation.

Depending on the application domain, two classes can be drawn:

- **Metrics applied in the uncompressed domain**
  To this class belong *data metrics* which do not take into consideration the video content and *picture data* which analyze the video in terms of visual information. The former are distortion-agnostic and for this reason their accuracy is influenced by the type and properties of the distortion. Instead, the latter are content-agnostic, and for this reason their accuracy is influenced from the fact that viewer perception varies based on the part of the image or video where the distortion occurs.

- **Metrics applied in the compressed domain**
  To this class belong *packet-based* and *bitstream-based* metrics. These metrics go a step further to evaluate the effect of packet drops during transmission. In these cases, losses directly affect the encoded bitstream. That is why the considered class of metric is based on parameters deducible from the transport stream and the bitstream with no or little decoding. Advantages are clear, as disadvantages they have to be adapted to specific codecs and network protocols. Indeed, such metrics allow to measure the quality of many video streams/channels in parallel.

Due to the high variety of possibilities defined above, standards are a natural need. The first step in this direction has been taken from the Video Quality Experts Group (VQEG), established in 1997, which conducted the first formal evaluation of video quality metrics on common test material. Despite, the slow start of this activity, a good initial outcome was the creation of a valid public database of video clips with their associate subjective rating which was, and still is, largely used by the whole scientific community in the field. We must wait a few years for the completion of the second round whose outcomes have been considered the starting point of other two ITU recommendations [32]-[33] both targeting full reference metrics. Many other efforts have been carried out since then, evaluating metrics for multimedia scenarios, for low bit rate and small frame size applications, for No or Reduced Reference metrics. For further information in the field of standardization the interested reader is invited to refer to [34] and to the web sites of the International Telecommunication Union (ITU) [1] and of the Video Quality Expert Group [2]. While for further reference on the video quality metric topic, some interesting readings are [35]-[36].

1.6.1 Peak Signal to Noise Ratio

The Peak Signal to Noise Ratio (PSNR) is the most popular video quality metric and upon it several other metrics have been developed. For this reason, it deserves a detailed introduction. Despite its popularity, PSNR has only a relative pertinence with subjective experience, it is content-agnostic and it is based on a pure pixel-by-pixel comparison, without providing any attention to the visual content. Notwithstanding that, it is fast to compute and easily interpretable.

PSNR goes in parallel with the Mean Squared Error (MSE), on which it is based. PSNR is defined as the ratio of the squared useful signal peak over the mean squared error in decibel. More in detail, the PSNR between the i-th frame of the uncompressed original video sequence and the j-th frame of the reconstructed/reference (after the decoding process) video sequence is defined as:

$$\mathrm{PSNR}(i,j) = 10\log_{10}\frac{(2^P-1)^2}{\mathrm{MSE}(i,j)} \tag{1.1}$$

where 2^P-1 is the peak value that a pixel can take for a P-bit representation, while the MSE is computed as the average quadratic pixel by pixel difference between the original video frame, $f_i(x,y)$, and the decoded video frame, $g_j(x,y)$:

$$\mathrm{MSE}(i,j) = \frac{1}{M\cdot N}\sum_{x=1}^{M}\sum_{y=1}^{N}[f_i(x,y)-g_j(x,y)]^2 \tag{1.2}$$

where M and N represent the horizontal and vertical resolution respectively.

PSNR can be computed for each frame of the video signal under test, and it can be evaluated on both luminance and chrominance components, as well as on the R,G,B chroma components. The PSNR of the entire sequence is obtained by averaging the sum of frame-by-frame PSNR values over the total number of considered frames.

Using MSE and various modifications as a basis, a number of additional data metrics have been proposed and evaluated. Although some of these metrics can

[1] http://www.itu.int

[2] http://www.its.bldrdoc.gov

predict subjective ratings quite successfully for a given compression technique, distortion type or scene content, they are not reliable for evaluations across techniques.

1.7 Conclusions

In this chapter, an overview of the possible error-control mechanisms that can be used to combat transmission errors/losses in video communications systems has been provided. The focus has been mainly on channel coding techniques applied at upper layers of the protocol stack. The main principles of channel coding and an high level classification of channel codes along with the introduction of the most prominent ones have been given. The concept of joint source and channel coding has been clarified. Finally, the error concealment and resilience techniques have been briefly introduced as suitable tools for providing an additional level of robustness to video flows against transmission errors and losses. In addition, the problem of video quality assessment has been afforded and a classification of the most important classes of methods has been provided.

CHAPTER 2

APPLICATION LAYER FEC AND UNEQUAL TIME INTERLEAVER

The increasing demand of multimedia contents everywhere and in every moment is one of the biggest innovation drivers in the scientific community. In order to tackle these needs, several solutions have been proposed, some of which deal with enhancing the transmission speed, others with reducing the computational complexity of one of more functional blocks which process the information to be delivered and finally others with making the transmission as reliable as possible for guaranteeing good services also in difficult transmission scenarios. The research community in the field, as well as several commercial enterprizes and standardization bodies, are daily involved in considering new and challenging ways to solve or enhance one or more of the involved issues.
The research outcomes presented in this chapter deal with methods for enhancing the reliability of scalable video flows broadcasted to mobile devices, while granting fast access to the provided services and progressive quality refinement. The main problem to be considered when dealing with a transmission scenario like the broadcast to mobiles is the strong and unavoidable presence of long error bursts. The solution proposed here to cope with this type of error is named Layer Aware Forward Error Correction with Unequal Time Interleaver (LA-FEC UI) and is the result of the joint application of different techniques which can be proved to be efficient in comparison with existing technologies.

The conducted research and the results presented in this chapter have been obtained working in close collaborations with the designers (C. Hellge et al.) of the theoretical background at the basis of this research [8]. The concepts of Layer Aware Forward Error Correction (LA-FEC) and of Unequal Time interleaver (UI) have hence been inherited and used as a theoretical background for the implementation activity performed in order to test its performance in different transmission scenarios.

2.1 Motivation and Goals

Providing TV services to mobile terminals is a challenging topic which involves several aspects. Among these, one of the most prominent is the presence of long error bursts which can result in a significant loss of data. Such long error burst are mainly caused by shadowing from obstacles that affect wave propagation and can range from milliseconds to several seconds. The literature in the field is extensive and current video compression standards include mechanisms which try to avoid, or at least to limit, the effect of long error bursts. In addition, being the transmission over the Internet assimilable, in terms of practical behavior, to a transmission over a bursty channel, also inherent protocols concerning video transmission over IP can be considered mechanisms for preventing and/or recovering information loss ascribable to error bursts. Moreover, the recent - although extremely fast - drift toward a massive use of mobile devices for enjoying multimedia contents, as portable TVs, has driven the attention on scalable video flows and - as a consequence - on methods for the efficient and reliable delivery of such types of streams. Clearly, these solutions have to match the needs of current commercial systems and ongoing standardization activities andthey have to guarantee a target Quality of Experience (QoE) to end users.

One of the most common solution to overcome long error bursts is the use of long time interleaving for increasing the time diversity of the signal and thereby its robustness against error bursts. In the streaming context, long time interleaving has a known limitation: the increase in service tune-in time (zapping time). In fact, long time interleaver benefits come at the price of an increasing service tune-in time proportional to the interleaving length. This means that, the longer is the interleaver length employed, the longer will be the time that a final user has to wait for tuning-in into the service, s.a. to start the play-out of a brodcasted or streamed movie. Long time interleaving requires the receiver to wait until all packets in the interleaving period have been received and filled into the deinterleaving buffer. For this reason, current video transmission systems try to minimize the interleaving time length in order to provide a tune-in time typically below two seconds (see 3GPP MBMS requirements [37]), even though the service robustness would benefit from a longer interleaving length [38].

Therefore, a practical solution should have two main goals: protect transmission against long error bursts and minimize the service tune-in time.
Today's standards for Mobile TV and IPTV contain approaches that couple fast service tune-in with long time interleaving. Two different solutions are for example specified for multicast services in IPTV [39]. The usual receiver procedure to access an IPTV service is to join an IP multicast stream using the Internet Group Management Protocol (IGMP). Without a solution for fast tune-in, the receiver might need to wait up to several seconds before it can start to play-out the video stream until the play-out buffer is filled and a Random Access Point (RAP) has been received. With the so called server-based solution, a receiver can simultaneously establish a unicast connection via RTP to another server, which has cached several seconds of the multicast stream. This cached content can be transmitted in a much faster way than the normal streaming rate. The client can immediately play-out the cached content from the recent past while the play-out buffer is continuously filled with the multicast stream. Thereby, the server-based approach allows to provide services with fast tune-in and long time interleaving at the price of an increased end-to-end delay due to the required caching period. However, such an approach cannot be applied to a pure broadcast service since it requires a return channel to establish the RTP connection. The second solution relies on a companion stream. At start-up, a receiver joins two multicast streams of the same content but with different qualities. The stream with the lower quality has a higher RAP frequency and enables fast tune-in. After a transition time, which depends on the difference in RAP frequencies, the receiver can jump on the higher quality stream. The companion stream solution can be applied to broadcast services but does not allow to combine fast tune-in with long time interleaving and it wastes valuable bandwidth due to the transmission of the same content twice. Another solution is specified in DVB-SH by means of a Link-Layer FEC (LL-FEC) [40]. The LL-FEC scheme can be generated over large FEC source blocks that cover several seconds and thereby increases time diversity and robustness against long burst errors. Obviously, a large FEC source block increases the tune-in time to the service in the same way as long time interleaving, since the receiver has to wait until all FEC data of that source block has been received. In order to enable fast tune-in, two different options are proposed in

the DVB-SH standard. The first is to fast tune-in into the systematic part of the LL-FEC data. This alternative implies that users will experience an interruption of the service the first time an error is encountered, as terminals would need to buffer the remaining parity data in order to be able to decode. The second optional solution allows to transit to the parity data without service interruption by use of adaptive media play-out codecs. With this solution, the initial play-out is slowed down in such a way that the buffer needed for the FEC data can be filled over time. However, this solution requires modifications to existing video and audio decoders due to the strict timing constraints of the decoder buffers.

The solution herein proposed satisfies both the requirements listed above and it consists of the joint application of:

- the Layer Aware Forward Error Correction (LA-FEC) mechanism - presented by Hellge et al. in [8];
- layered video codec - specifically the SVC extension B.2.1 of the H.264/AVC standard has been employed;
- a clever unequal time interleaver scheme;
- an appropriate transmission scheduling.

As will be demonstrated by graphical and tabular results, the targeted solution enables broadcast services with a robustness comparable with that of traditional methods but with a much faster service tune-in time.

2.2 Theoretical Background

As highlighted above the solution herein presented is the result of the joint application of different techniques. For the sake of completeness, in this section a clear description of these ”sub-techniques” is provided.

2.2.1 Scalable Video Coding

In this activity, the Scalable Video Coding (SVC) extension [41] of the H.264/AVC standard introduced in section B.2.1 has been used. SVC generates layered bit-streams, hence a bit stream is made up of two or more sub-streams, enabling the extraction of different video representation, called media layers, from a single bit-stream. The layered bit stream will generally have a slight overhead in comparison

with a non-layered bitstream, which amount to around the 10%.
Respecting the terminology used in Sec. B.2.1, in the following the term Base Layer (BL) addresses the basic quality layer. The BL, when decoded, provides the lowest level of quality and it is a H.264/AVC compliant bit-stream that ensures backwards-compatibility with existing receivers.
While the term Enhancement Layer (EL) describes the layer - or layers - which incrementally refine the base layer quality. It is important to underline that the enhancement layer(s) can improve the video quality in one of three possible scalability dimensions, which for the SVC standard comprise spatial, temporal and SNR/Quality scalability.

In the rest of this dissertation the term *single layer* (SL) will be used to refer to a video flow encoded in the general mono-layer way, like for example a H.264/AVC stream. Single layer media streams allow decoding of a single and predefined bit-rate and media quality. While with the term *layered media stream* we refer to a video flow encoded by SVC. It is worth noticing that the solution proposed in this chapter can also use other scalable video codecs and not exclusively SVC.
In Fig. 2.1 the difference between a single layer media stream and a layered media stream is illustrated. A layered media stream consists of multiple sub-streams that allow to extract multiple bit-rates and media quality levels from a single bit-stream. Layered bit-streams typically contain a hierarchy between the layers which results from the encoding algorithm. In SVC, the BL is more important than the EL due to inter-layer prediction. Therefore, in case of missing BL information, the EL information becomes useless due to missing prediction information. This concept is better clarified in Fig. 2.1.

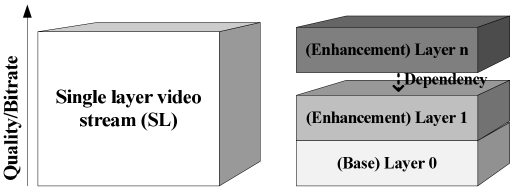

Figure 2.1: Single layer (SL) media stream and layered media stream [1].

2.2.2 Layer Aware Forward Error Correction

The Layer Aware Forward Error Correction (LA-FEC) is a powerful techniques presented by Hellge et al. in [8]. The LA-FEC is based on the idea of extending the algorithm for generating FEC data across dependent media layers. As already mentioned, the video encoding procedure, as well as the intrinsic nature of video contents, implies a strong correlation of data within a video flow. This intrinsic correlation persists and is further stressed in layered video flows. Results clear how important it would be, for layered video, exploiting these correlations for improving the level of protection without varying the overhead introduced. In the LA-FEC approach, as presented in [8], a scalable video flow of n media layers is considered. The BL, or layer 0 is protected with a traditional FEC algorithm. This allows the BL to be independently decoded by existing receivers. In the encoding procedure of the EL(s), base layer data are used as well. This means that FEC data of the first EL are generated over both base and enhancement layer source data, while FEC data of the n-th EL are generated over layers 0-$(n-1)$. In this way FEC data of higher layers will additionally and progressively protect lower layers. This enhanced level of protection is very important considering that the lower layers are also the most important layers, and without them higher layers are completely useless. In order to understand the mechanism of the LA-FEC, propaedeutic for the LA-FEC Unequal time Interleaver (LA-FEC UI) approach herein addressed, Fig. 2.2 intuitively shows which is the dependency structure used for generating redundancy. Base layer redundancy is generated by a traditional FEC approach, while FEC data of each enhancement layer are generated over source data of the EL itself and all the layers on the bottom of it.

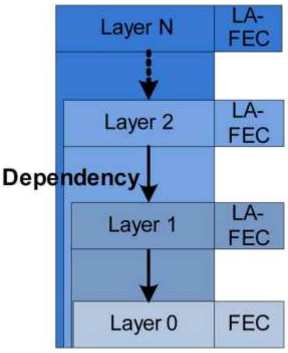

Figure 2.2: LA-FEC data generation [8].

In figures 2.3-2.4 the comparison between encoding and decoding procedures of LA-FEC and traditional approaches, denoted by Standard Forward Error Correction (ST-FEC), is illustratively given. In the figures, base and enhancement layer are indicated as layer 0 and layer 1 respectively. As FEC algorithm XOR combinations of source symbols are used. In Fig. 2.3, the ST-FEC approach is described on the left side and the LA-FEC on the right one. In the ST-FEC case, parity data are generated independently for each media layer and FEC data of a certain media layer can be used only for recovering source data of the layer itself. In the LA-FEC case, BL FEC data are generated over BL source data, while EL FEC data are generated over the source bits of both layers. Encoded data are organized into codeword and transmitted over an error prone channel, s.a. a Packet Erasure Channel (PEC) where erroneous symbols are considered lost and discarded. In Fig. 2.4 the decoder side is

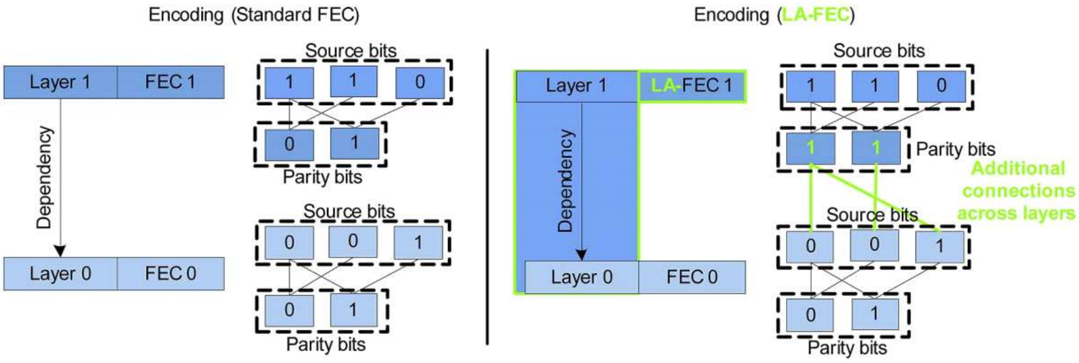

Figure 2.3: ST-FEC vs LA-FEC: encoding procedure [8].

illustrated: ST-FEC decoding on the left and LA-FEC decoding on the right. The transmission over a lossy channel has determined some data to be corrupted or lost. Three transmission errors have occurred on the BL codeword, while the EL codeword is error free. The ST-FEC decoding approach is not able to recover for losses because parity data are not enough. This makes also enhancement layer data useless. If LA-FEC is used, the BL can be successfully decoded, because EL FEC data can now be used for decoding also the BL. Clearly, the enhancement layer(s) will suffer for a lower protection with respect to ST-FEC. Neverthless, being the enhancement layer useless without lower layers this aspect can be considered negligible.

Stemming from the assumption that the FEC code used for both ST and LA-FEC is ideal, the condition for a successful decoding of a media flow (or layer in this case) requires the reception of r out of n symbols like shown in equation 2.1 with k_x and r_x being respectively the number of source and received symbols of layer x:

$$r_x \geq k_x \tag{2.1}$$

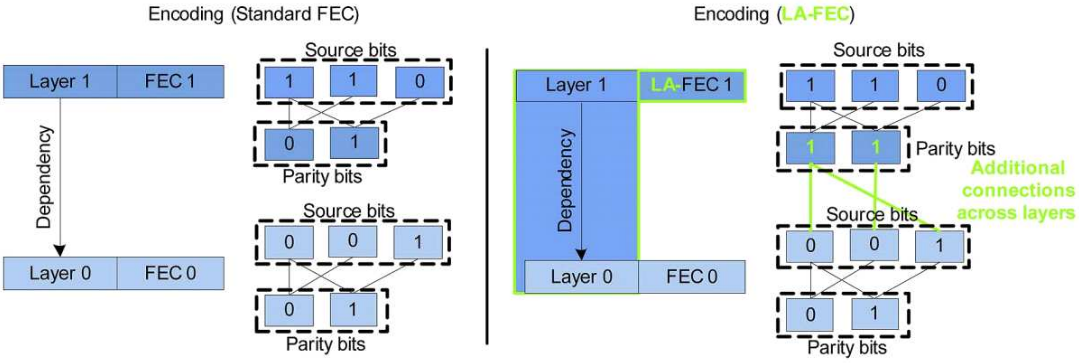

Figure 2.4: ST-FEC vs LA-FEC: decoding procedure [8].

Hence if ST-FEC is applied, the base layer decoding condition is:

$$r_{BL} \geq k_{BL} \tag{2.2}$$

Due to media layer dependencies, the EL can only be decoded with a successfully received BL, the condition for successful enhancement layer decoding becomes:

$$(r_{BL} \geq k_{BL}) \wedge (r_{EL} \geq k_{EL}) \tag{2.3}$$

When LA-FEC is applied, the conditions for base and enhancement layer successful decoding become respectively:

$$((r_{BL} \geq k_{BL}) \vee (r_{BL} + r_{EL} \geq k_{BL} + k_{EL})) \tag{2.4}$$

$$((r_{EL} \geq k_{EL}) \wedge (r_{BL} + r_{EL} \geq k_{BL} + k_{EL})) \tag{2.5}$$

where r_{BL} and r_{EL} represent the received symbols and k_{BL}, k_{EL} the source symbols of each media layer while the symbols $\vee$ and $\wedge$ are the OR and AND logic operations.

2.2.3 Time Interleaver

The time interleaver is an efficient way to overcome errors and losses which affect the transmission. The basic concept of interleaving is simple and notably a variety of different applications exist. The time interleaver spreads information data over a longer period of time increasing the time diversity of a message. For example, during one transmission slot whose lenght is expressed in seconds, a total of $n = k + p$ encoded symbols are allocated. If during this transmission slot, a certain number of losses occur, the number of correctly received symbols over the n

originally transmitted may become too low for recovering the original data. When time interleaver is applied, the n encoded symbols are not allocated in one time slot anymore, but they are spread over more time slots. This procedure in then repeated for the whole sequence in a structured way. The new error pattern to which the n symbols, originally belonging to one time slot only, are now subject is hopefully maintained under the threshold of admissible losses. An example of what described above is given in Fig. 2.5.

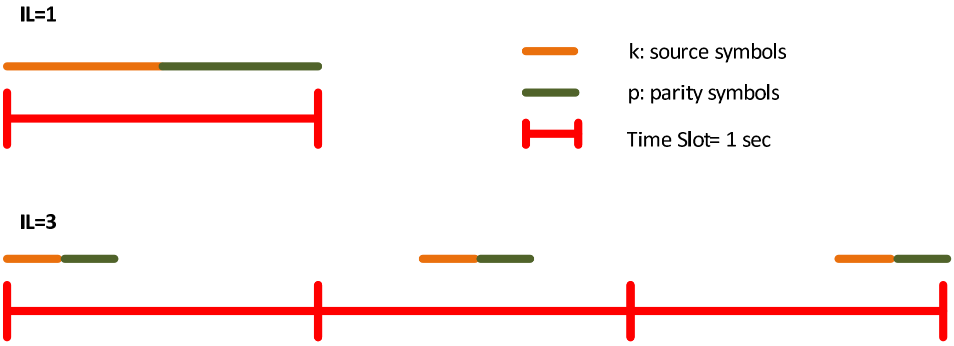

Figure 2.5: Concept of time interleaving with interleaving length IL=1 and IL=3.

It is important to note that the transmission scheduling of source and parity data reported in Fig. 2.5 is just one of the possible examples and many other alternatives exist. Applying the time interleaver, the required minimum time to recover all the n symbols (originally allocated within one time slot) is referred to as IL_{delay} and is easily calculated by multiplying the number of transmission slots over which data have been spreaded minus one, for the time slot duration expressed in seconds. In the case illustrated in Fig. 2.5, the IL_{delay} is then equal to the duration of two time slots. In other words, the interleaving delay is the additional time that the decoder has to wait in order to start the decoding procedure for a certain data block when it has been interleaved.

2.3 Layer Aware FEC with Unequal Time Interleaver

The Layer Aware Forward Error Correction with Unequal Time Interleaver (LA-FEC UI) technique merges together scalable video coding, LA-FEC and an ad hoc interleaver structure enabling a new way of service provisioning that jointly achieves fast zapping and enhanced protection against error bursts thanks to the application of a long time interleaver. Commonly, the term ”zapping” refers to the use of a device to switch a television channel/service. In this work, the word zapping is used in a similar way and it addresses the time that a final user needs to start to enjoy a video service after tuning-in it. The LA-FEC UI enables fast zapping by granting a fast tune-in with basic quality and a quality refinement time function of the length of the interlever applied to protect the service and of the instantaneous channel conditions. More in detail, the fast zapping is achieved applying a short time interlever to the base layer at the cost of lower robustness against burst losses. Long time interleaving is provided by the SVC enhancement layer with stronger robustness against burst losses but increased FEC latency. However, due to the enhancement layer being FEC coded with LA-FEC across the base layer, the base layer also benefits from the improved time diversity of the enhancement layer, resulting in a significant robustness improvement compared to traditional techniques.

The presented scheme will be explained in detail for an exemplary two-layer SVC video flow, but it is generally applicable to any kind of layered media and time synchronized data and it can be easily extended to an higher number of layers.

The LA-FEC UI approach can be described in seven steps:

- Step #1: Video compression. A video is encoded in a two-layer stream: a base layer (BL) - or layer 0 - and one enhancement layer (EL) - or layer 1. The two layers consist of k_{BL} and k_{EL} source symbols respectively.

- Step #2: FEC generation. The layered video flow resulting from step #1 is encoded following the LA-FEC paradigm. As FEC code, a Raptor code is used. Moreover, other FEC code can be applied as well. The two layers can be protected equally or unequally. The Raptor encoder is opportunely modified as outlined in [8] to fit the LA-FEC encoding structure. A similar procedure is applied to the Raptor decoder. Parity data, p_{BL} and p_{EL} are generated and along with source symbols are reorganized into FEC Data Blocks (FEC DBs). Since LA-FEC is used, p_{BL} symbols are generated over k_{BL} symbols only, while p_{EL} symbols are generated across base and enhancement layer source symbols.

- Step #3: Unequal Time Interleaver. FEC DBs coming from step #2 are now interleaved. In order to grant a fast tune-in time, no or short interleaver is applied to BL data. In order to enhance the robustness against error burts, enhancement layer data are typically interleaved over a longer period of time. The term Interleaver Length (IL) addresses the number of FEC DBs over which

FEC data are interleaved. In the following, IL_{BL} and IL_{EL} will indicate the interleaver lengths applied to base and enhancement layer respectively.

- Step #4: Transmission Scheduling. Base and enhancement layer data must be rearranged into a single layer media stream accordingly to the interleaver factors applied in step #3.
- Step #5: Media transmission. The video flow compressed, encoded, interleaved and opportunely synchronized is transmitted over an error prone channel characterized by a certain error probability.
- Step #6: Sequence Deinterleaving. The received video sequence may have some missing symbols/packets due to transmission errors. Data are deinterleaved as it is, received data are reorganized in the original FEC DB order.
- Step #7: Decoding. Media decoding is performed previa verification of decodability (cf. equ. 2.4-2.5).

To allow the reader to clearly understand the proposed solution a toy example is provided in Fig. 2.6. The outlined example illustrates steps from 3 to 7 of the LA-FEC UI procedure.

In the example, a two layers SVC stream with a $1 : 2$ bit rate ratio between BL and EL is considered, this means that the EL source symbols are twice the BL source symbols. A FEC code rate of 0.5 is applied. Source and parity data of both layers are rearranged into FEC data blocks (FEC DB) ideally matching one second of media data each. In the particular case of Fig. 2.6, the base layer consists of $k = 1$ source symbol and $p = 1$ parity symbol, while the enhancement layer consists of $k = 2$ source symbols and $p = 2$ parity symbols. Each FEC DB is represented with a different color. The full color blocks represent BL data, while the striped ones represent EL data. BL data are not interleaved (IL_{BL}=1), while an interleaver length of 4 is applied to the EL (IL_{EL}=4). The upper part of the figure shows the data stream at the transmitter side. The first row describes the original transmission order used for the media stream: FEC DBs are transmitted in sequential order and source symbols of each layers are transmitted before respective parity symbols. Moreover, data belonging to different media layers are combined. The described situation is a snapshot of the sequence to be given as input to the unequal time interleaver. The second row (at the transmitter side) shows the transmission order resulting from the LA-FEC UI procedure. In between them, the interleaver process is outlined for the two last FEC DBs. For achieving the long time interleaving, the EL symbols are fed into a convolutional interleaver. The symbols of a FEC data block are written column-wise into the interleaver memory. The data is read from the diagonal of the memory matrix, thereby the symbols of different FEC data blocks are interleaved. The interleaved symbols of the EL replace the equivalent number of EL symbols in the original transmission order. After the writing process, the memory buffer is shifted column-wise towards the right side, hence the already processed data are dropped. Such an interleaving over multiple FEC data blocks requires buffering on the server side. This increases the end-to-end delay of the system by a factor IL_{delay} given by the difference of the interleaver lengths $(IL_{delay}{=}IL_{EL}\text{-}IL_{BL})$. Since in the example $IL_{BL} = 1$, base layer data are not delayed and they are transmitted as they are. EL data are reorganized over IL_{EL} consecutive FEC DBs.

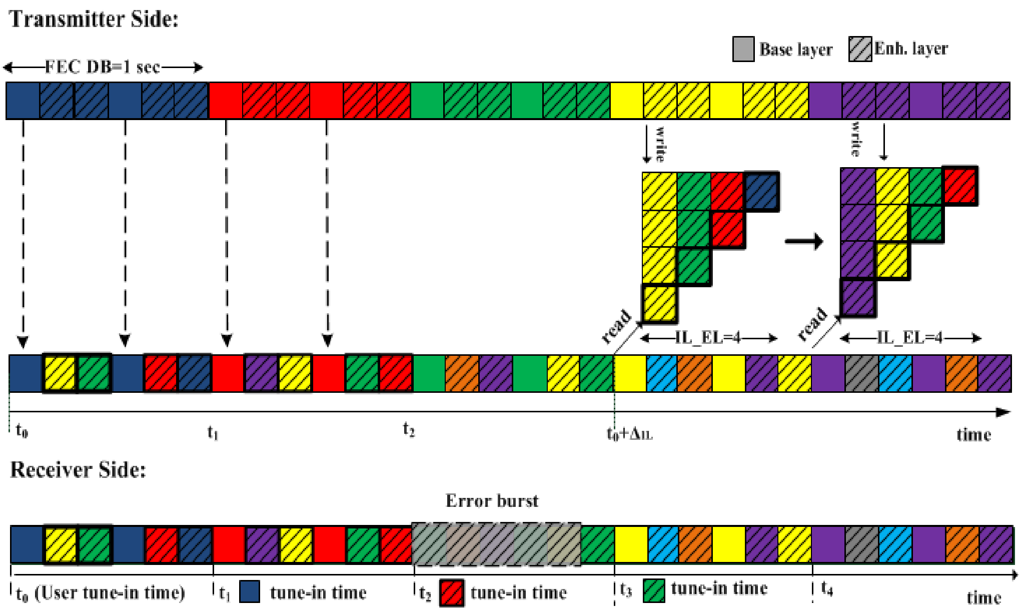

Figure 2.6: Unequal time interleaving of an exemplary SVC media stream. Situation at transmitter and receiver side [2].

The bottom part of Fig. 2.6 illustrates the situation at the receiver side after the transmission over a lossy prone channel. The received stream is affected by a

burst erasure which lasts for five symbols.
As can be observed on the time axis in the picture, the user tunes in the service at time instance t_0 and at time instance t_1 the first FEC DB can be decoded in base layer quality. Accordingly with the decoding conditions pointed out in section 2.2.2, at time instance t_2 also the second FEC DBs can be decoded, already in EL quality. Starting form t_2 a burst error affects the sequence but, thanks to the applied LA-FEC approach and to the interleaved order of the sequence, enough enhancement layer symbols belonging to the third FEC DB have already been received, allowing to start successfully the decoding procedure in EL quality. As illustrated, the proposed methodology allows the recovery of both media layers and it enables the play-out to start in base layer quality after one second only. In the error-free case, only one additional second is needed to switch to full quality. Considering the same settings, but applying a standard approach for FEC generation, the experienced quality will substantially differ from the LA-FEC case. For example, pointing always at the situation depicted in Fig. 2.6, the third FEC DB (the green one) cannot be decoded at all and a service interruption is experienced. It is worth noticing that, the tune-in time to full quality is a function of the instantaneous channel conditions. In the error free case, the full quality tune-in time is inversely proportional to the applied code rate (CR); for example using a CR=0.5, the full quality tune-in time is only 2 seconds. Neverthless, the full error correction capability is reached at time instant t_4, due to the longer interleaver applied to the enhancement layer. In case of an early tune-in, in the error free case, the service robustness increases automatically over time.

If the single layer case is considered, and the same interleaver factor applied to SVC enhancemet layer is used, a user which tunes-in the single layer service at time instance t_0, has to wait until t_4 in order to be able to start the play-out.

All the above is more easily described in Fig. 2.7, where the quality experienced by final users in the specific situation of Fig. 2.6 is illustrated in comparison with a state-of-the-art single layer transmission and with the ST-FEC approach. The unequal time interleaver procedure is applied to both layered cases (ST and LA-FEC). The single layer stream is provisioned with the same interleaving length like used in the SVC EL.

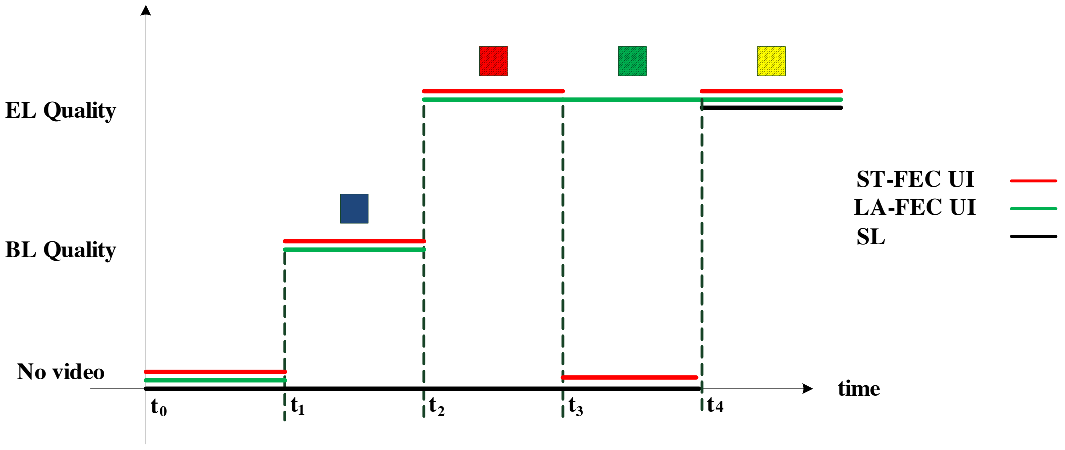

Figure 2.7: Experienced quality of users for the SL case (black line, IL=4), ST-FEC UI (red line) and LA-FEC UI (green line) both with $IL_{BL} = 1$, $IL_{EL} = 4$.

2.4 Use cases

In this section, simulation results are provided for two exemplary scenarios. The first scenario concerns the application of LA-FEC UI as Upper Layer FEC in the context of video broadcasting to mobile applications. The second, proves the efficiency of the LA-FEC UI as Application Layer FEC in a satellite to mobile context.

It is important to note that my personal contribution has been primarily focused on the implementation of the unequal time interleaver for both simulation scenarios herein considered. More in detail, the interleaver implementation has been performed by jointly providing means for evaluating the optimal interleaver length to be applied in function of the bit rate distribution between BL and EL and the compromise between interleaver length and FEC code rate. Therefore, the LA-FEC UI implementation is also able to support equal and unequal error protection of scalable media layers. Please note that, in the following, the expression *equal error protection* (EEP) refers to the cases in which the same code rate is applied to each media layer, while the expression *unequal error protection* (UEP) refers to the schemes where the total amount of redundancy is unequally distribuited among media layers, this unequal allocation can be optimized for base or enhancement layer protection. Hence, all considered, the achieved implementation allows to perform a system optimization in function of one or more of the following parameters:

1. bit rate distribution among layers,

2. code rate distribution among layers,

3. interleaver lengths.

Different evaluation parameters will be employed for validating the benefits of the proposed solution. For all considered simulation setups, the performance of the LA-FEC UI is compared to that of traditional FEC scheme (ST-FEC UI) and with that of single layer media streams, with and without the application of the unequal time interleaver methodology herein proposed. When the interleaver is applied, the single layer media flow has to be interleaved with the same factor employed to the enhancement layer of the SVC schemes. In the cosidered settings, no interleaver is applied to the base layer and a longer interleaver is used for the enhancement layer. In the single layer case, performance of schemes with $IL > 1$ are reported as upper bounds since they do not achieve the assumed service tune-in time requirement. This consideration is fundamental in order to correctly interprete simulation results.

The main objective of this study has been to provide services with a sufficient robustness against error bursts while assuring final users a fast access to the choosen service (tune-in time). The constraint to be satisfied is a service tune-in time of 1 second, as specified in the MBMS requirements [37].

2.4.1 Broadcasting to Mobile Scenario

In this section the performance of the LA-FEC UI technique when applied in the broadcasting to mobile scenario are presented. Simulations are based on real encoded video streams encapsulated within RTP/UDP/IP protocols. As FEC scheme, the layer-aware Raptor code presented in [8] is used. As a transmission channel, a Gilbert-Elliot channel model with parameters taken from simulation conditions defined in MPEG [42] is employed. Performance are shown in terms of FEC block error rate (FEC BER) and in terms of resulting video quality (PSNR), consideryng varying channel conditions (p_{er}).

2.4.1.1 Media streams characteristics

The same video clip has been encoded into two dinstinct flows: one single layer (SL) bitstream and one two-layer (SVC) bitstream. In the former case, the H.264/AVC standard has been employed, therefore the resulting flow is backward compatible with existing receivers. When decoded, it provides an indivisible stream with a fixed play-out quality. In case of layered encoding, the SVC standard extension has been employed. Two different play-out qualities can be decoded. For the encoding procedure, the JSVM software [42] has been used. The SL stream has been encoded at QVGA and 30fps using the High Profile at 570 kbps with a GOP size of eight frames and a RAP frequency of one per second. The SVC stream is encoded with coarse grain scalability (CGS) at the same resolution and frame-rate using the Scalable High Profile of H.264/AVC. The overall SVC stream shows a coding penalty compared to the SL stream of around the 10%, which results in an overall bit-rate of 627 kbps. The largest share with 70% of the overall bit-rate is allocated to the EL, while the remaining 30% is allocated in the BL stream. The average video quality of the SL and SVC stream is 34.5 dB in terms of PSNR and the SVC BL quality is 31.0 dB. The network abstraction units (NALUs) of the video stream are encapsulated into the RTP protocol with a target maximum transmission unit (MTU) size of 1500 byte. The RTP packets are encapsulated into the UDP and IP protocol and the resulting IP stream is forwarded to the FEC Framework. The FEC generation is done in a way that each FEC data block (FEC DB) covers one second of media data and it contains one RAP. The FEC data block generation follows the IETF FecFramework specified in RFC 6363. In order to enable a fair comparison between single and layered media streams performance, the total amount of service bit-rate should be the same for both media flow types (single and multi-layer). Taking into consideration the intrinsic overhead introduced by SVC encoding, the single layer flow will have more parity data than the layered ones. Therefore for aligning the two media types in terms of overall service bit-rate, a strictly higher code rate will be used for the SVC-FEC schemes. The code rate employed for single and layered stream protection will be indicated with CR_{SL} and CR_{SVC} respectively. Note that, if only one CR_{SVC} value is specified, it will be applied to both media layers resulting in equal error protection (EEP) schemes according to the definition given above. Otherwise, two different values will be specified, one for each media layer. In this case, unequal error protection (UEP) is used.

2.4.1.2 Simulation Set and results

The FEC data block length and thereby the introduced delay of the FEC is set to 1 second. This value leaves enough room for other system components to comply with the maximum channel switching time of two seconds as given by the 3GPP MBMS requirements.

In terms of FEC protection, two different levels have been tested: CR_{SL}=0.5 and CR_{SL}=0.7. In terms of interleaving length, two exemplary cases have been considered:

- no interleaver,
- no interleaver for the base BL and IL=5 for both SVC EL and SL.

As highlighted above, the SL case with IL=5 is reported as a reference since it does not fullfill the assumed tune-in time condition.

For simulation of a mobile channel, a Gilbert-Elliot model whose parameters are derived from the evaluation criteria of AL-FEC in MPEG [42] is used. The model assumes a fixed Average Burst Error Length ($ABEL$) and varying average loss probabilities, which are described by the erasure probability p_{er}. For each erasure probability value, 10000 repetitions with different random seeds have been conducted. For the simulations, $ABEL$=1 second is assumed. Performance is expressed in terms of FEC block error rate and of overall PSNR for incresing erasure probability values. For a given media layer, the FEC block error rate is defined as the ratio between the number of undecodable FEC DBs and the total number of FEC DBs within the media layer. In order to have an insigth into the quality experienced by final users, PSNR values are calculated on a sequence basis.

Figures 2.8-2.9 and 2.10-2.11 are related to the equal error protection case. More in detail, results in figures 2.8-2.9 report the case of $CR_{SL} = 0.5$ and $CR_{SVC} = 0.55$; while results in figures 2.10-2.11 cover the case of $CR_{SL} = 0.7$ and $CR_{SVC} = 0.78$.

Looking at Fig. 2.8, it can be easily deducted that the LA-FEC UI approach outperforms both the ST-FEC UI and the SL IL1 approaches. When no interleaver is applied (IL1), the LA-FEC base outperforms the SL and ST-FEC approaches, similarly to what is shown in [8]. While LA-FEC enh performance is weaker with respect to the single layer case. The same trend can be observed when a longer interleaver is applied. When the unequal time interleaver is applied (IL1-IL5), thanks to the intrinsic nature of the LA-FEC UI approach, both layers will benefit from the longer interleaver applied to the enhacemet layer. Contrarily to that, if ST-FEC is used, only the enhancement layer will benefit from the longer time interleaving, because each layer is protected independently. Therefore, the overall video quality does not improve due to the media dependencies between BL and EL.

The best performing setting is the LA-FEC base IL1-IL5, which shows performance comparable with that of the SL IL5 case, where the latter has the drawback of

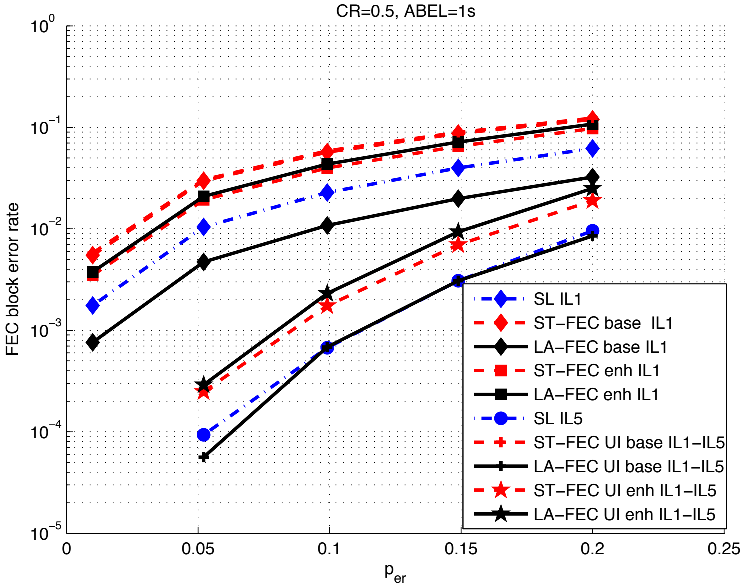

Figure 2.8: FEC block error rate of selected settings comparing LA-FEC UI, ST-FEC, and SL in case of Equal Error Protection[1].

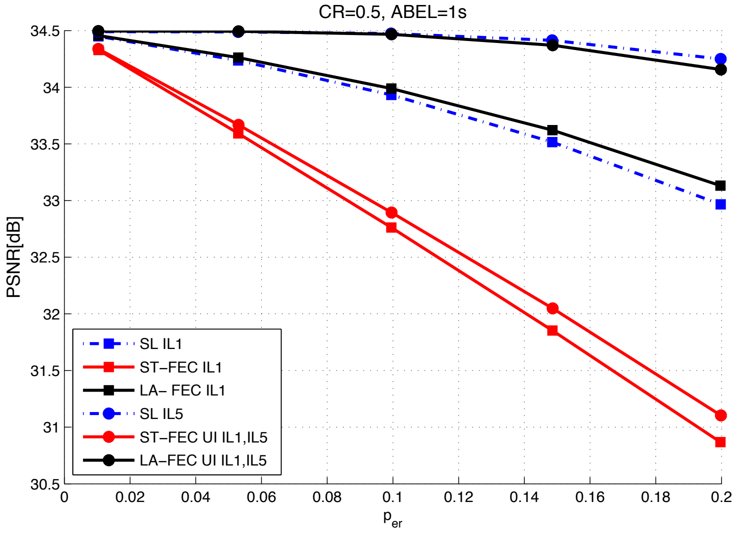

Figure 2.9: Video quality in terms of PSNR of selected settings comparing LA-FEC UI, ST-FEC, and SL in case of Equal Error Protection[1].

a longer tune-in time (5 seconds). At low erasure probabilities ($p_{er} \in [0.01 - 0.05]$), SL IL5 and LA-FEC base IL1-IL5 are always successfully decoded. At higher erasure probabilities, the two settings present comparable performance, whereas the LA-FEC enh IL1-IL5 performs slightly worse compared to the BL and SL cases.

In Fig. 2.8 the overall video quality for the different settings is shown. As can be observed, the performance of the LA-FEC UI scheme and of the SL IL5 scheme are very similar and they both largely outperform the ST-FEC UI case. In order to perform a fair comparison among settings, the SVC UI cases have to be compared with SL IL1, otherwise the tune-in time requirement is not satisfied. In this case, the gain introduced by the LA-FEC UI scheme is significant: at a PSNR quality of 34.1 dB, the LA-FEC UI IL1-IL5 allows to overcome a 0.13 higher channel erasure rate p_{er} with respect to the SL IL1 case at the same PSNR quality.

Considering Fig.2.10, which refers to the application of a CR_{SL} of 0.7, it is easy to see that LA-FEC UI outperforms SL IL5 for erasure probabilities higher than 0.12, while the two have comparable performance for error rates in between 0.05 and 0.12. For lower p_{er} values, SL IL5 outperforms all other settings in terms of FEC Block Error Rate (FEC BER). The LA-FEC UI base outperforms also the ST-FEC base UI and the SL IL1, both of which fulfill the tune-in time requirement. For the enhancement layer, the same consideration given for the $CR_{SL} = 0.5$ case is valid. Complessively, the LA-FEC UI IL1-IL5 is the best performing setting. In terms of overall PSNR quality (cf. Fig. 2.11) the LA-FEC UI IL1-IL5 largely outperforms the equivalent ST-FEC case and has performance slightly weaker that the SL IL5 reference case. As already introduced above, a fair comparison has to be done in between the LA-FEC UI IL1-IL5 and the SL IL1 case. The gain introduced by the LA-FEC UI approach is significant. At a PSNR quality of 34.1 dB, LA-FEC UI allows to overcome 0.08 channel erasure rate with respect to the SL IL1 case of the same PSNR quality. The lower gain, with respect to the $CR_{SL} = 0.5$ is due to the weaker protection used.

Figures 2.12-2.13 show the same performance comparison, as given for the above settings, in case of unequal error protection. For the UEP case, several possible code rate distributions among media layers have been tested, some of which aimed at granting stronger protection to the base layer while others to the enhancement one. In order to evaluate the potential benefits of the unequal error protection approach with respect to the equal error protection case, the two settings must be comparable in terms of overall service bit rate of the FEC-encoded media streams.

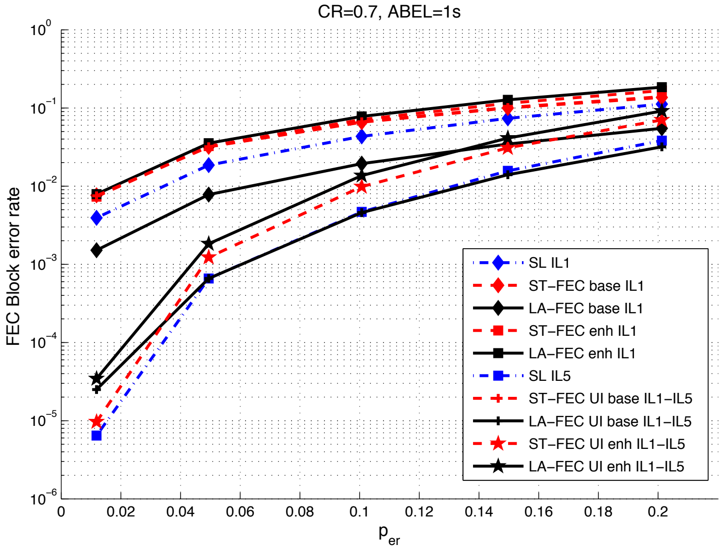

Figure 2.10: FEC block error rate of selected settings comparing LA-FEC UI, ST-FEC, and SL in case of Equal Error Protection.

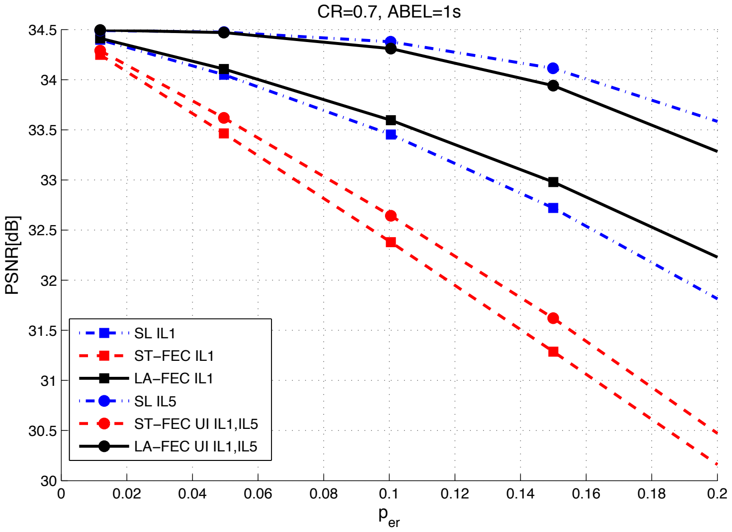

Figure 2.11: Video quality in terms of PSNR of selected settings comparing LA-FEC UI, ST-FEC, and SL in case of Equal Error Protection.

To this extent, when a code rate of 0.5 is used for protecting the single layer flow $CR_{SL} = 0.5$, the SVC video flow is protected with a $CR_{SVC} = 0.45/0.6$. In figures 2.14-2.15, $CR_{SL} = 0.7$ and $CR_{SVC} = 0.65/0.82$ have been simulated. The code rate distribution among layers is optimized for enhanced base layer protection, while the bit rate distribution is unchanged with respect to the equal error protection case. This means that for both UEP settings, the base layer is protected more than the enhancement layer. Presumably, this will result in an increased robustness of SVC streams against error bursts in comparison to EEP. Clearly, both ST and LA-FEC schemes will benefits from this enhanced protection but LA-FEC approaches still outperforms standard cases, highlithging the efficiency of the proposed solution.

Pointing at fig. 2.12, it is easy to note that the LA-FEC UI base IL1-IL5 and the SL IL5 case present comparable performance. In addition, the LA-FEC UI base IL1-IL5 outperforms both ST-FEC base UI and SL IL1. The associated enhancement layer performance is comparable with that of the ST-FEC counterpart, which - as for the preceeding case - greatly benefits for the longer time interleaver. Due to the stronger base layer protection, the gap between the LA-FEC UI enhancement and the ST-FEC UI enhancement is smaller with respect to equal error protection cases. In terms of FEC Block Error Rate performance, the base layer of the LA-FEC UEP case has better performance than ST-FEC UEP and SL IL1. In terms of overall quality (cf. Fig.2.13), the LA-FEC UI setting outperforms SL IL1: at a PSNR of 34.1 dB, it is able to overcome 0.12 higher erasure rate than the SL IL1 at the same quality.

Considering now the case of unequal error protection and $CR_{SL} = 0.7$ (cf. 2.14-2.15) the observations done for the other cases are valid. The LA-FEC UI IL1-IL5 base outperforms both equivalent cases (SL IL1, ST-FEC UI IL1-IL5) and has performance comparable to SL IL5 for erasure rate higher than 0.05. In terms of overall quality, LA-FEC UI allows to overcome 0.08 higher erasure rate then the SL IL1 at the same quality.

For the sake of completeness, figures 2.16-2.17 shows the performance of equal and unequal protection on the same graph for the IL5 case only.

2.4.2 Satellite to Mobile Scenario

In this section, the LA-FEC UI method is employed as application layer FEC scheme in the context of mobile satellite video communication applications.

Satellite distribution provides a cost-efficient way to expand the coverage of mo-

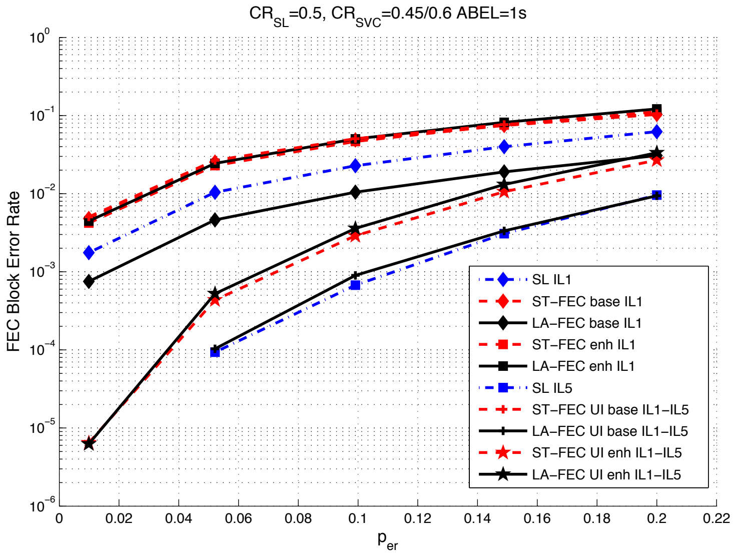

Figure 2.12: FEC block error rate of selected settings comparing LA-FEC UI, ST-FEC, and SL in case of Unequal Error Protection.

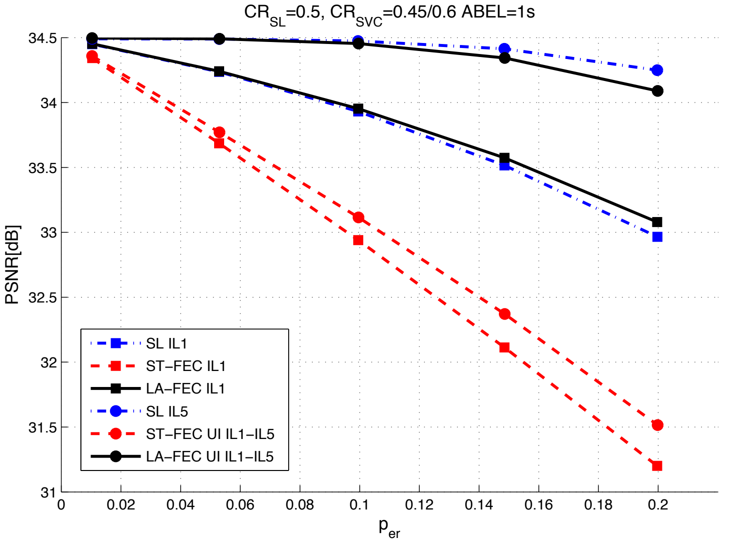

Figure 2.13: Video quality in terms of PSNR of selected settings comparing LA-FEC UI, ST-FEC, and SL in case of Unequal Error Protection.

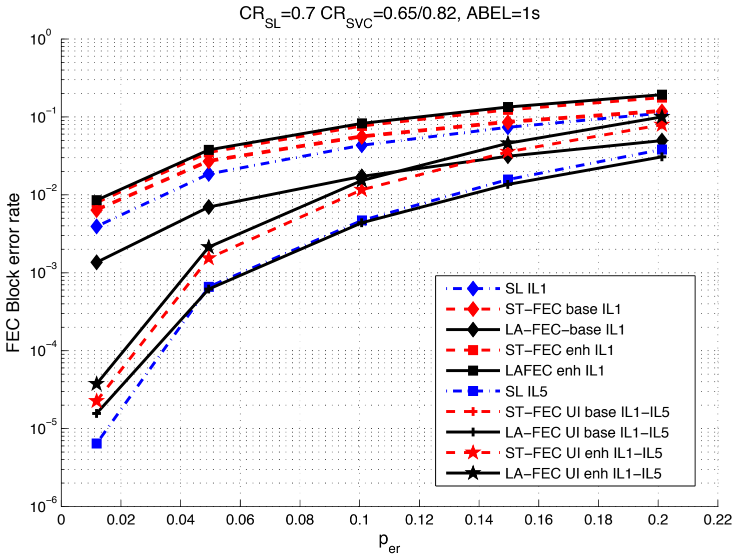

Figure 2.14: FEC block error rate of selected settings comparing LA-FEC UI, ST-FEC, and SL in case of Unequal Error Protection.

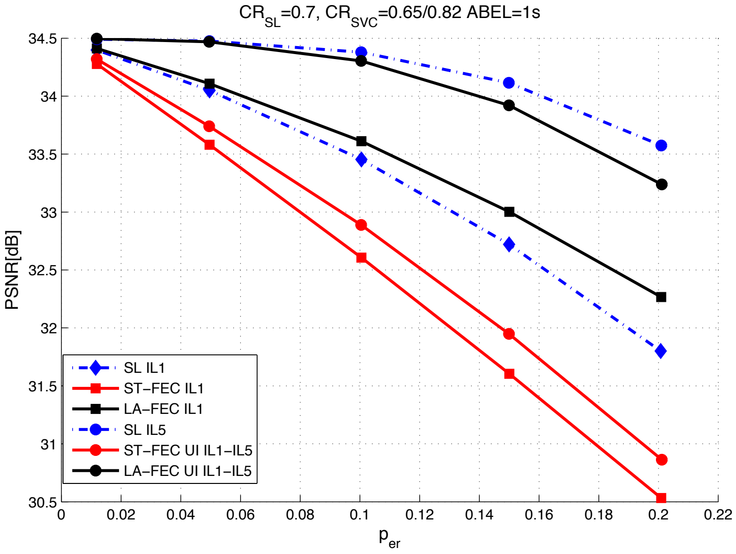

Figure 2.15: Video quality in terms of PSNR of selected settings comparing LA-FEC UI, ST-FEC, and SL in case of Unequal Error Protection.

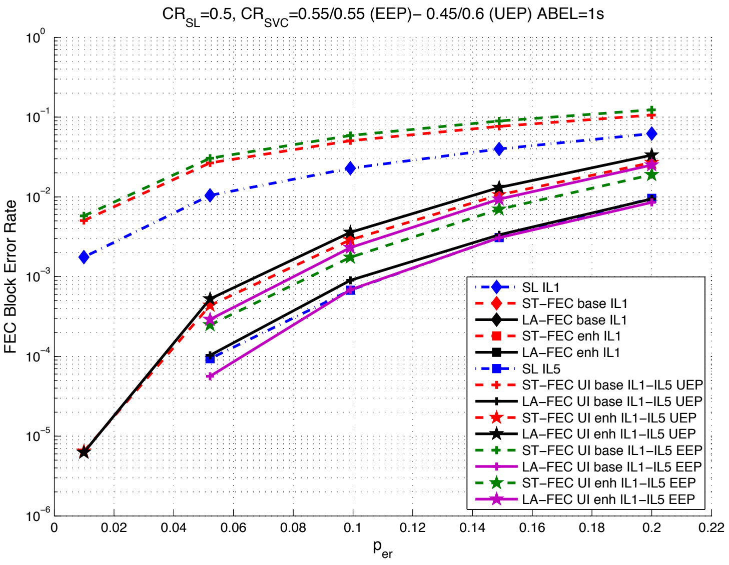

Figure 2.16: Equal Error Protection vs Unequal Error Correction. FEC Block error rate performance. $CR_{SL} = 0.5$, $CR_{SVC}^{EEP} = 0.55/0.55$, $CR_{SVC}^{UEP} = 0.45/0.6$, $ABEL = 1sec$.

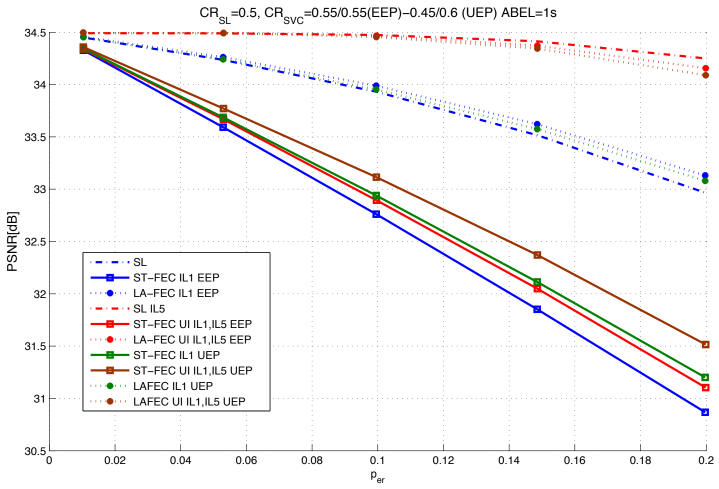

Figure 2.17: Equal Error Protection vs Unequal Error Correction. PSNR performance. $CR_{SL} = 0.5$, $CR_{SVC}^{EEP} = 0.55/0.55$, $CR_{SVC}^{UEP} = 0.45/0.6$, $ABEL = 1sec$.

bile networks to rural areas. However, mobile reception by satellite is characterized by shadowing periods where the Line-Of-Sight (LoS) is lost. A possible way to overcome these outage periods is to provide the means for implementing long time interleaving. Nonetheless, existing approaches for long time interleaving come at the cost of a long service tune-in time, which may last in the satellite case up to several seconds [43]-[40]-[44]-[45]. In this context, the LA-FEC solution may represent a valid alternative to existing solutions, since it enables fast tune-in into the service, while granting media streams a good level of robustness against error bursts. The rest of this section shows the achievable performance of the LA-FEC UI approach in comparison to existing solutions. The LA-FEC UI performance is compared with that of the ST-FEC UI case and of the SL IL case. Results are provided in tabular form and they are expressed in terms of failure decoding probability and overall media quality. The failure decoding probability is evaluated as a function of the layer tune-in time, while the overall media quality as a function of the service tune-in time. The **service tune-in time** is the time instance in which a receiver starts playing out the first picture of the video, while the **layer tune-in time** is the time instance in which the first picture of a certain quality layer can be played out by the receiver. The failure decoding probability is the probability of failing the decoding of a certain percentage of FEC DBs over the total number. The overall quality is expressed in terms of PSNR. For details on media stream characteristics and parameters, please refer to section 2.4.1.1. For the two considered SVC-FEC schemes, both Equal Error Protection (EEP) and Unequal Error Protection (UEP) have been tested. The two error protection schemes result in similar performance, hence for sake of simplicity the provided results will address only the EEP case.

2.4.2.1 Simulation Set and Results

The constraint to be satisfied is again that of 1 second of service tune-in time, like specified in the MBMS requirements [37]. In order to assess the performance of the DVB-SH system and to compare different modes, the DVB-SH implementation guidelines contain simulation results for representative parameter sets [46]. Annex A.9 in [47] contains a comprehensive list of representative simulation cases. In this work, two of them have been used, more in detail the ID14 and ID18 referenced in the table A.14 in [47], in the following addressed as ID1 and ID2 respectively. Both of them are related to a vehicular transmission. As channel model an LMS-ITS (Land Mobile Satellite- Intermediate Tree Shadow) [48] is used. For simulation purposes, these error traces have been mapped into IP transport stream composed of virtual transmission blocks (TB). Each TB is mapped into a MPEG-2 TS (Transport Stream) packet, reaching a full mapping of the error traces on the IP stream. Physical layer parameters of the used error traces are reported in Table 2.1, while other additional parameters are listed in Table 2.2 where MPEG-2 TS packets directly refer to TBs. Both error traces present a bit rate of 420 kbps per service with 8 services per channel. The average error rate is 0.3 for ID1 and 0.25 for ID2. The video sequence used for performing the simulations is composed of 32 FEC DBs, which are transmitted with 11600 TBs. Each FEC DB contains one second of media data, hence each FEC DB is made up of approximately 360 TBs.

Physical Setting Parameters	Trace ID1	Trace ID2
FEC Block Length	12282 bits	12282 bits
Coding Rate	1/4	1/2
Modulation Order	16 QAM	QPSK
Sub Carriers per OFDM sym.	1512	1512
OFDM Symbol Duration	0.000448 s	0.000448 s
Word Length	1504 bits	1504 bits
Frame Length	391168 bits	391168 bits
Nb FEC Blocks	125528 bits	125528 bits
Speed	50km/h	50km/h
C/I	11.9 dB	11.9 dB
C/N	10.8 dB	11.2 dB
Nb FEC CW per Burst	8	8
Burst Duration	925 ms	925 ms
Burst Period	122 ms	122 ms

Table 2.1: Physical Layer Parameters [2].

Parameters [MPEG-2TS packets]	Trace ID1	Trace ID2
Total Number of MPEG2TS simulated	1004224	1004224
Total Number of Burst	982	981
Average Burst Error Length (ABEL)	329.9491	282.1804
Error Burst Length Standard Deviation	307.8513	283.9807
Minimum Burst Length	7	7
Maximum Burst Length	3463	3431
Average Error Rate	0.3	0.25

Table 2.2: Channel Traces additional parameters [2].

Three exemplary cases have been evaluated, no interleaving (IL1) and interleaving lengths of 5 (IL5) and 10 (IL10) for both SL and SVC. As already pointed out, in the SVC case the interleaver is applied only to the EL. In addition, for the SL case, IL greater than one does not fulfill the assumed tune-in condition and they are integrated as references. The results are shown in terms of PSNR of the received sequence and in terms of failure decoding probability against the service tune-in time and the layer tune-in time respectively. For each FEC scheme considered, the best performing setting in terms of failure decoding probability granting a tune-in time of 1 sec is reported in bold. The same setting are reported in bold also in terms of PSNR, showing that in the ST-FEC case, an improvement of the enhancement layer in terms of failure decoding probability does not equivalently results in an improvement in terms of overall quality.

Settings	PSNR[dB]	Service tune-in time[s]
SL IL1	**31.0955**	**1**
SL IL5	33.6408	5
SL IL10	34.2351	10
ST-FEC IL1:IL1	30.5969	1
ST-FEC IL1:IL5	30.4460	1
ST-FEC IL1:IL10	**29.4262**	**1**
LA-FEC IL1:IL1	31.9953	1
LA-FEC IL1:IL5	32.6832	1
LA-FEC IL1:IL10	**33.4131**	**1**

Table 2.3: Trace ID1. PSNR value and service tune-in time for all the settings considered [2].

Tab. 2.3-2.4 report the results related to the packet error trace ID1, while Tab.

2.5-2.6 are related to the packet error trace ID2. In Tab. 2.3-2.5 the PSNR values have been evaluated with respect to the service tune-in time. SL IL5 and SL IL10 performance are used as upper bounds. As highlighted by numerical results, the LA-FEC UI outperforms the ST-FEC UI case in terms of PSNR. Focusing on Tab. 2.3, the gain introduced by our method is significant for both IL1-IL5 and IL1-IL10. The same considerations can be done in relationship to Tab. 2.5.

Settings	Failure Decoding Probability	Layer tune-in time[s]
SL IL1	**0.1127**	**1**
SL IL5	0.041938	5
SL IL10	0.014798	10
ST-FEC IL1:IL1 BL	0.1098	1
ST-FEC IL1:IL1 EL	0.1882	1
ST-FEC IL1:IL5 BL	0.1570	1
ST-FEC IL1:IL5 EL	0.077419	5
ST-FEC IL1:IL10 BL	**0.2052**	**1**
ST-FEC IL1:IL10 EL	**0.025381**	**10**
LA-FEC IL1:IL1 BL	0.0697	1
LA-FEC IL1:IL1 EL	0.1634	1
LA-FEC IL1:IL5 BL	0.0652	1
LA-FEC IL1:IL5 EL	0.11518	5
LA-FEC IL1:IL10 BL	**0.0491**	**1**
LA-FEC IL1:IL10 EL	**0.0625**	**10**

Table 2.4: Trace ID1. Failure Decoding Probability and layer tune-in time for all the settings considered [2].

Settings	PSNR[dB]	Service tune-in time[s]
SL IL1	**32.6408**	**1**
SL IL5	34.0277	5
SL IL10	34.2358	10
ST-FEC IL1:IL1	31.8048	1
ST-FEC IL1:IL5	30.5252	1
ST-FEC IL1:IL10	**29.5392**	**1**
LA-FEC IL1:IL1	32.8298	1
LA-FEC IL1:IL5	33.4592	1
LA-FEC IL1:IL10	**34.1085**	**1**

Table 2.5: Trace ID2. PSNR value and service tune-in time for all the settings considered [2].

In Tab. 2.4-2.6 the failure decoding probability of each layer is shown in function of the interleaver length(s) and of the layer tune-in time. Even in this case, the performance of the SL cases with IL>1 are reported as upper bound.

Settings	Failure Decoding Probability	Layer tune-in time[s]
SL IL1	**0.0522**	**1**
SL IL5	0.020495	5
SL IL10	0.014757	10
ST-FEC IL1:IL1 BL	0.0773	1
ST-FEC IL1:IL1 EL	0.1061	1
ST-FEC IL1:IL5 BL	0.1556	1
ST-FEC IL1:IL5 EL	0.039787	5
ST-FEC IL1:IL10 BL	**0.1891**	**1**
ST-FEC IL1:IL10 EL	**0**	**10**
LA-FEC IL1:IL1 BL	0.0471	1
LA-FEC IL1:IL1 EL	0.1096	1
LA-FEC IL1:IL5 BL	0.0351	1
LA-FEC IL1:IL5 EL	0.073611	5
LA-FEC IL1:IL10 BL	**0.0162**	**1**
LA-FEC IL1:IL10 EL	**0.016164**	**10**

Table 2.6: Trace ID2. Failure Decoding Probability and layer tune-in time for all the settings considered [2].

As can be derived from Tab. 2.4, for IL>1 the LA-FEC UI approach outperforms both the ST-FEC and the SL IL1. With IL1-IL5, IL1-IL10 the LA-FEC BL presents lower failure decoding probability compared with the other settings, highlighting the power of the proposed method, which at the same time allows a tune-in time of 1 sec and improves the decoding probability. The same trend can be observed in Tab. 2.6. The best performance is provided by the LA-FEC base IL1-IL10 for both error patterns. The results obtained are in line with the average error rate of the error traces used for performing the evaluation. In fact, ID1 is characterized by an average error rate of 0.3, while ID2 presents a lower error rate equal to 0.25. With LA-FEC UI, the performances of both layers are significantly improved thanks to the intrinsic nature of the LA-FEC UI approach. The LA-FEC UI allows the base layer to benefit from the longer interleaver applied to the enhancement layer. With the ST-FEC the base layer does not benefit form the longer interleaver applied to the enhancement layer. In fact in the ST-FEC approach each layer is protected independently, hence the apparent gain reported by this scheme in terms of failure decoding probability of the enhancement layer does not result in a gain in terms of overall quality as already highlighted above.

2.5 Conclusions

In this chapter, the scheme Layer Aware Forward Error Correction with Unequal Time Inteleaver (LA-FEC UI) has been introduced in detail. The proposed scheme is able to serve as UL-FEC solution for the two video communication applications considered. In both cases, it outperforms the state-of-the-art single layer coding and equivalent standard FEC approaches. All the above considerations and simulation results have been focused on demostrating the superiority of the addressed solution in applications in which the robustness against burst losses is not the only requirement. In fact, the joint application of layered media, LA-FEC and unequal time interlever enables service provisioning able to achieve a fast tune-in time of 1 second and an enhanced robustness to transmission error/losses with respect to existing solutions fullfilling the same tune-in time requirement. In addition, simulation results allow also to evaluate the performance of the proposed solution with respect to equivalent solutions in terms of time diversity (SL IL5). Also for this latter case, the achieved performance are satisfying especially considering that the LA-FEC UI does not require any system modification and the introduced overhead is unchanged with respect to single layer transmissions. In conclusion, the LA-FEC UI solution is a valid alternative to existing solution, it assures reliable video transmission and a fast access to the services which makes it suitable for real-world conditions.

It is worth noticing that, the literature in the field proliferates of solutions for unequal error protection of layered media flow, some of which are really challenging solutions. Among the others, the Priority Encoding Transmission (PET) scheme, introduced by Albanese et el. in [49] is overall one of the example and many alternatives have been builded upon it. Nonetheless, the methodoloogy herein presented can be easily adapted in order to suit the way in which multiple sub-bitstream are generated or the transmission is prioritized. In addition, the main outcome of the proposed LA-FEC UI methodoly is its capacity of enabling fast tune-in to video services while providing a level of robustness sufficient also to avoid service interruptions.

CHAPTER 3

MDLA-FEC DECODING FOR IPTV AND BROADCASTING TV APPLICATIONS

3.1 Motivation and Goals

The video technology field is considerably expanding and a strong driver is the interest towards multidimensional media flows, such as 3D movies, free-viewpoint applications, and many others. This new generation media stream is extremely challenging and brings several advantages. Viewers' experience is strongly enhanced, people do not simply watch a movie anymore, but they feel immersed in it.

A multidimensional media stream has to be efficiently compressed for being delivered over any kind of communication network, because its bandwidth requirements are much higher than any two-dimensional media stream. This can be efficiently achieved with ad-hoc encoding tools, such as the well know Multiview Video Coding (MVC) extension of the H.264/AVC standard [50] or the emerging High Efficiency Video Coding (HEVC) standard, which already encloses tools for multidimensional media. The encoded video stream is generally organized into sub-streams, or layers, with strong inter-layer dependencies. Each layer provides additional quality refinements or additional views on top of lower layers.

In the following, the term **multi-layer/multi-dimensional media stream** will be generally used for addressing layered encoded multidimensional media streams. This class of streams may be used to serve the heterogeneity of receivers delivering a single encoded bit stream in a cost-efficient way: one multi-layer/multi-dimensional bitstream can be broadcasted for serving all receivers classes, rather than broadcasting a single-layer bit-stream for each targeted receiver/quality. At the decoder end, each user can decode only the quality it is able to decode, thus it can exploit only data belonging to the set of sub-streams providing its targeted quality/service. This means that, users equipped with 3DTVs can enjoy the contents in three dimensions and full quality by decoding the whole received bit-stream, HDTVs' users will display the same content in HD quality by considering the set of sub-bitstreams needed for decoding a 2D video in HD quality only, and, similarly, mobile receivers will display the lower quality video, by decoding the basic quality layer only. All the receivers can thus enjoy the same visual content, although with different quality levels, by receiving the same bitstream.

The potential benefits of this multi-layer approach are easy to deduce, nonetheless also some disadvantages exist. One of the most sensitive aspects of multidimensional media streams is the prediction structure employed in the compression stage. The inter-layers dependency structure deriving from the prediction mechanisms employed can result very complex. As a consequence, the decoding complexity of such a kind of layered media is considerably increased. In addition, the correlation among layers highly influences the robustness of these flows against channel impairments. For these reasons, efficient mechanisms of reliable video transmission are mandatory.

In Chapter 2, the LA-FEC UI approach has been introduced and it has be proved to be a suitable solution for protecting scalable video flow (SVC/MVC) during transmissions over lossy channels. As described in [8], the basic idea of the LA-FEC approach is to generate the redundancy over layers in a media stream following existing dependencies, in such a way that higher layers redundancy symbols can also be employed as additional protection of lower layers. Stemming from the assumption that a certain media layer $l \in \{0, ..., L-1\}$ can be decoded if and only if all lowers layers have already been decoded, the concept of *dependency path* can be introduced. A **dependency path** (DP) is the sub-set of layers on which a certain media layer depends, sorted in order of importance. In other words, the dependency path of a media layer comprises all the lower layers that must be decoded in order to allow the decoding of the media layer itself. Given the above, it can be said that, at the decoder side, the LA-FEC approach exploits all the redundancy symbols in the same dependency path for combined error correction.

In [51] an extension of the LA-FEC approach to dependency structure in more SVC dimensions has been proposed. The authors describe the generation of redundancy symbols along three dependency paths, according to the temporal, spatial and fidelity dimension and the dependency path scheme of each layer. Rateless codes have been employed as the FEC scheme. In order to account for inter-layer dependencies, FEC data of a certain layer are generated across all layers belonging to its dependency path which may include lower layers within the same scalability dimension as well as layers belonging to other dimensions. This way they can be jointly used for achieving an enhanced robustness for this kind of media stream against transmission impairments. It is worth noticing that, the lowest quality layer is included in all FEC symbols. Hence, there are multiple paths where redundancy symbols can be jointly used for correcting errors in it.

In this chapter, a further extension of the LA-FEC procedure is presented. The multidimensional encoding scheme presented in [51] has been used as a basis, but some substantial differences have been introduced in the considered view setup. The proposed extended scheme is named Multi-Dimensional/Multi-Layer Aware Forward Error Correction (MDLA-FEC). The redundancy symbols are generated along three dependency paths according to the temporal, spatial, spatio-temporal and quality dimensions. More in detail, in the following, a scheme of multilayer/multidimension dependency structure, suitable and compliant with real systems implementations, is provided and analyzed in function of the transmission system employed, i.e. traditional TV broadcasting or IPTV. The dependency path of each quality level is derived for both cases and their compact formulaic representation is provided. In addition, a mathematical formula of the decoding probability of a FEC protected multidimensional data block is given for both Broadcasting and IPTV.
The performance analysis of the presented method is, for the moment, out of the scope of this dissertation.

3.2 Theoretical Background

3.2.1 Multidimensional media stream and view setup

The targeted multidimensional media stream comprises 9 sub bitstreams, more generally addressed as layers. One of these layers provides basic quality and it is common to all the dependency paths. In the following, it will be addressed as Base Layer (BL). All other layers, Enhancement Layers (ELs), provide enhanced quality on the top of the BL over three possible scalability dimensions: temporal (framerate), spatial (number of views) and spatio-temporal and quality (from 2D-mobile up to multiview HD), indicated with T, S and STQ respectively. For the considered view

setup, along the vertical dimension (T), higher frame rate sequences (i.e. 25 fps, 50 fps, 100 fps) are provided, along the horizontal dimension (S) each enhancement layer adds a new view (i.e. 2D, Stereoscopic, Multiview), while the diagonal dimension (STQ) is a combination of the two: each enhancement layer adds a new view at an increased frame rate (i.e. 2D/25 fps, Stereo/50fps, Multiview/100fps).
Note that, the proposed 9-layer system is just exemplary. The model that follows can be either extended or down-scaled to a different number of layers and adapted to different layer-dependency structures.

Indicating with the term **Operation Point** the set of sub-streams needed for decoding a predefined video quality/service, the described setup can be represented in matrix form by using the following **Operation Points Matrix** (M_{OPs}), in which each element represents an operation point. Each element of the matrix is univocally identified by two indexes, one represents its position along the horizontal dimension and the other along the vertical one.

$$M_{OPs} = \begin{bmatrix} (3,1) & (3,2) & (3,3) \\ (2,1) & (2,2) & (2,3) \\ (1,1) & (1,2) & (1,3) \end{bmatrix}$$

Indicating with the term **dependency path** of a layer/OP of a multi-layer/multi-dimensional media stream the sorted list of sub-streams on which it depends on, that may belong to the same dimension only or may be inter-dimensional. An $OP_{(A,B)} \in M_{OPs}$ can exploit all the sub-streams belonging to its dependency path for decoding. For example, the dependency path of the operation point indexed (2,2) is the set $DP(OP_{(2,2)})$=$\{(1,1),(1,2),(2,1),(2,2)\}$. This means that, at the encoder side, FEC data of $OP_{(2,2)}$ are generated over all those layers. At the decoder end, FEC data of all layers in the OP's dependency path can be exploited for combined decoding.

Note that, in this work, the dependency path is not only a function of the inter-layer dependency structure but also of the application, i.e. Broadcast or IPTV. In order to help with the comprehension of the mathematical description which follows, in Fig. 3.1 the exemplary three-dimensional view setup is graphically represented and a one-to-one mapping of this three-dimensional system and the M_{OP} is provided in Fig. 3.2.

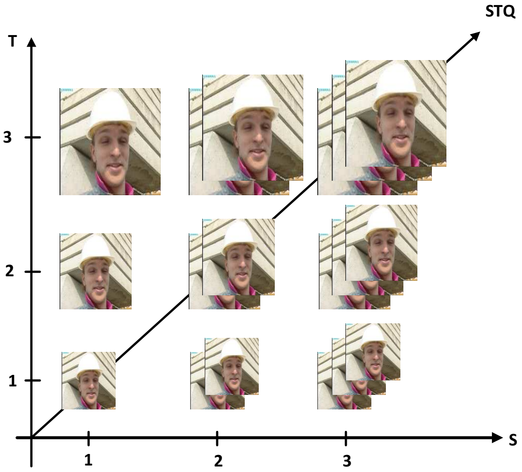

Figure 3.1: View setup.

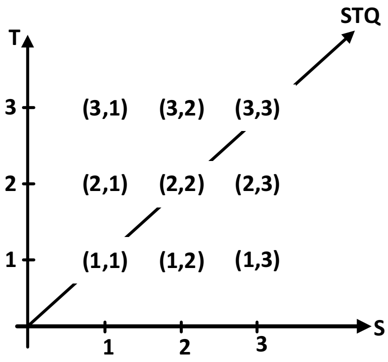

Figure 3.2: Mapping of M_{OPs} elements on the three dimensional space (S,T,STQ).

3.2.2 Broadcasting TV vs IPTV

In this context, the activity presented here entails the study of a mathematical approach for evaluating the decodability of FEC-protected multidimensional media streams taking into account the inter-layer dependency structure derived from the compression process and the employed transmission system. As transmission system, in the following, Broadcasting TV and IPTV are considered.

- Broadcasting TV: the encoded multidimensional media stream is FEC-protected and transmitted at once. The whole media stream is received, eventually loss affected. Receivers can decode the desired quality extrapolating all the sub-streams belonging to the dependency path of the targeted OP. In addition, if needed, also FEC data of layers not belonging to the current dependency path can be exploited for decoding, being actually available at the receiver end.

- IPTV: the encoded multidimensional media stream is FEC-protected and all the comprised sub-bitstreams are delivered individually in different multicast flows. In function of the targeted quality, receivers will join as many multicast groups as needed for decoding the corresponding OP. This means that a final user has to join all the multicast groups bringing data belonging to the OP's dependency path. As a consequence, even if needed, FEC data of sub-bitstreams not belonging to the dependency path of the considered operation point cannot be exploited for FEC decoding.

3.2.3 From LA-FEC to MDLA-FEC

As introduced above, the methodology herein presented is an extension of the LA-FEC scheme to multidimensional setups where coding dependencies along the layered source message stretch along several different dimensions. The proposed model brings also the advantage of being "transmission system aware" in the sense that the given formulaic representation is a function of the underlying transmission system employed. It is important to note that this solution is tailored for UL-FEC codes, whose elementary units are data packet.

At the encoder end, the transmission system employed is not relevant: the FEC data generation depends only on the FEC scheme employed.

If traditional FEC approaches (ST-FEC) are used, each layer is FEC-protected individually. Intrinsic correlations existing among media layers are not taken into account. Redundancy symbols of a certain layer/OP are generated over its source data only. This means that, for decoding a certain OP, FEC data of its dependency path can not be used for combined FEC decoding, hence higher layers FEC are useless for lower layers protection.

If the MDLA-FEC approach is used, each enhancement layer FEC data can be used for protecting lower layers in the same dimension and/or dependent layers in the other dimensions. FEC data of a certain OP are generated over source data of all layers belonging to its dependency path. Due to the multidimensional nature of the tailored media streams, the dependency path of a certain OP may comprise sub-streams belonging to more than one scalability dimensions. This enables the possibility of exploiting not only inter-layer dependencies within the same scalability dimension, but also inter-dimensional dependencies. The introduced gain is manifold: from one side the introduced scheme may be an useful tools for coding strategy optimization, from another it allows to exploit this complex structure for efficient FEC schemes design.

As an example, consider the operation point indexed as $(2, 2)$ and the situation presented in Fig. 3.2. This OP stands on the STQ dimension, hence the sub-bitstreams needed for its decoding belong to two scalability dimensions and its dependency path is $DP(OP_{(2,2)})=\{(1, 1), (1, 2), (2, 1), (2, 2)\}$. Its parity symbols will be calculated over the source symbols of all those sub-streams enabling a combined FEC decoding at the receiver end and providing additional protection to all of them. Nonetheless, due to media dependency, $OP_{(2,2)}$ can be decoded only if all lower layers have been decoded as well. In the decoding procedure of $OP_{(2,2)}$, redundancy symbols of layers $(1, 2), (2, 1)$ and $(2, 2)$ can be exploited for decoding layer $(1, 1)$. In addition, in the Broadcasting case only, also FEC data of layers $(3, 1), (3, 2), (3, 3), (1, 3), (2, 3)$ may be used. The robustness of the media stream results strongly enhanced by this mechanism.

3.3 MDLA-FEC Decoding for Broadcasting and IPTV services

In this section the mathematical description of the decoding procedure of the MDLA-FEC is provided. More in detail the dependency path for each of the elements of the above Matrix of Operation Points M_{OPs} is calculated in function of the transmission system actually employed and in consideration of the intrinsic media layer interdependency structure. As highlighted above, the tailored transmission

system determines the actual bitstream available at the decoder end. For each matrix element, or OP, two sub-sets have to be defined, one representing all the lower sub-streams on which it depends, and another enclosing the higher sub-streams, within the sub-set actually available, that can be exploited for decoding thanks to the MDLA-FEC encoding procedure. These sub-sets are named Lower Dependency Set (LDS) and Upper Dependency Set (UDP) respectively.
The dependency structure of layers within the considered view setup is rather complex and describing it mathematically is not straightforward. The approach herein presented provides a solution by describing the inter-layers dependency structure in a clever way and in function of the employed transmission system. In addition, it details how to exploit the dependency structure in the decoding procedure and it leads to a compact formulaic representation of the condition to be satisfied for successful decoding of a certain operation point, for a given erasure probability. The resulting mathematical model, provides a compact and rather general representation of inter-dependencies in multi-layer/multi-dimensional streams. It enables fast calculation of whether a loss affected FEC-protected data stream may be decoded or not, for a targeted channel erasure probability. Hence, it can be a valid tool in performance evaluation. The introduced model is "optimum" from the decoder point of view, in the sense that it can exploits all FEC-data actually available at the decoder end for combined error protection. This aspect can be strongly beneficial for Broadcasting TV systems' performance.

3.3.1 IPTV

- Lower Dependency Set.
  The lower dependency set of a certain $OP_{(A,B)}$, indicated by $\Omega^{IP}_{A,B}$, contains all the couple of indexes $<(a,b)>$, each indicating a sub-stream on which it depends, within the set of layers available at the decoder end. Note that, the lower dependency set of $OP_{(A,B)}$ is equivalent to its dependency path.

- Upper Dependency Set.
  The upper dependency set of a certain sub-stream $<(a,b)>$ when $OP_{(A,B)}$ is targeted is indicated by $\Psi^{IP}_{<(a,b)>(A,B)}$. It contains all higher layers - within the available ones - which can be additionally exploited for decoding $<(a,b)>$ thanks to the MDLA-FEC scheme employed.

These sub-sets (LDS and UDS), in the context of IPTV applications, are mathematically described as follow:

$$\Omega^{IP}_{A,B} = \{<(a,b)>: \quad a \leq A \quad b \leq B\} \tag{3.1}$$

$$\Psi^{IP}_{<a,b>(A,B)} = \{(x,y): \quad a \leq x \leq A \quad b < y \leq B\} \tag{3.2}$$

Let $P^{IP}_{OP_{(A,B)}}$ be an indicator function which assumes value 1 for indicating successful decoding of $OP_{(A,B)}$ and value 0 otherwise. The following equation is the MDLA-FEC decoding condition that a generic OP should satisfy in order to be decodable.

$$P^{IP}_{OP_{(A,B)}} = \Lambda(\Omega^{IP}_{A,B})\{\Theta(\Psi^{IP}_{<(a,b)>(A,B)})[\sum_{t=a}^{x}\sum_{s=b}^{y}(r_{t,s}) \geq \sum_{t=a}^{x}\sum_{s=b}^{y}(k_{t,s})]\} \tag{3.3}$$

where:

$$\Lambda(\Omega^{IP}_{A,B}) = \bigwedge_{\forall<(a,b)>\in\Omega^{IP}_{A,B}} \tag{3.4}$$

$$\Theta(\Psi^{IP}_{<(a,b)>(A,B)}) = \bigvee_{\forall(x,y)\in\Psi^{IP}_{<(a,b)>(A,B)}} \tag{3.5}$$

and $r_{t,s}$, are the number of received symbols of the sub-stream (t,s) after transmission over a loss affected channel characterized by a certain erasure probability p_{er}, while $k_{t,s}$ are the number of source symbols that sub-stream (t,s) originally comprises.

Eq. 3.3 can be divided in two parts: an outer condition and an inner condition. The outer condition (see Eq. 3.4) verifies that all the sub-bitstreams belonging to the OP's LDS (thus to its dependency path) can be decoded. This is done performing a logic AND operation in between the decoding condition of each media layer. The inner part (see Eq. 3.5) exploits the UDS of each sub bitstream in order to exploit available FEC data. Since additional layer FEC data is used only if needed, it is enough to satisfy only one of the decoding conditions in order for the whole inner condition to be satisfied, a logic OR serves the scope.

3.3.2 Broadcast

- Lower Dependency Set.
  The lower dependency set of a certain $OP_{(A,B)}$, indicated by $\Omega^{BC}_{A,B}$, is the set of couple of indexes $<(a,b)>$, each indicating a sub-stream on which it depends within the set of available layers at the decoder end.

- Upper Dependency Set.

  The upper dependency set of a certain sub-stream $<(a,b)>$ when $OP_{(A,B)}$ is targeted is indicated by $\Psi^{BC}_{<(a,b)>(A,B)}$. It contains all higher layers - within the available ones - which can be additionally exploited for decoding $<(a,b)>$ thanks to the MDLA-FEC scheme employed.

The formulas of the LDS and UDS sub-sets for the broadcasting case are the following:

$$\Omega^{BC}_{A,B} = \{<(a,b)>: \quad a \leq A \quad b \leq B\} \tag{3.6}$$

$$\Psi^{BC}_{<(a,b)>(A,B)} = \{(x,y): \quad a \leq x \leq MaxLayerPerDim \quad b \leq y \leq MaxLayerPerDim\} \tag{3.7}$$

Let $P^{BC}_{OP_{(A,B)}}$ be an indicator function which assumes value 1 for indicating successful decoding of $OP_{(A,B)}$ and value 0 otherwise. The following equation is the MDLA-FEC decoding condition that a generic OP should satisfy in order to be decodable.

$$P^{BC}_{OP_{(A,B)}} = \Lambda(\Omega^{BC}_{A,B})\{r_{a,b} \geq k_{a,b} \vee [\sum_{\forall(t,s)\in\Psi^{BC}_{<(a,b)>(A,B)}} (r_{t,s}) \geq \sum_{\forall(t,s)\in\Psi^{BC}_{<(a,b)>(A,B)}} (k_{t,s})]\} \tag{3.8}$$

where:

$$\Lambda(\Omega^{BC}_{A,B}) = \bigwedge_{\forall<(a,b)>\in\Omega^{BC}_{A,B}} \tag{3.9}$$

and $r_{t,s}$ are the number of received symbols of the sub-stream (t,s) (after transmission over a loss affected channel characterized by a certain erasure probability p_{er}) and $k_{t,s}$ represents the number of original source symbols of layer (t,s).

In the Broadcasting scenario, independently from the quality targeted by the final user, the whole media stream is available at the receiver end. FEC data of all composing sub-bitstreams can be exploited in the decoding procedure.

The condition for successful decoding of a certain (OP) is that all layers in its DP can be decoded. The logic AND operation over all these layers performs this check. In addition, for FEC decoding of each dependent layer, all the FEC-data of the media stream can be exploited, as expressed by the inner logic OR operations in Eq. 3.8.

3.4 Conclusions

In this chapter the multi-dimensional/multi-layer FEC (MDLA-FEC) decoding procedure has been mathematically described in function of the transmission system infrastructure employed for an exemplary view setup. The presented approach is an extension of the LA-FEC concept to more scalability dimensions and it can be applied on the top of SVC/MVC/HEVC encoded streams, therefore it may be suitable in describing decoding conditions of forthcoming transmission setups when the FEC-encoding is performed accordingly with the layer/dimension aware approach. The concepts of Operation Point and Lower and Upper Dependency Sets have been introduced. The mathematical model provides a compact mathematical representation of the dependency path of each MDLA-FEC protected sub-layer, by means of LDS and UDS and of the decoding condition to be satisfied in function of the complex inter-layer dependency structure and of the underlying transmission system. The presented activity is still open, the future developments in the short term are the performance evaluation of the MDLA-FEC scheme with respect to standard and single layer FEC schemes and the evaluation of its benefits in real system implementation. On the long term, it would also be interesting to investigate possible methodologies for adopting the proposed scheme for coding strategy optimization.

CHAPTER 4

MODELING OF IMPULSIVE NOISE EFFECTS ON AL-FEC SCHEMES

4.1 Motivation and Goals

In the previous chapters, techniques of reliable video transmission have been addressed proving to be suitable countermeasures to errors and losses which may occur over the transmission path. More in detail, the presented solutions mainly deal with burst loss effects, being this typology of loss representative of the errors occurring in the primary current broadcasting and streaming systems. It has been clarified, for example, that video delivery services targeting mobile receivers are unavoidably affected by deep fading events, lasting up to several seconds and resulting in consecutive packet losses, and that these losses can be detected, recovered and, if needed, concealed by means of techniques acting at different layers of the protocol stacks and applied at different stages of the transmission chain. Nevertheless, depending on the underlying physical layer transmission media employed, errors may also occur in different forms. A representative case is the impulsive noise - one of the dominant error sources in the signal band of DSL channels - characterized by fixed-length error bursts [52]-[53]. DSL provides a broadband connection on already existing twisted pair cables and it is widely used to serve as last mile technology in Internet Protocol Television (IPTV) architectures. Real-time video services, such as IPTV [39], are highly sensitive to packet losses. Therefore, transmission system design has to overcome packet losses introduced by network congestion or impulse noise bit errors on the physical layer, such as inflicted by the Repetitive Electrical

Impulse Noise (REIN).

In order to increase the robustness against impulse noise impairments, one countermeasure in IPTV systems design consists in the application of Forward Error Correction on the Application-Layer (AL-FEC) [54]. AL-FEC schemes require extensive performance analysis to ensure proper parameter setup. This can either be done based on real channel measurements [55] or by relying on channel models [56] to approximate the characteristics of the target channel. The state-of-the-art approach for determining the performance of an AL-FEC scheme under REIN influence is to rely on Monte-Carlo simulations, which has the drawback of large test sets and an accuracy that depends on the number of repetitions. Another way is to rely on a deterministic model obtained from the Markov chain, that enables fast and accurate performance analysis. In this chapter, a mathematical model for deriving the correction probability of an AL-FEC scheme after transmission over a channel characterized by randomly distributed error bursts with fixed length, is presented. More in detail, the focus has been posed on the REIN noise case, albeit the model can be applied also to impulse noise of higher duration and whenever the noise affecting the transmission has a constant duration.

The effect of physical layer REIN on the application layer can be modeled by a Markov chain with error bursts of fixed length. The model herein proposed serves exactly this scope and calculates, rapidly and accurately, the decoding probability of blocks of data delivered over a DSL line considering growing burst probabilities. Extensive simulation campaigns, in the context of AL-FEC protected IPTV services, are used to prove the correctness of the proposed model and additionally, the runtime behavior of the model is analyzed.

4.2 Theoretical Background

The following section provides the technical background for the proposed model, i.e. the impulse noise classification, the Markov chain modeling a channel with fixed-length error bursts and ideal Forward Error Correction codes.

4.2.1 Impulse Noise

Impulse noise is characterized by burst of energy spikes with random amplitudes and random inter-arrival time. It can have several sources, e.g. due to man action or to natural electromagnetic events, i.e. switching of electronic equipment in the

telephone network or other nearby disturbances. It can lead to the occurrence of short bursts of bit errors and therefore to packet losses on higher layers and it might almost destroy the DSL signal [52].
Describing statistically impulsive noise is not straightforward due to its non-stationary nature. Nevertheless, it can be classified based on amplitude, spectrum, burst duration and Inter-Arrival Time(IAT) [53],[57].

Based on its amplitude and duration, impulse noise can be classified as follow:

1. **Repetitive Electrical Impulse Noise (REIN)**
   Burst Length < 1[ms]

2. **Prolonged Electrical Impulse Noise (PEIN)**
   Burst Length $1 \div 10$ [ms]

3. **Single Isolated Impulse Noise (SHINE)**
   Burst Length > 10 [ms]

In this work, the attention has been focused on REIN effects on the performance of AL-FEC scheme.

4.2.2 Fixed length burst error channel

The transmission channel affected by fixed-length randomly distributed error bursts is modeled as an $(B_L + 1)$-state Markov chain with memory B_L, where B_L represents the burst length expressed in number of symbols. Figure 4.1 gives the associated state diagram of the Markov chain. Out of the $(B_L + 1)$ states, a state referred to as the *good state* represents reception of a symbol with probability p_R, a second state, referred to as the *bad state*, corresponds to the start of a burst with a probability p_B, where $p_B = (1 - p_R)$. All remaining states are burst states and each of them can only transit to the following burst state except for the last burst state. The last burst state can transit to the good state with probability p_R, in case no further bursts occur, or to the bad state with probability p_B, in case of a direct consecutive burst. Therefore, the model can be fully described with a transition probability matrix, containing the transition probabilities between states and a vector containing the initial distribution.

The transition probability matrix (P_T) is a $[(B_L + 1) \times (B_L + 1)]$ square and sparse matrix. Only the first and the last rows have more than one non-zero value. The sub-matrix obtained by excluding the first and the last rows and the first two columns is an identity matrix.

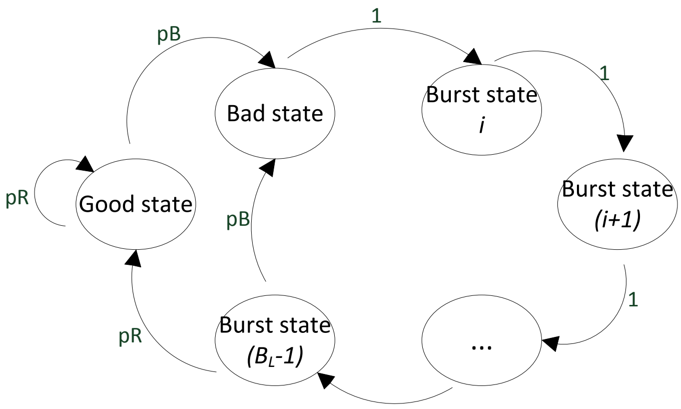

Figure 4.1: Markov chain state diagram describing randomly distributed error bursts with fixed length [4].

As an example, the transition matrix with $B_L = 4$ is reported in the following:

$$P_T \doteq \begin{pmatrix} p_R & p_B & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ p_R & p_B & 0 & 0 & 0 \end{pmatrix}$$

In addition to the transition probability matrix, the initial distribution is required to fully describe the model. It consists of a row vector p of $(B_L + 1)$ elements. As the addressed channel is not memoryless, knowledge of former states is necessary to calculate the current state. For this reason the initial distribution is used to initialize the channel and it is found by satisfying the following steady state condition:

$$p \cdot P_T = p$$

Therefore, with B_L=4, setting g as the probability of being in the good state and b as the probability of being in the bad state or in a burst state, the initial distribution vector is as follows:

$$p = [g \qquad b \qquad b \qquad b \qquad b]$$

where b and g are derived as $b = \frac{1}{(pR/pB)+BL}$, $g = (\frac{pR}{pB}) \cdot b$.

4.2.3 Ideal Forward Error Correction

It is worth noticing that, in this work, the focus is on ideal FEC codes.
For each block of k source symbols, p parity symbols are created. Then, the resulting encoding block consists of n symbols, where $n = (k + p)$ and the associated code rate is defined as $CR = k/n$.
For an ideal (n, k) FEC code a number of received symbols r greater or equal to the number of source symbols k is required to decode the data successfully.

4.3 The proposed model

In this section, the proposed model is presented in detail. The combinatorial analysis performed in order to identify all possible fixed-length burst distributions still leading to decodable settings is explained. Then, the mathematical model build upon it is introduced.

4.3.1 Combinatorial Analysis

The mathematical model herein presented is able to calculate if an n-symbol FEC-protected data block can be decoded or not in presence of burst losses. In order to do this, all the possible fixed-length burst combinations that lead to decodable data blocks have to be determined and the corresponding probabilities have to be summed up.

For helping with the presentation of the combinatorial analysis and of the corresponding mathematical description, the set of possible burst distributions is organized in three cases and associated sub-cases. In Fig. 4.2, illustrative examples with the characteristics of each case and sub-case are given. In order to consider all burst distributions that allow for decoding a certain data block (i-th), bursts that start in the preceding block (i-1)-th) and bursts that last until the following one ((i+1)-th) have to be taken into account as well and are referred to as *partial bursts*. The following gives a list of cases and sub-cases along with their distinct characteristics.

- **Case 1: Entire burst.**

  Up to maximum number of $Max_B = \lfloor n/B_L \rfloor$ entire bursts can be completely located within the i-th data block, where floor$(x) = \lfloor x \rfloor$ is the largest integer not greater than x.

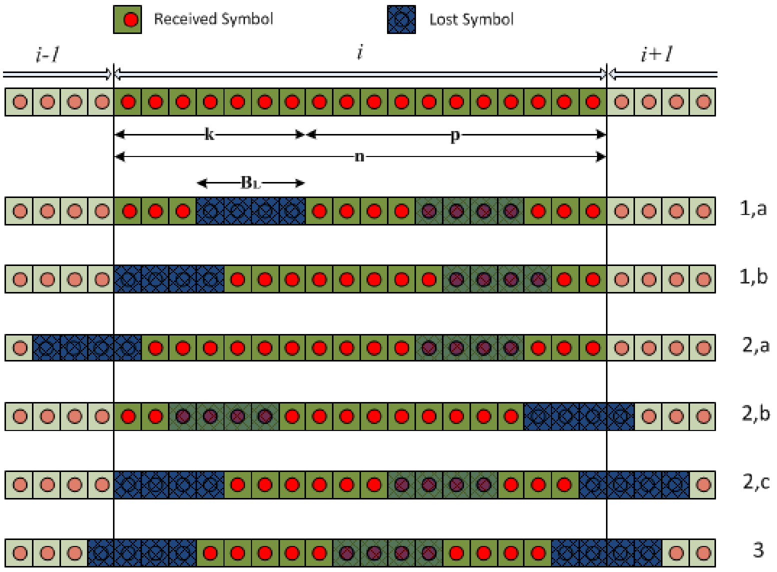

Figure 4.2: Illustration of various burst distributions [4].

  - **Case1a** The first symbol of the i-th data block is in good state and successfully received.
  - **Case1b** The first symbol of the i-th data block is not received and in bad state, i.e. the beginning of a burst.

- **Case 2: Initial or final partial burst.**

  Up to $(B_L - 1)$ symbols at the beginning or towards the end of the i-th data block are in burst state due to a partial burst that either affects the preceding or the following data block.

  - **Case2a** Initial partial burst: The first symbol of the i-th data block is not received due to a partial burst in the beginning, i.e. a burst occurs at the end of the (i-1)-th data block and affects the first symbol(s) of the current one. There is no final partial burst affecting the last symbol of the current data block.
  - **Case2b** Final partial burst: The first symbol of the i-th data block is successfully received while its last symbol is not received due to a partial burst, i.e. a burst occurs towards the end of the i-th data block and it affects symbols of the (i+1)-th data block as well.

- **Case2c** Final partial burst with loss of first symbol: The first symbol of the i-th data block is in bad state, hence a burst starts exactly at the first symbol of the data block. In addition, the last symbol of the i-th data block is not received due to a partial burst, i.e. a burst occurs towards the end of the i-th data block and affects symbols of the (i+1)-th data block as well.

All the sub-cases of Case 2 include the optional presence of further entire bursts within the i-th data block.

- **Case 3: Initial and final partial bursts.**

Up to $2 \cdot (B_L - 1)$ symbols at the beginning or towards the end of the i-th data block are in burst state due to partial bursts that affect both the preceding and following data blocks.

Case 3 includes the optional presence of further entire bursts within the i-th data block.

4.3.2 Mathematical Model

The decoding probability of an ideal (k,n) FEC code, given a certain random burst probability p_B, is equal to:

$$\begin{aligned} Pr(r \geq k) = & \sum_{B=0}^{Max_B} [Pr(Case1a)(B)+ \\ & + Pr(Case1b)(B) + Pr(Case2a)(B)+ \\ & + Pr(Case2b)(B) + Pr(Case2c)(B)+ \\ & + Pr(Case3)(B)] \end{aligned}$$

If the number of received symbols corresponding to one of the sub-cases listed above is smaller than the minimum amount of symbols required to satisfy the decoding condition, the corresponding probability will not be included in the sum.

Let x specify the number of received symbols when the transmission starts in good or burst state and y specify the number of received symbols when the transmission starts in bad state. It is necessary that the number B of bursts considered still satisfy the decoding condition, i.e. that x or y are actually greater or equal to the number of source symbols k.

Following the above with x and y as $x = n - B \cdot B_L$, $y = n - (B+1) \cdot B_L$ and binomial coefficient as $\binom{n}{k}$, we have:

- **Case 1: Entire burst.**

$$Pr(Case1a)(B) =$$
$$= \left\{ \binom{x-1+B}{B} \cdot g \cdot {p_B}^B \cdot {p_R}^{(x-1)} |(x \geq k) \right\}$$

$$Pr(Case1b)(B) =$$
$$= \left\{ \binom{y+B}{B} \cdot b \cdot {p_B}^B \cdot {p_R}^y |(y \geq k) \right\}$$

- **Case 2: Initial or final partial burst.**

When initial or final partial bursts are considered, there can be up to $(B_L - 1)$ deleted symbols either at the beginning or at the end of the considered data block.

$$Pr(Case2a)(B) =$$
$$= \sum_{i}^{B_L-1} \{Pr(Case2a)(B,i)|(x-i) \geq k\}$$

$$Pr(Case2b)(B) =$$
$$= \sum_{i}^{B_L-1} \{Pr(Case2b)(B,i)|(x-i) \geq k\}$$

$$Pr(Case2c)(B) =$$
$$= \sum_{i}^{B_L-1} \{Pr(Case2c)(B,i)|(y-i) \geq k\}$$

Where:

$$Pr(Case2a)(B,i) =$$
$$= \binom{x-i+B-1}{B} \cdot g \cdot {p_B}^{(B+1)} \cdot {p_R}^{(x-i-1)}$$

$$Pr(Case2b)(B,i) = \binom{x-i+B}{B} \cdot b \cdot {p_B}^{B} \cdot {p_R}^{(x-i)}$$

$$Pr(Case2c)(B,i) = \binom{y-i+B}{B} \cdot b \cdot {p_B}^{(B+1)} \cdot {p_R}^{(y-i)}$$

- **Case 3: Initial and final partial bursts.**

Partial bursts at the beginning of the data block and at its end are considered at the same time. This means that, there can be up to $2 \cdot (B_L - 1)$ deleted symbols in the data block.

$$Pr(Case3)(B) =$$
$$= \sum_{j=0}^{B_L-2} \sum_{i=2+j}^{2(B_L-1)-j} \{Pr(Case3)(B,i)|(x-i) \geq k\}$$

where:

$$Pr(Case3)(B,i) = \binom{x-i+B}{B} \cdot b \cdot {p_B}^{(B+1)} \cdot {p_R}^{(x-i)}$$

4.4 Simulation Set and Results

In order to investigate the correctness of the proposed model, extensive empirical simulations were carried on. The statistical relevance is ensured by performing 10000 repetitions per simulated point. Data blocks of different sizes and with different levels of protection have been simulated over a channel with fixed length bursts. The probability of decoding the FEC data block for increasing burst probabilities p_B was calculated. The different source data block sizes considered are: $k = 20$, 100 and 200, all expressed in number of symbols. As code rates, $CR = 0.3$, 0.5 and 0.7 have been tested for each of the source data block size considered. As illustrated above, the presented model takes as input data block comprised of both source and FEC data. For this reason, the actual data block length for a given source data block size varies in function of the applied code rate. Moreover, it can be easily calculated by summing the source data block size to the number of parity symbols, easily computable using the formula ($p = (1 - CR) \cdot k/CR$).

In terms of burst length size, two different cases have been targeted:

1. The burst length of fixed size expressed in terms of symbols.

2. The burst length of fixed size expressed in fractions of seconds and thus scaled by the code rate.

In the first case, a fixed length burst of size $B_L = 16$ symbols has been simulated. In the second case, the burst length is calculated by mapping the duration of the burst - expressed in msec - to a corresponding number of affected symbols. As a consequence, in this case, the burst size will vary with the considered code rate. In order to perform this mapping operation, the work hypothesis is to deal with a data block of 1 sec in length and bursts lasting for 100 msec. Simulation results will show on the x-axis the burst probability range and on the y-axis the corresponding decoding probability, for all considered settings.

Fig. 4.3 shows the resulting decoding probabilities of the empirical model for a source data block size of 100, considering a fixed burst length of 16 symbols, and the three code rate setups listed above and Fig. 4.4 shows the error signal between the results of the proposed model and those of the empirical simulations. From the figures, it can be seen that the results of both approaches correlate to a high extent and the remaining error signal between the two has the characteristics of random noise without any persisting bias.

Fig. 4.5-4.7 show the decoding probabilities achieved with the proposed model in comparison with those calculated by means of empirical simulations. Source data block size of 20, 100 and 200 respectively, considering fixed burst length size of 100 msec and the three code rate setups listed above, have been simulated. The error signal between results of the proposed model and of the empirical simulations shows similar performance with respect to the $B_L = 16$ case, reported above.

Also in this case, the results of both approaches correlate to a high extent and the remaining error signal has the characteristics of random noise without any persisting bias. The model performance, as well as that of empirical tests, scales accordingly with the error probability and the employed code rate. Lower code rate values, result in stronger protection against error bursts of any length. This trend is valid also when the burst length consists of a fixed number of symbols. The proposed model is then really flexible and allows to evaluate FEC data block decoding probability

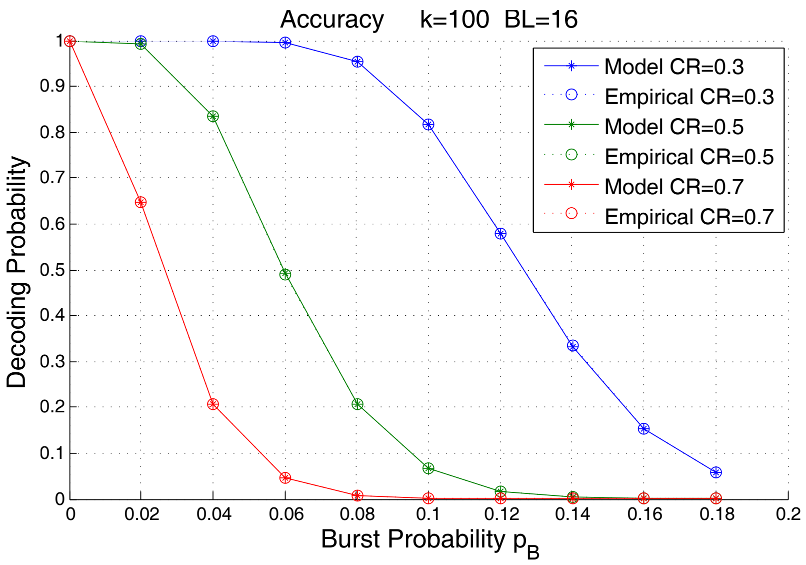

Figure 4.3: Comparison of the proposed model with empirical simulations for source data block of $k = 100$ and B_L=16 symbols [4].

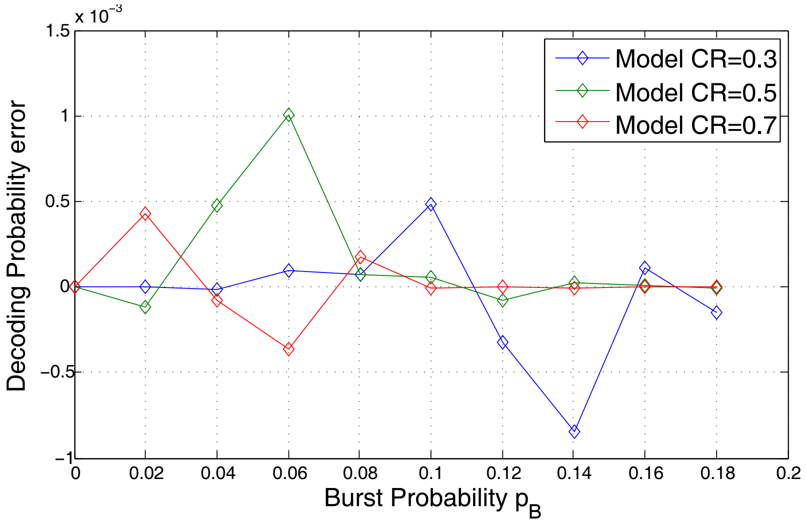

Figure 4.4: Error signal between proposed model and empirical simulations [4].

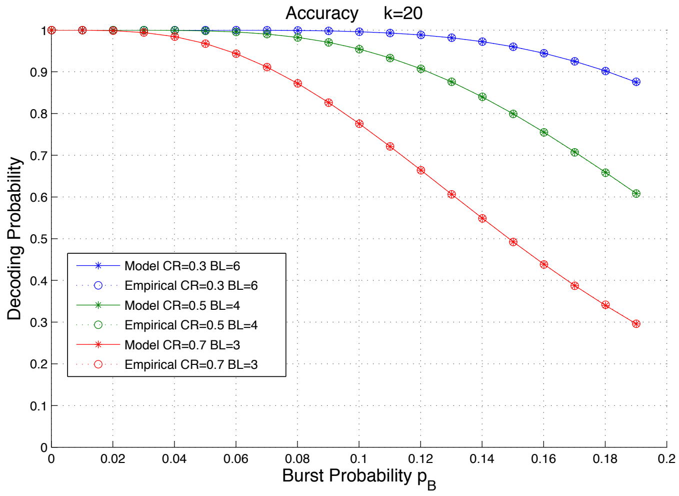

Figure 4.5: Comparison of the proposed model with empirical simulations for source data block of $k = 20$ and B_L=100 msec.

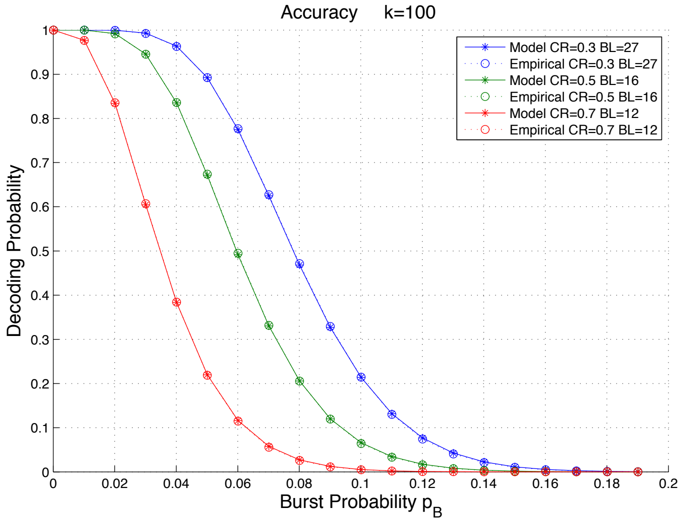

Figure 4.6: Comparison of the proposed model with empirical simulations for source data block of $k = 100$ and B_L=100[msec].

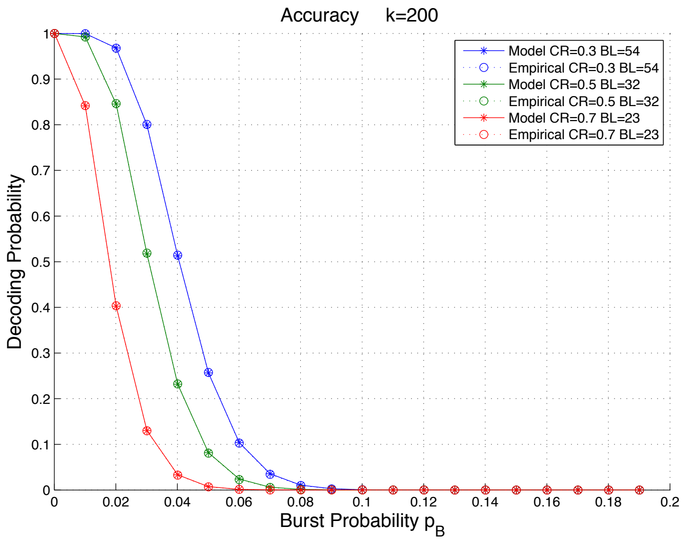

Figure 4.7: Comparison of the proposed model with empirical simulations for source data block of $k = 200$ and B_L=100[msec].

for any burst length, with a level of accuracy comparable to that achieved by means of time-extensive and resources-consuming empirical simulations. The proposed scheme brings the advantage of having a fast execution time, which makes it a really valuable tool for simulation environments. As can be seen from the formulas of the proposed model in section 4.3, the structure of the model consists of similar operations for each of the cases described. Each of these operations consists of multiplications, exponentiations and the calculation of a binomial coefficient.

In order to evaluate the runtime behavior of the proposed model, the number of necessary operations over the number of source symbols k and the burst error length in terms of symbols B_L is empirically computed and given in Fig. 4.8 for a code rate of $CR = 0.5$. The proposed model scales well with a given burst size over any amount k of symbols converging to a maximum of operations at $k = B_L$ symbols. For a given k, the number of operations grows quadratically with B_L.

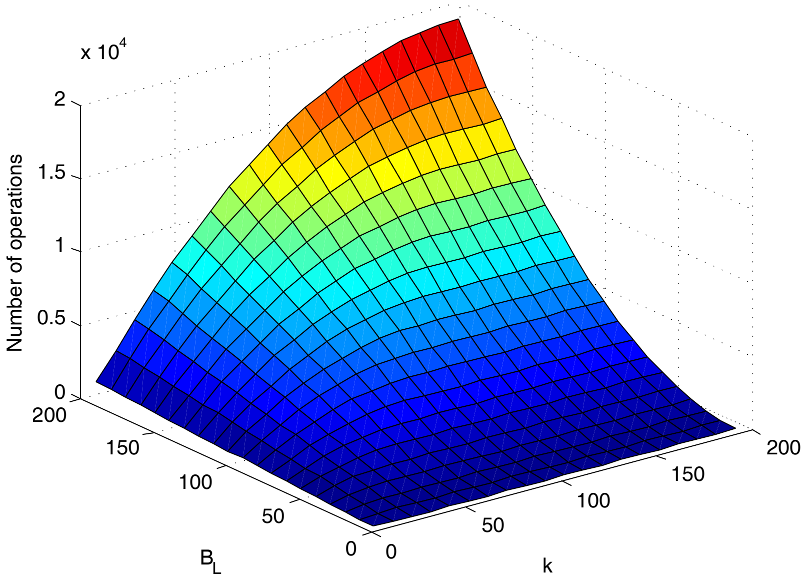

Figure 4.8: Number of operations of the proposed model over symbols k and burst length B_L [4].

4.5 Conclusions

In this chapter, a model able to evaluate the decoding probability of an ideal FEC code, in the presence of REIN impulse noise, has been introduced. The model is able to cope with FEC data impaired by fixed length randomly distributed bursts. The presented results prove the accuracy of the proposed solution in comparison with extensive empirical simulations. The correlation between the model and the empirical simulations is really satisfying and the error signal between the two has the characteristics of random noise without any persisting bias, confirming the accuracy of the designed model. In addition, model benefits have been also proved in terms of runtime performance. In fact, the model can clearly outperform empirical simulations as the latter need a large number of repetitions to achieve statistically relevant results, whereas the proposed model does not require effort in this regard. Therefore, the model can be beneficial whenever empirical simulations are not feasible, e.g. due to time constraints.

CHAPTER 5

WINDOWED PSNR

In the previous chapters, techniques for making reliable the transmission of video flow over lossy prone channels and networks have been introduced as well as some other important aspects of a typical video transmission chain. Among them, a really important aspect that must be duly considered is the video quality that a service or application are able to guarantee. The term *quality* or *quality level* commonly addresses the perception that a final user has of a decoded video flow.

In general, each service and application organizes the video flow and makes use of some specific techniques (i.e. for compression, encoding, etc.) to carefully adapt their parameters to achieve a well defined level of quality while respecting the underlying system constraints. Clearly this strongly depends on the audience of the considered service/application and on the kind of receiver terminals. As a matter of fact, the level of quality to be provided for satisfying mobile users is much lower than the one needed for HDTV applications. Given this, results clear how important is being able to evaluate this "perceived" quality and how useful would be doing it by means of models and methods able to evaluate it coherently with the real human perception. In section 1.6 of this dissertation, the quality evaluation topic has been defined and a classification of the different possibilities currently applied for performing this evaluation has been provided. Quality evaluation methods have been classified following different criteria and the Peak Signal to Noise Ratio (PSNR) has been introduced as the most widely used objective video quality metric.

However, traditional PSNR calculations do not take packet loss into account thus giving an inaccurate representation of the video quality in lossy scenarios. Indeed, transmission packet losses can become the cause of visible destructive distortions, which can result in a progressive quality degradation due to error propagation effects. In this chapter a novel video quality evaluation methodology based on the Y-PSNR is introduced along with simulation results. This technique, called Windowed PSNR (WPSNR), allows to handle the packet loss problem and to evaluate the PSNR of the received sequence.

5.1 Motivation and Goals

Streaming and broadcasting applications are highly demanding in terms of bandwidth resources and they comprise the vast majority of the Internet traffic. Scientists from all over the world, coming from both academic and industrial sectors, have tried to overcome typical challenges in video delivery. However, digital video transmission over wired and wireless packet networks is a relatively new field and many challenges still remain open. One of the hottest and most complex open trend is the optimization of the Quality of Experience (QoE), which measures the application and user oriented quality of video and multimedia services, as it is perceived by the final users [36].

In this context, the real human perception has to be taken into account and this is not an easy task since several free variables come into play. Evaluating how humans effectively "perceive" the quality of a video clip and mapping this perception into a numerical scheme stems from the assumption of a total comprehension and knowledge of the human visual system (HVS). In addition, real users opinions are also strongly tied by not-scientific factors like personal interests, etc. Therefore, many studies have been carried out and their outcomes have been applied as a basis for further researches and improvements in video quality assessment. As an answer to this need, both subjective and objective metrics have emerged. The two classes of methodologies go hand in hand for achieving an universally applicable method which possesses a real significance in terms of human perception and - at the same time - it is easy and rapid to evaluate numerically.

From one side, there are subjective quality evaluation methods based on real viewers quality rating which are performed in structured and controlled environment. From the other, objective approaches measure video quality through mathematical models trying to take into account - in a way or in another - how our perception system works. Nonetheless, objective metrics do not always match the real perceived qual-

ity, and the designer must be aware of possible sub-optimalities introduced by this mismatch.
Within the activity herein described significant steps in the direction of the definition of objective/subjective methods applicable to mobile satellite video transmission are made. Our study considers the PSNR (Peak Signal to Noise Ratio) metric, but stemming from the assumption that human beings are more sensitive to luminance than to chrominance variations, the Y-PSNR (Luminance - Peak signal to Noise Ratio) is herein adopted. This allows on the one side to move towards a more subjective evaluation methodology, accounting for human perception, and on the other to lower computational complexity. In the following we will refer to the specific considered Y-PSNR technique as PSNR.

As application scenario, the video transmission over mobile satellite links is considered. It is well known that this kind of transmission links are very susceptible to packet losses and errors caused by deep fading events lasting up to several seconds. The packet losses cause visible destructive distortions, which can also further degenerate due to error propagation effects and the resulting video quality can be degraded beyond viewer tolerance. In addition, the loss of video packets contributes to the loss of synchronization between audio signal and video frames, as well as the misalignment between the transmitted and the received video sequence and video stream shortening. When the packets comprising the video stream are lost or discarded, the corresponding frames are either partially or fully absent at the decoding end, therefore, at the receiver end the decoded video stream will be shorter than the originally transmitted video stream resulting in misaligned sequences. In addition to the temporal misalignment, due to the lack of data packets at the decoder side, other factors could contribute to the misalignment between the processed video sequence with respect to the unimpaired one.
For the sake of simplicity only video data will be considered in the rest of this chapter.

The solution herein proposed is able to take into account loss of frame alignment and to evaluate the perceived quality of the received video sequence after transmission on a lossy channel. With respect to other methods, our solution allows the use of the PSNR metric, which is notoriously fast and easy to use, in situations in which it cannot be applied. As already introduced in section 1.6.1, the PSNR is a full-reference metric hence it can be used only if both original and reconstructed video sequences are available in full-length. One of our objectives is to overcome effectively this weakness and to enable the use of the PSNR metric in any possible context and to any possible content. In addition, the proposed method allows to estimate the lost frame positions within the received sequence exploiting the original transmitted sequence in the uncompressed domain. Nonetheless, determining the position of lost frames within the uncompressed received sequence is very difficult, since no header information and frame number are present. The strength of the presented Windowed PSNR (WPSNR) method is to allow the sequence alignment recovering without the need of side information or sequence pre-processing but by means of a quality evaluation procedure, resulting in a fast and reliable solution which is fully independent of the specific kind of video sequence addressed and of the codec used.

The alignment between transmitted and received video is recovered by means of a sliding window mechanism. This mechanism follows from the assumption that the PSNR calculated on a couple of peer frames will be higher than the value obtained comparing two non-peer frames. Once the position of lost frames has been detected, losses are recovered with a specific interpolation procedure which enables peak signal to noise ratio evaluation everywhere. For the sake of simplicity, our method recovers the sequence alignment by freezing, for each detected frame loss, the last correctly received frame as many times as needed. This procedure is mandatory in order to enable the PSNR calculation. The literature introduces several possible techniques for recovering the sequence alignment, mostly based on linear interpolation procedures. These techniques are especially reliable from the resulting quality point of view, but as drawbacks they require higher computational complexity without introducing any additional gain to the frame loss position detection. The WPSNR solution is source coding independent since it is applied on the raw format sequence. It is very fast, it presents low computational complexity being based on fast PSNR calculations and it could be applied to any application and content. In addition, the proposed methodology is particularly suitable to the future design and analysis of error protection techniques applied to video transmission.

5.2 Theoretical Background

5.2.1 System Overview

The application scenario addressed in this work is the transmission of video contents over a mobile satellite channel. More in detail, the DVB-SH standard is considered, exploiting its hybrid satellite-terrestrial structure and its properties of ubiquitous IP-based multimedia services. The Digital Video Broadcasting Satellite to Handheld (DVB-SH) has been designed in order to address the need for receiving satellite TV on mobile devices. It presents several advantages with respect to terrestrial mobile TV services. The main advantage is the ability to cover an extended geographical area. Indeed, the hybrid structure of the standard allows a collaborative use of satellite and terrestrial networks, covering both rural and urban areas. In addition, its structure leads to lesser network infrastructure costs if compared with purely terrestrial distribution networks designed to cover the same area. Moreover, the nature of the services provided allows the coverage of difficult to serve areas where establishing a network infrastructure is not feasible or not economically affordable.

The DVB-SH standard has been designed with the intention of extending UHF-based services to frequencies below 3GHz granting a huge coverage area, a reduced network infrastructure cost and the possibility to provide a richer offer in terms of TV channel. The system and waveform specification standards [58]-[59] have been published by ETSI in March-April 2008, while the implementation guidelines [60] have been made available by the DVB workgroup since May 2008. The DVB-SH standard has been developed reusing as much as possible the existing international standards, therefore it inherits large parts from the DVB-S2 [61] standard for the satellite part and from the DVB-H [62] for the terrestrial one.

The mobile channel is characterized by several phenomena which affect signal propagation. The most important ones are refraction/reflection/absortion of the signal which lead to path losses, shadowing and multipath fading for both terrestrial and satellite channels. Therefore, the transmission over such channels is highly subject to packet losses, which in video transmission applications, may determine strong visible disturbing effects. To be more specific, the addressed scenario is characterized by error bursts from a few hundred milliseconds up to ten seconds. The standard envisages different solutions to overcome these issues, but they are not subject of this work, for further references on the topic see [40],[43].

5.2.2 Video Quality Metrics Overview

Due to the migration of video processing from classical analog to current digital techniques, the methods for assessing video quality have been changed accordingly. In analog television services, rating video quality was easy and several simple and well defined metrics were defined and widely used. Among these we can cite the peak signal to noise ratio and the nonlinear distortion of the video signal. These metrics were enough for providing an evaluation on the effective overall quality, but the same is not available for the digital case. In fact, digital processing introduces different impairments in video images, which may have different visible impacts influencing quality degradation. Furthermore, for digital videos an universally valid metric able to give a sufficient insight of the overall video quality does not currently exist. For this reason, video quality evaluation metrics represent a very hot research topic. Several studies have been conducted to determine a universally valid quality metric; a huge variety of video quality metrics are reported in the literature that could be classified following different criteria. A more in deep analysis of this topic has been already provided by the author in section 1.6.

5.2.3 Gilbert Elliott Channel Model

As a channel model the widely used Gilbert-Elliott has been employed. The Gilbert-Elliott model is a simple channel model introduced by Gilbert[63] and Elliott [64]. The Gilbert-Elliott Channel (GEC) model is largely used for the emulation of burst error patterns in transmission channels, hence it is a valid tool for simulating performance of error/loss affected transmission links. In addition, it is a suitable model for evaluating coding efficiency for error correction and detection.

The model is based on a two states discrete-time Markov chain with memory one. A state referred to as the **good** state represents reception of a symbol (or bit or packet) with probability $p_R = 1 - p_{er}$, the second state, referred to as the **bad** state, corresponds to an error/losses occurrence with a probability p_{er}. Herein the common notation is used and the two states are respectively indicated by G(1) and B(0).

Focusing on the current application of the model, the good state denotes a successful reception and the bad state denotes a loss of the actual symbol or packet.

The model can be fully described by a transition probability matrix, in the following indicated by P_T and a vector containing the initial distributions.

P_T (cf. 5.1) contains the probability that each state has either to transit to a new state or to remain in the same state for another time instance.

$$P_T = \begin{bmatrix} P_{11} & P_{10} \\ P_{01} & P_{00} \end{bmatrix} \tag{5.1}$$

As shown in [65]-[66], given the average error rate (p_{er}) and the Average Burst

Error Length ($ABEL$), P_T can be fully determined being $P_{11} = (1 - P_{10})$ and $P_{00} = (1 - P_{01})$, where $P_{01} = 1/ABEL$ and $P_{10} = p_{er}/(ABEL \cdot (1 - p_{er}))$.
The initial distribution vector, indicated by p, is used to initialize the channel and it is found by satisfying the following steady state condition:

$$p \cdot P_T = p \tag{5.2}$$

and for the given transition matrix is equal to $p = (1 - p_{er}, p_{er})$.

5.2.4 Peak Signal to Noise Ratio

The Peak Signal to Noise Ratio definition and mathematical formulas are reported for the sake of completeness.
PSNR is the most used objective quality metric, suitable for both video and still images. It is computationally lightweight, applicable to any content type, source coding independent and easily interpreted using ”standard” quality intervals. PSNR is primarily used in evaluating codec performance, particularly as a comparison method between different video codecs [67]. Even if some studies [68] have shown that PSNR does not strongly correlate with subjective quality measures, it can be considered as a benchmark in all cases.

PSNR is defined as the ratio of the squared useful signal peak over the mean squared error in decibel. More precisely, the PSNR between the i-th frame of the uncompressed original video sequence and the j-th frame of the reconstructed (after the decoding process) video sequence is defined as:

$$\mathrm{PSNR}(i,j) = 10 \log_{10} \frac{(2^P - 1)^2}{\mathrm{MSE}(i,j)} \tag{5.3}$$

where $2^P - 1$ is the peak value that a pixel can take for a P-bit representation, while the MSE is computed as the average quadratic pixel by pixel difference between the original video frame, $f_i(x,y)$, and the decoded video frame, $g_j(x,y)$:

$$\mathrm{MSE}(i,j) = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} [f_i(x,y) - g_j(x,y)]^2 \tag{5.4}$$

where M and N represent the horizontal and vertical resolution respectively.
PSNR can be computed for each frame of the video signal under test, and it can be evaluated on both luminance and chrominance components, as well as on the R,G,B chroma components. The PSNR of the entire sequence is obtained by averaging the sum of frame-by-frame PSNR values over the total number of considered frames.

In this work the PSNR evaluation is performed on the luminance dimension only, bringing about also the additional advantages of lowering the computational complexity. Note that, for evaluating the Y-PSNR the computation should be restricted to the luminance frame only, hence the value used for P is generally equal to 8.

5.3 Windowed Peak Signal-to-Noise Ratio (WPSNR)

The purpose of the proposed Windowed PSNR method is to enable the computation of an objective quality metric in the presence of unknown frame losses in the uncompressed domain. In particular the algorithm solves the problem of detecting the position of lost frames, efficiently recovering the alignment between the original sequence and the decoded loss-affected version.

The quality of the video can experience a drop due to, on the one side the entire frame loss and, on the other, the decrease of quality ascribable to packet losses which lead to partial frame reconstruction. In order to evaluate this drop, the proposed method allows the evaluation of both the PSNR computed on the subset of decompressed frames only and the PSNR computed on the whole sequence, after the alignment recovering procedure. The PSNR of the shorter received sequence is evaluated considering the subset of corresponding frames. The corresponding frames are the couples of "peer" frames between the original sequence (full length version) and the decoded one. In order to detect these corresponding couples of frames a sliding window mechanism, which will be further explained in the following, is used. When considering only these subsets of frames the PSNR value obtained will be indicated as PSNR$_{decoded}$. In addition to that, the presented WPSNR mechanism enables also the possibility to evaluate the PSNR of the whole sequence obtained after the alignment recovery procedure. In this case, the calculated PSNR value is indicated as PSNR$_{aligned}$. In the literature, various methodologies are used for recovering the alignment between video sequences and also between video and audio traces. In particular, focusing only on video sequence alignment, the most widely used techniques are frame duplication and frame interpolation. Another category of methods are based on prediction mechanisms hence they are coding-dependant and for this reason they are behind the area of interest of this dissertation. With the presented method the alignment between the original and the decoded sequences is reached by freezing, for each burst of losses detected, the last correctly received frame. This freezing frame procedure will be indicated in the following as one-way duplication. The sequence under evaluation has been previously compressed, data packets resulting from the encoding process have been progressively encapsulated in the data units of the underlying levels of the considered protocol stack and transmitted over a loss-prone channel with a variable erasure probability and with different average burst error length (ABEL). The losses are applied to transport level packet units and the ABEL is expressed in terms of packets. For emulating the mobile channel behavior characterized by long error burst, the Gilbert Elliott Channel (GEC) Model introduced in section 5.2.3 has been used. The sequence decoded at the receiver side may result in a shorter sequence with respect to the original in case of losses.

Losses and their position within the received video sequence are recovered taking into account the fact that, even if the two sequences have obvious differences due to the effect of the encoding and decoding process and to the effects of packet losses, the PSNR evaluated between a couple of "peer" frames will be higher with respect to the one evaluated between two "non-peer" frames. The innovation of the proposed solution consists in its ability of identifying the skipped frames in the received sequence using a quality indicator like the PSNR. In fact, in general loss detection is performed in the compressed domain thus exploiting the information contained in the data packets header.

The WPSNR algorithm is comprised of three steps:

1. **Alignment recovery**

   In the first step the alignment of the two sequences is recovered. To this end a sliding window mechanism is used: at each iteration a comparison between the j-th frame of the reconstructed video sequence and a number of frames from the original one equal to the number of frames lost during transmission plus one is performed. Once the PSNR values between the j-th frame and all the frames within the window have been calculated, the maximum value is selected and the frame whose index was selected is elected as peer frame to the one under test.

2. **Full sequence reconstruction**

   One-way duplication is performed to obtain a fully reconstructed sequence and overcome the sequence shortening issue.

3. **Decoded and Aligned PSNR evaluation**

   The aligned PSNR evaluation is performed computing the PSNR frame-by-frame between the original video sequence and the fully reconstructed sequence following from the 2^{nd} step and averaging the result over the total number of frames composing the video. In addition, the decoded PSNR is calculated summing up the PSNR value of the peer frames averaged over the total number of frames of the shorter received sequence.

The comparison between the two values thus computed give us a measure of the quality drop caused by compression and lossy-transmission.
Hence, taking into account the definition of PSNR and MSE given in (5.3) and (5.4) respectively, we can formulate the algorithm as detailed in the following.
At the generic k-th iteration, the algorithm evaluates:

1. *The sliding window size*
   Indicating with L_{or} the number of frames which compose the original sequence, with L_{rec} the number of frames of the decoded video sequence and with $loss_{count}(k-1)$ the number of frame losses detected up to the k-th iteration, the sliding window size is:

$$L_W(k) = L_{or} - L_{rec} - loss_{count}(k-1) + 1 \tag{5.5}$$

2. *The PSNR* between the k-th frame of the received sequence and $L_W(k)$ frames of the original sequence. The original frames are selected by varying the h index in the range $h = [\widehat{h}_{(k-1)} + 1 : \widehat{h}_{(k-1)} + L_W]$, where $\widehat{h}_{(k-1)}$ identifies the frame in the original sequence with the maximum PSNR value inside the sliding window of the $(k-1)$-th iteration

$$\text{PSNR}(h,k) = 10 \log \frac{(2^P - 1)^2}{\text{MSE}(h,k)} \tag{5.6}$$

3. *The "peer" frame*

$$\widehat{h}_k = arg \max_h \text{PSNR}(h,k) \tag{5.7}$$

4. *The presence of losses* and updates the number of detected lost frames

$$\begin{aligned} l(k) &= \widehat{h}_k - (\widehat{h}_{(k-1)} + 1) & (5.8)\\ loss_{count}(k) &= l(k-1) + l(k) & (5.9) \end{aligned}$$

5. Performs the *one-way duplication*

6. *Decoded and aligned PSNR*

$$\begin{aligned} \mathrm{PSNR}_{aligned}(k) &= \mathrm{PSNR}_{aligned}(k-1) + \mathrm{PSNR}(\widehat{h}_k, k) \\ &+ \sum_{n=1}^{l} \mathrm{PSNR}(\widehat{h}_k - n, k-1) \end{aligned} \tag{5.10}$$

$$\mathrm{PSNR}_{decoded}(k) = \mathrm{PSNR}_{decoded}(k-1) + \mathrm{PSNR}(\widehat{h}_k, k) \tag{5.11}$$

The total number of iterations is equal to the length of the received video sequence expressed in number of frames. Once the last iteration has been performed, the resulting $\mathrm{PSNR}_{aligned}$ value must be divided over the total number of frames of the original video sequence, while the resulting $\mathrm{PSNR}_{decoded}$ must be divided over the number of frame actually received.

In order to support the provided algorithmic description, the key aspects of the algorithm are clarified by using a graphical example.
In Fig. 5.1 a snapshot of the initial situation is given. On the upper part, the shorter, loss-affected, received sequence is shown. The sequence in the bottom part represents the one originally transmitted. In order to simplify the comprehension of the ”peer frames” concept, in the following figures, frames belonging to the two sequence but of the same color are the right corresponding frames. As highlighted above, frames which compose the received sequence may be affected by other negative factors (s.a. compression effect, packet losses which do not lead to entire frame losses) but this aspect is not addressed in the pictures for sake of simplicity, although it has been taken into consideration while defining the method. In the figure, the lengths of the two video sequences are indicated, using the same terminology used above.

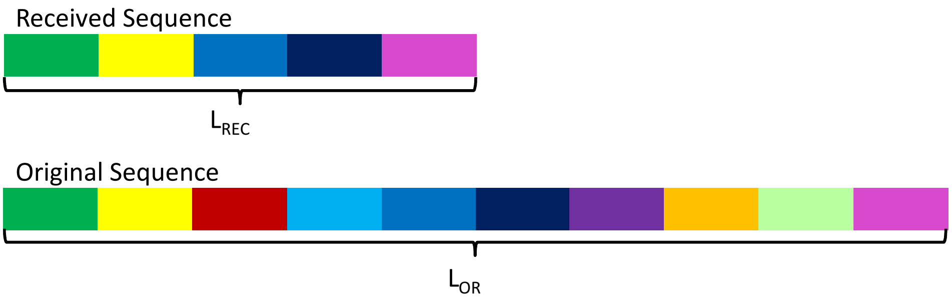

Figure 5.1: Transmitted and Received Video sequences with their own lengths.

In Fig. 5.2, an exemplary iteration is depicted. More in detail, the considered iteration cover the exemplary case of no losses within the current sliding window. It is easy to note that each iteration starts sliding a window of size 1 over the received video sequence frames, for this reason the total number of iterations to be performed is equal to the length of this sequence (L_{REC}).

The PSNR values between the frame within the sliding window of size 1 and the L_W frames within the sliding window of size L_W are calculated as described in step #2 of the WPSNR algorithm. The right corresponding "peer" frame is found by solving Eq. 5.7. Since no frame losses have occurred, in this case the elected frame is the first of the sliding window (the green one). Sliding window size and loss counter are updated, even if their values in this case rest the same. The $PSNR_{aligned}$ and the $PSNR_{decoded}$ values are calculated and summed up to corresponding values of preceding iterations, accordingly to Eq. 5.10-5.11.

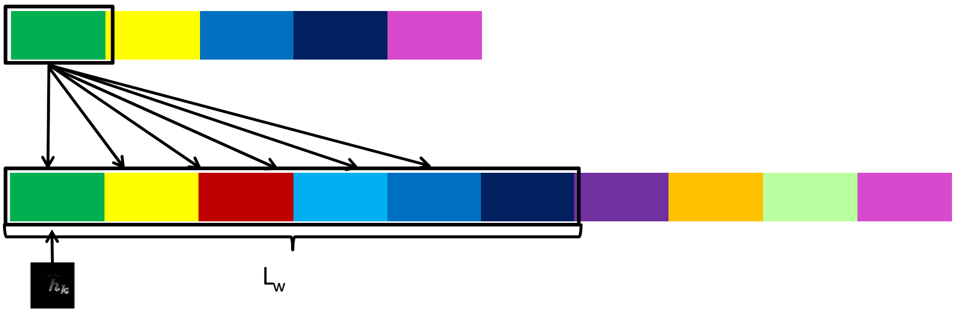

Figure 5.2: Exemplary iteration of the windowed PSNR algorithm. Case without losses within the sliding window. Election of the corresponding "peer" frame for the currently considered received frame.

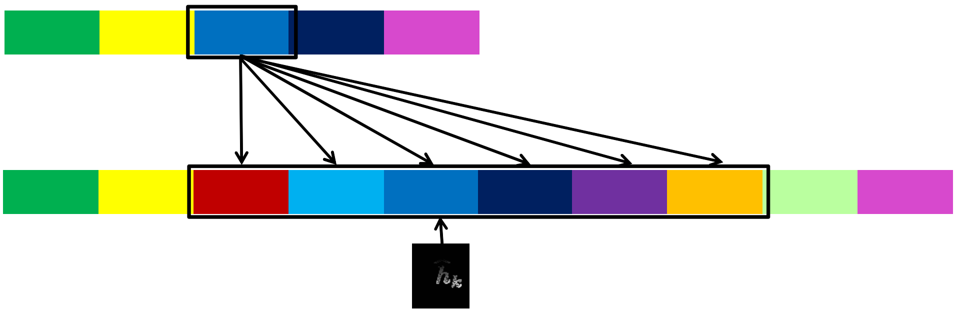

Figure 5.3: Exemplary iteration of the windowed PSNR algorithm. Case with losses within the sliding window. Election of the corresponding "peer" frame for the currently considered received frame, sliding window size updates.

In Fig. 5.3 an exemplary iteration in which one or more missing frames are detected is shown. The sliding window of size 1 is now on the blue frame of the original video sequence. The sliding window - which slides over the original uncompressed sequence - begins with the red frame.

The PSNR values between the blue original frame and the L_W frames within the sliding window are computed. The index which identifies the right corresponding peer frame is found and corresponds to the blue frame of the original video flow. As can be seen, this frame is not anymore the first within the current sliding window, but the third, hence two intermediate frames are missing. The sliding window size must be opportunely down-scaled and the loss counter updated. The $PSNR_{aligned}$ and $PSNR_{decoded}$ are calculated and preceding values updated.

In Fig. 5.4 is shown how the sliding windows are shifted after the detection of one or more losses. In addition note the updated size of the sliding window which moves over the transmitted sequence.

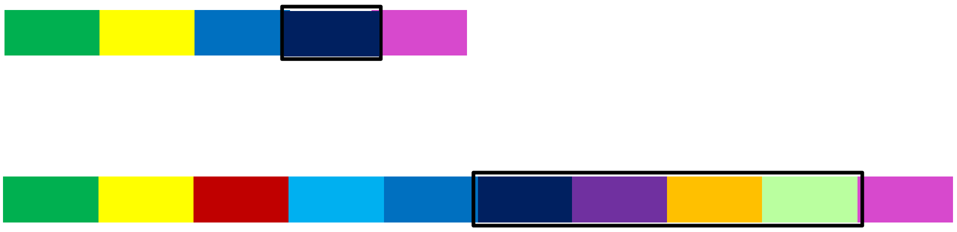

Figure 5.4: Situation after the detection of one or more missing frames.

5.4 Simulation Set and Results

The video clip used in the evaluation has been downloaded from the database of test clips provided by the VQEG (Video Quality Expert Group)[1]. The downloaded sequence is composed by YCrCb images, but for the WPSNR elaboration a pre-processing for reconstructing a raw video stream has been performed. The sequence lasts 10 seconds, with a frame rate of 15 frames per second and 352x288 resolution. In order to compress the video flow and to encapsulate the resulting encoded data units in RTP packets, the ffmpeg tool [2] has been used. The RTP packets have then been dumped by means of the rtpdump tool [3]. The generated RTP video sequence is then transmitted over a Gilbert Elliot Channel (GEC), with different erasure probability and different Average Burst Error Length (ABEL). For simplicity, each data unit has

[1] http://www.its.bldrdoc.gov/vqeg/vqeg-home.aspx

[2] http://ffmpeg.org

[3] http://www.cs.columbia.edu/irt/software/rtptools

been encapsulated into one RTP packet, hence no payload segmentation has been used. The resulting RTP flow presents a variable number of packets in function of the compression ratio applied. The ABEL imposed on the channel is expressed in number of consecutive lost RTP packets. It is worth noticing that the payload length of the packets is different for different compression ratios. An RTP packet loss may result in different kinds of impairments, due to the different payload types carried in the RTP packet and to the error propagation effect related to the coding dependency structure applied during compressing. As already mentioned above, the WPSNR method is coding independent and it must be applied in the uncompressed domain, therefore the position of losses within the sequence is found without any previous knowledge on the sequence under analysis. In order to evaluate the performance of the proposed method a simulation campaign has been performed.

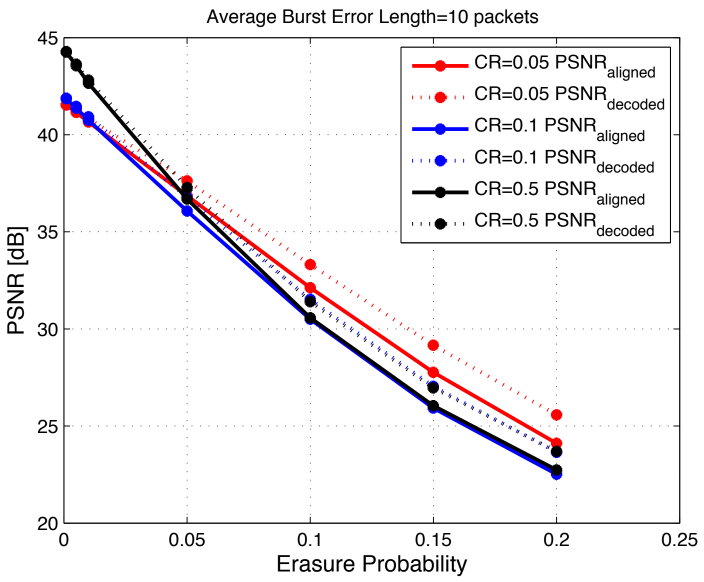

Figure 5.5: Comparison between PSNR values obtained by means of the sequence alignment procedure and PSNR of the received frames with ABEL=10 [RTP Packets] for different Compression Ratios

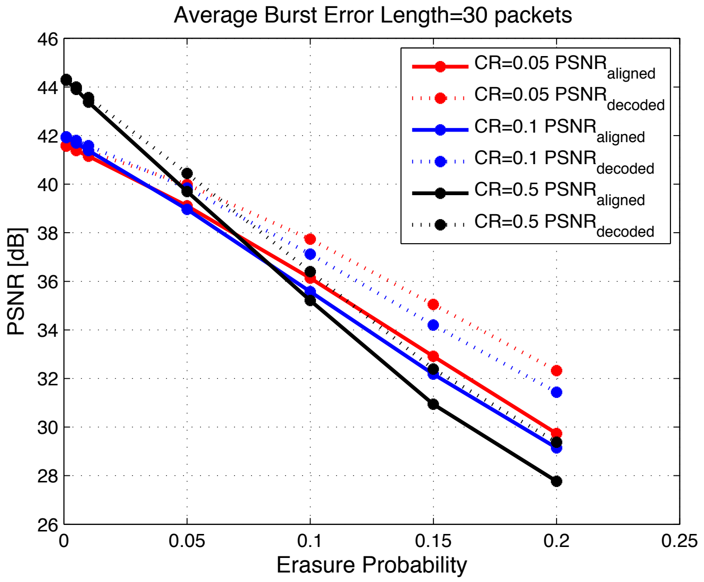

Figure 5.6: Comparison between PSNR values obtained by means of the sequence alignment procedure and PSNR of the received frames with ABEL=30 [RTP Packets] for different Compression Ratios

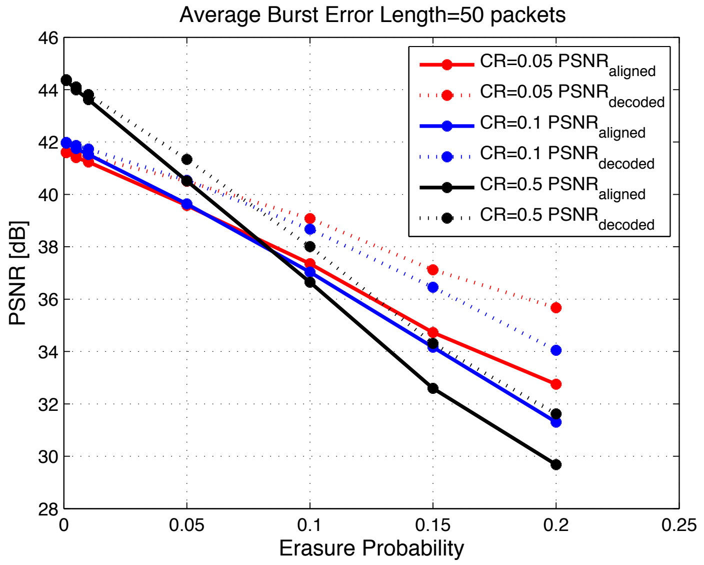

Figure 5.7: Comparison between PSNR values obtained by means of the sequence alignment procedure and PSNR of the received frames with ABEL=50 [RTP Packets] for different Compression Ratios

In Fig. 5.5-5.7 the performance of the WPSNR method is reported. For easy understanding of the presented results, it is fundamental to remember that no error concealment/recovery techniques have been applied and no channel coding technique has been used. The results are expressed in terms of PSNR with respect to the Erasure Probability of the channel for different ABEL values. The sequence is encoded with different compression ratios, evaluated comparing the size in bits of the sequence post compression over the size in bits of the original sequence. The impact of the compression on the overall quality can be deduced looking at the PSNR values for different compression ratio corresponding to an erasure rate equal to 0. The results show that the quality evaluated on sequences decreases as the erasure probability increases both for the sequences reconstructed by means of a frame duplication procedure and for the sequences actually received. As explained in section 5.3, first loss positions are detect by means of the sliding window mechanism, then the sequence alignment is obtained performing a one-way frame duplication. We compare the PSNR value obtained considering the whole duplicated sequence $PSNR_{aligned}$ with the one evaluated only on the subset of frame actually received $PSNR_{decoded}$, skipping the frame of the original sequence that have no correspondence with the received shortened sequence. The results clearly shown a gap between these two values, highlighting the potentialities of the proposed method, whose objective is mainly detecting the position of lost frame within an uncompressed, loss-affected video sequence without having any previous knowledge. As can be deducted from the presented results, even if there is a quality degradation ascribable to RTP packet losses which does not result in the loss of the entire frame, $PSNR_{decoded}$ values are always higher than $PSNR_{aligned}$, this is due to the fact that the alignment is recovered by freezing the last correctly received frame belonging to the received loss-affected sequence. As a consequence, the duplicated frames suffer from the drop of quality due to RTP packet losses. However, the reliability of the $PSNR_{decoded}$ value is strongly related to the effectiveness of the loss position estimation procedure. In fact, the $PSNR_{decoded}$ is evaluated on the subset of frames actually received thus needing the knowledge of all received frames' peers. The performance of the proposed method improves with the length of the ABEL considered. This is due to the fact that the longer the error burst is, the easier it is to determine its position within the sequence. Therefore the solution herein proposed is particularly suitable for the application scenario under consideration, typically affected by long error bursts. In addition, the relationship between Frame Error Rate (FER) and Packet Error Rate

(PER) is linear and independent from both compression ratio and ABEL imposed.

5.5 Conclusions

In this chapter a novel method that allows at the same time to recover the alignment of a loss-affected video sequence with its original unimpaired one and to calculate the PSNR values of the received short and full-lengths video sequences has been introduced. The problem of evaluating the video quality in case of frame dropping is difficult to tackle. As highlighted in the first part of this chapter, the fame dropping phenomenon is strictly related to other bothersome phenomena which can afflict the quality perceived by final users. In the preceding chapters of this dissertation, we have seen some possibilities for improving the robustness of a video flow while broadcasting or streaming it. Here, as a natural continuation of the presented activities, an attempt has been done in order to provide a tool for self-made evaluation of video quality in case of losses and in the uncompressed domain. For providing an effective quality assessment method also in the presence of bursts of errors, the WPSNR algorithm, which is able to detect the position of lost frames in a loss-affected uncompressed received sequence, has been introduced. The simulation results shown the effectiveness of the proposed method and its capacity of locating couples of corresponding frames without the need of additional information. Indeed, the algorithm allows the recovery of the video sequence alignment, performed by means of the frame freezing technique and the evaluation of the quality drop due to RTP packet losses resulting in completely loss of the frame or introduced quality impairments. The relationship between the FER and the PER has also been computed, showing a complete independency from both compression ratios and ABEL. The aim of this study has been mainly the creation of a solid framework for testing the performance of packet forward error correction mechanisms applied at the upper layers of the protocol stack. In fact losses position detection previa and post application of one or more techniques for error correction/concealment is an useful possibility that gives an insight into the effective robustness enhancement achieved by their applications. The presented activity is still ongoing. The comparison of the proposed methodology with other well know quality metrics is a work in progress. In addition, as it can be gathered from an analysis of the proposed algorithm, the WPSNR is an "optimum" method from the computational point of view. In fact, the natural approach for detecting the presence and position of one or more losses within a received uncompressed video sequence is to compare each of its frames with all the frames of the original sequence. This means that the number of operations to be performed is much higher with respect to the proposed WPSNR. Defining a priori the maximum number of losses which have occurred during the transmission and re-scaling the sliding window size every time that a dropped frame has been found, minimize the number of iterations optimizing the overall procedure.

APPENDIX A

VIDEO BASICS: PARAMETERS, NETWORKS AND PROTOCOLS

In this appendix a description of the most important video parameters and characteristics is provided along with an overview of the basic techniques used for video communication over different communication networks and video-oriented Internet protocols.

A.1 Video Basics

The word **video** is used to address the technology of electronically capturing, recording, processing, storing, transmitting, and reconstructing a sequence of still images representing scenes in motion. Digital video is a representation of a natural or real-world visual scene, sampled spatially and temporally. A scene is sampled to produce a frame, which represents the complete visual scene at that point in time, or a field, which typically consists of odd or even numbered lines of spatial samples. Sampling is repeated with constant intervals of 1/25 or 1/30 seconds to produce a moving video signal. In order to represent a scene in colors, up to three components or set of samples are required. A digital video must have a binary format consisting of $0s$ and $1s$. Unlike still images, it is dynamic and its visual content evolves over the time and contains moving objects. This makes video more coherent with our real world which is continuously moving and changing its status. Besides the positive and exciting aspects of video, it is important to realize that a video is a multidimensional signal, function of three dimensions (two spatial and one temporal), therefore handling digital video is not an easy task since significant bandwidth and high computational and memory resources are required.

A.1.1 Video Characteristics

Video can be classified and categorized according to different criteria:

- **Frame rate**

  The *frame rate* represents the number of still images displayed in one second of video. It ranges from 6-8 up to 120 frame per second (fps) or even more. The frame rate should be high enough, otherwise the displayed video will appear to flicker. The minimum frame rate to achieve the illusion of a moving image is about fifteen frames per second. The two standards currently in use are the the PAL/SECAM and the NTSC which specify frame rates of 25 and 29.97 fps respectively.

- **Refresh Rate**

  The *refresh rate* represents the number of times in a second that a display hardware draws the data. It should be greater or equal to 50 fps.

- **Sampling**

  A digital video is spatially and temporally continuous. This means that representing visual scenes in digital form involves sampling scenes spatially, usually on a rectangular grid, and temporally, as a series of still frames or components of frames (fields) sampled at regular intervals in time. Each spatio-temporal sample, a picture element or pixel, is represented as one or more numbers that describe its brightness or luminance and its color. The number of sampling points used influences the visual quality of the image, the higher is the number of samples used the higher is the resolution. A higher temporal sampling rate gives smoother motion in the video but it requires more samples to be captured and stored.

- **Scanning**

  Two main kinds of video sampling scanning exist: progressive and interlaced. In the progressive scan, each frame is sequentially scanned line by line. In the interlaced scan, two fields are used to create a frame. One field contains all the odd lines in the image while the other contains all the even ones. Video flows captured in one of the two formats (Progressive or Interlaced) must be converted in order to be displayed in the other one.

- **Color spaces**

  A monochrome image requires just one number to indicate the brightness or luminance (Y) of each spatial sample. Color images, on the other hand, require at least three numbers per pixel position to accurately represent colors. The method chosen to represent both brightness and color is described as a *color space*. The most common color spaces are the RGB (Red, Green, Blue) and the YCrCb (Luminance, Red Chrominance, Blue Chrominance).
  In the RGB color space, a color image sample is represented with three numbers that indicate the relative proportions of red, green and blue, whose combination can create any color. The RGB color space is well suited to capture and display of color images. Capturing an RGB image involves filtering out the red, green and blue components of the scene and capturing each with a separate sensor array. Color displays show RGB images by separately illuminating the red, green and blue component of each pixel according to the intensity of each component. The three colors are equally important and so are stored at the same resolution, moreover color images can be represented more efficiently by separating the luminance from the color information and by using different resolutions for the two. In the YCrCb color space, the luminance component (Y) is calculated as a weighted average of R, G and B, while the color components (Cr,Cg and Cb) are calculated by subtracting the mean luminance value of each sample from the color intensity. Cr+Cb+Cg is a constant and so only two of the three chrominance components (generally Cr and Cb) need to be stored or transmitted. In addition, Cr and Cb may be represented with a lower resolution than Y because the Human Visual System (HVS) is less sensitive to color than luminance variations. In this way, the amount of data required to represent the chrominance components is reduced without having an obvious effect on visual quality. To the casual observer, there is no significant difference between RGB and YCrCb images with reduced chrominance resolution.

- **Aspect Ratio**

  The term *aspect ratio* describes the proportional relationship between the width and the height of a video frame. Generally, it is expressed as two numbers separated by a colon. For an x:y aspect ratio, x represents the number of equal length units in which the frame width is divided, while y represents the number of units of the vertical dimension. The most common video aspect ratios are 4:3 (1.33:1) and 16:9 (1.77:1). Other cinema and video aspect ratios exist, but they are used infrequently.

- **Video Formats**

  A wide variety of video frame formats exist. A list of the most common is reported in Table A.1, with associated luminance resolutions.

Format	Luminance Resolution
sub-QCIF	128x96
Quarter CIF (QCIF)	176x144
CIF	352x288
4 CIF	704x756

Table A.1: Video Frame Formats [11].

The video format choice is function of the application and of the available storage or transmission capacity. For example, 4CIF is appropriate for standard-definition television and DVD-video; CIF and QCIF are popular for video conferencing applications; QCIF or SQCIF are appropriate for mobile multimedia applications where the display resolution and the bit-rate are limited.

Parameters	30 fps	25 fps
Fields per second	60	50
Lines per complete frame	525	625
Luminance samples per line	858	864
Chrominance samples per line	429	432
Bits per sample	8	8
Total bit rate	216 Mbps	216 Mbps
Active lines per frame	480	576

Table A.2: ITU-R BT.601-5 Parameters [11].

The standard video format, widely knows as Standard Television, is the commonly used for digital videos and it has been defined by the International Telecommunication Union (ITU) in the ITU-R Recommendation BT.601-5 [69]. In Table A.2 its parameters are listed. The luminance component of the video signal is sampled at 13.5 MHz and the chrominance at 6.75 MHz to produce a 4:2:2 Y:Cr:Cb component signal. The parameters of the sampled digital signal depend on the video frame rate. The higher frame rate of NTSC is compensated for by a lower spatial resolution so that the total bit rate is the same in each case. More recently, other formats have emerged focusing on High Definition (HD). In Table A.3 a list of the HD formats currently in use is provided.

Format	Scanning	Width	Height	Frames/fields per sec
720	progressive	1280	720	25 frames
1080i	interlaced	1920	1080	50 fields
1080p	progressive	1920	1080	25 frames

Table A.3: HD Display Formats [11].

A.2 Video communication networks overview

A variety of communication networks has proliferated over the past few decades. All of them have the aim to assure a good service in terms of available bandwidth while minimizing infrastructure costs. In order to do this, the research efforts have been focused mainly in finding ways of reusing existing communication system infrastructures and several solutions have been proposed and implemented. For example, the hybrid of fiber optics and coaxial cable used in the cable television system has been adapted for data communications. Moreover, the traditional use of air as a conduit in wireless systems (radio and television, cellular telephony, etc.) has recently been extended to accommodate high-bandwidth data communications. Nevertheless, even if several efforts have been made for reusing existing infrastructure, other applications have driven the development of new and more powerful communication backbones. Nowadays, the general design philosophy is to assure very high-bandwidth communication to network backbones and to exploit existing infrastructures to connect individual users. As introduced in [9], this open the local distribution or ”last mile” problem, for which various solutions have been proposed by the cable television, telephone, and wireless industries.

A.2.1 Hybrid Fiber-Coax Networks

The Hybrid Fiber-Coax (HFC) networks are a mixture of fiber optic and coaxial cable. Coaxial cables are used to connect users' home to a central point. The central point is then connected to a head end with optic fiber. Communication networks deployed over existing cable television systems must accommodate both data communications and television broadcasting, bandwidth resources must be shared among all customers. In real scenarios, existing cable television systems do not guarantee data communication rates above 700 kbps. The communication rate is not sufficient to handle MPEG-2 video streams.

A.2.2 Digital Subscriber Loop

The Digital Subscriber Loop (DSL) is a communication standard which has been proposed by the telephone industry in order to build up an infrastructure able to exploit copper twisted wiring present in every home. In fact, it has been introduced with the aim of reusing the Public Switched Telephone Network (PSTN) for communication of data. It is based on an efficient modulation scheme that exploits the copper wires for data exchange. The voice communications are limited to 4 kHz and the required filtering is performed to the end user premises. Instead, the data signal is switched to avoid the filter. Fundamental bandwidth limitations are consequently due to the physical properties of the copper twisted pair in the local loop. A new implementation, known as Asynchronous DSL (ADSL) [70] has taken a few steps in fixing this aspect, introducing the Discrete Multi-Tone (DTM) modulation. The basic idea of DMT is to split the available bandwidth into a large number of sub-channels. DMT is able to allocate data so that the throughput of every single sub-channel is maximized. If some sub-channel can not carry any data, it can be turned off and the use of available bandwidth is optimized. This implementation provides higher bandwidth for downstream than upstream, efficiently exploiting the available bandwidth [71]. For relatively short distances, DSL can provide communication at rates that exceed 8 Mbps. For instance, some ADSL standards [72]-[73] allow for data rates of 8 Mbps downstream and 1 Mbps upstream. Nevertheless, most practical scenarios demand longer transmission lengths and the bandwidth provided by DSL decreases rapidly as the transmission distance increases. For this reason telephone companies can guarantee all users data communications over DSL at rates much lower then the ones requested for video communications [9].

A.2.3 Wireless Networks

Wireless networks have a really ancient story, but it only with Guglielmo Marconi, at the beginning of the 19-th century, that for the first modern wireless communication has taken place. Originally, wireless networks were designed for paging and real-time speech communications and were analog systems. Recently, many efforts have been undertaken to accommodate also data communications and the system is now fully digital. A new generation of cellular standards has appeared approximately every ten years since the first generation (1G) system, based on analog standards, was introduced in 1981/1982. Each generation is characterized by new frequency bands, higher data rates and non backwards compatible transmission technology. The second generation (2G) cellular networks were commercially launched with the GSM standard [74] in 1991. 2G introduced data services for mobile, starting with SMS text messages and 2G networks are still used in many parts of the world. The third generation (3G) is a set of standards used for mobile devices and mobile telecommunication services and networks that comply with the International Mobile Telecommunications-2000 (IMT-2000) specifications by the International Telecommunication Union. 3G finds application in wireless voice telephony, mobile Internet access, fixed wireless Internet access, video calls and mobile TV. The most known 3G standards are the UMTS and the CDMA2000. Both these systems and radio interfaces are based on spread spectrum radio transmission technology. The third generation has enabled new and powerful applications, such as Mobile TV, Video on Demand (VoD), Video Conferences, Global Positioning System (GPS) and many others. Both 3GPP [1] and 3GPP2 [2]are working on potential extensions based on an all-IP network infrastructure and that use advanced wireless technologies such as MIMO. The 4G is the fourth generation of cell phone mobile communications standards. 4G systems provide mobile ultra-broadband Internet access, for example to laptops with USB wireless modems, to smart phones, and to other mobile devices. Conceivable applications include mobile web access, IP telephony, gaming services, high-definition mobile TV, video conferencing and 3D television. Two 4G candidate systems are commercially deployed: the Mobile WiMAX standard (at first in South Korea in 2006), and the first-release of the Long Term Evolution [3] (LTE) standard (in Scandinavia since 2009). The larger bandwidth and significant increase

[1] http://www.3gpp.org/

[2] http://www.3gpp2.org/

[3] http://www.3gpp.org/Technologies/Keywords-Acronyms/LTE-Advanced

in data rates supported by the various standards in IMT-2000 will facilitate video communication over wireless networks. Moreover, the packet switched channel data connection option provided by the various standards in IMT-2000 will allow for the implementation of many of the methods and protocols used for real-time IP networks for video communication over wireless networks (e.g. RTP, RTSP, SIP protocols). Another form of wireless networks is provided by satellite communications. Video broadcasting over satellites has been conducted for many years. Both analog and digital video broadcasting have been used over satellite networks. More recent efforts have attempted to use satellites for real-time video communications. Limited success of this endeavor is due to the large number of satellites that are required to be launched into low orbit to reduce the communication delay.

A.2.4 Fiber Optics

Optical fiber refers to the medium and the technology associated with the transmission of information as light pulses along a flexible, transparent fiber made of glass (silica) or plastic, slightly thicker than a human hair, which functions as a waveguide. Optical fibers are widely used in fiber-optic communications, enabling transmission over longer distances and at really high data rates. Signals travel along the optical fibers with few losses and they are also immune to electromagnetic interference. There are two main methods provided by the telephone industry for local distribution using fiber optics: fiber to the curb (FTTC) and fiber to the home (FTTH). FTTC requires the installation of optical fibers from the end office to central locations such as residential neighborhoods. An even more ambitious design is provided by FTTH which envisages the deployment of optical fiber lines directly to customer's home. Consequently, really high rates can be accommodated, suitable for virtually any multimedia communication application desired. The prohibitive factor in FTTH is the cost, since installing fiber optics into every home is a very expensive task.

A.3 Internet Protocol Networks

The Internet is really powerful and different with respect to the networks discussed so far since it allows for communication across various networks having different physical media and lower layer protocols. This is made possible because of the abstraction of lower layer protocols by the Internet Protocol (IP) common network protocol. IP is the widely used network protocol. The current version is the number 4 (IPv4) and it is characterized by 20-bytes fixed length header plus an optional variable-length one. Popularity of the Internet and forecasts of its future applications and the need to accommodate a larger number of network nodes has led to the emergence of a new version - version 6 (IPv6) - characterized by a longer header field for addresses (16-bytes against 4-bytes of IPv4) and a more flexible header. An illustration of the protocol stack used for video communication over the Internet is depicted in Fig. A.3.

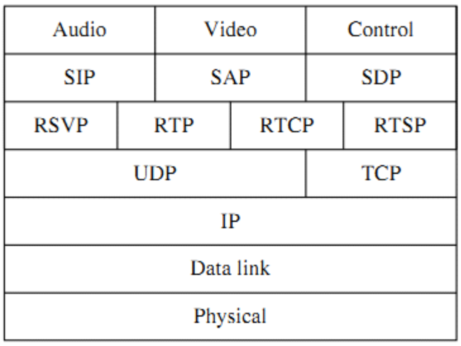

Figure A.1: IP Protocol Stack for Video Communications [9].

As shown in Fig. A.3, over IP there are two transport layer protocols: the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). The main difference between the two is that the first is connection oriented, while the second is connection less. That's why, UDP is typically used for real-time applications such as audio and video communications that require prompt delivery rather than accurate delivery and flow control, while TCP is used for applications which require precise delivery of the contents, such as remote login, electronic mail and file transfer.

A.3.1 Real-Time Transport Protocol

The Real-Time Transport Protocol (RTP), along with its associated profiles and payload formats, can be considered as the key standard for audio/video transport over IP. It provides services such as timing recovery, media synchronization, loss detection and correction. Initially designed for use in multicast conferences, it has proven useful for a range of other applications and in both wired and cellular telephony. The protocol has been demonstrated to scale from point-to-point use to multicast sessions with thousands of users, and from low-bandwidth cellular telephony applications to the delivery of uncompressed High-Definition Television (HDTV) signals at gigabit rates. In addition, it provides end-to-end network transport functions independently of the underlying network or transport protocols. RTP was developed by the Audio/Video Transport working group of the IETF and has been adopted by the ITU as part of its H.323 series of recommendations, and by various other standards organizations. Its first version was completed in January 1996 [75]. In Fig. A.2 a typical RTP data packet with all its fields is shown.

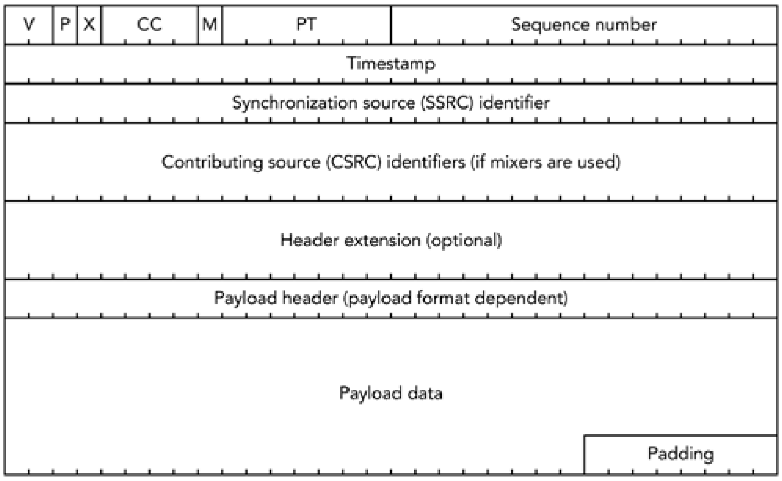
V
P
X
CC
M
PT
Sequence number
Timestamp
Synchronization source (SSRC) identifier
Contributing source (CSRC) identifiers (if mixers are used)
Header extension (optional)
Payload header (payload format dependent)
Payload data
Padding

Figure A.2: RTP Data Packet [10].

The RTP protocol supports the use of intermediate system relays known as translators and mixers. Translators convert each incoming data stream from different sources separately. Mixers combine the incoming data streams from different sources to form a single stream. An example of a mixer is used to re-synchronize an incoming audio or video packet stream from high-speed networks to a lower bandwidth packet stream intended for low-speed networks.

The quality of real-time multimedia transmission in noisy environments is very poor due to high packet loss rates. This problem can be faced by using generic FEC codes (e.g., parity, Reed-Solomon, Hamming codes, etc.) to compensate for packet loss. The payload of a FEC packet provides parity blocks obtained by exclusive-or based operations on the payloads and some header fields of several RTP media packets. The FEC packets and media packets are then encapsulated and sent as separate RTP streams. Since video flows are transmitted in compressed form, a payload specific header is needed. This header is defined by the RTP payload format specification in use, and provides an adaptation layer between RTP and the codec output. RTP has been designed to support a wide range of multimedia formats (such as H.264, MPEG-4, MJPEG, MPEG, etc.) and to allow the introduction of new ones without standard modifications. This is achieved by defining profiles and associated payload formats for each class of applications. The profiles define the codecs used to encode the payload data and their mapping to payload format codes. Each profile is accompanied by several payload format specifications, each of which describes the transport of a particular encoded data. Some of the audio payload formats include: G.711, G.723, G.726, G.729, GSM, QCELP, MP3, DTMF etc., and some of the video payload formats include: H.261, H.263, H.264, MPEG-4 [76]-[77]-[78]-[79].

A.3.2 Real Time Control Protocol

The Real Time Control Protocol (RTCP) works along with the RTP protocol and has the aim of monitoring the QoS and the data delivery and of providing minimal control and identification capability over unicast and multicast services independent of the underlying network or transport protocols. Its main function is to provide feedbacks on the quality of data distribution, which are useful for flow and congestion control. Therefore, it is used for the transmission of a persistent source identifier to monitor the participants and associate related multiple data streams from a particular participants.

An RTCP implementation has three parts: the packet formats, the timing rules, and the participant database.

The packets format can be of five different types, all of them defined in the protocol specification. The five standard packet types are: Receiver Report(RR), Sender Report (SR), Source Description (SDES), Membership Management (BYE) and Application-defined (APP). They all have a common 4-bytes header, followed by packet data and optional padding fields. RTCP packets are never transported individually; instead they are always grouped together for transmission - accordingly to well defined rules- forming compound packets. Each compound packet is encapsulated in a single lower layer packet for transport and sent periodically.

Each implementation is expected to maintain a participant database, based on the information collected from the RTCP packets it receives. This database is used to fill out the reception report packets to perform lip synchronization between received audio and video streams and to maintain source description information. Each RTP session is identified by a network address and a pair of ports: one for RTP data and one for RTCP data. All participants in a session should send compound RTCP packets and, in turn, will receive the compound RTCP packets sent by all other participants.

In conclusion, the peer-to-peer nature of RTCP gives each participant in a session knowledge of all other participants: their presence, reception quality and personal details such as name, e-mail address, location, and phone number [10].

A.3.3 Real-Time Transport Streaming Protocol

The Real-Time Transport Streaming Protocol is an application layer protocol acting as a "remote control" of multimedia communications systems. It is intended for the control of channels and mechanisms used for multiple synchronized data delivery sessions from stored and live sources such as audio and video streams between media servers and clients. The RTSP protocol relies on a presentation description - for which it uses the Session Description Protocol- to define the set of streams that it controls. These controls support for the following basic operations:

- retrieval of media from a media server,
- invitation of a media server to a conference,
- addition of media to an existing presentation.

Control requests and responses using RTSP may be sent over TCP or UDP. The order of arrival of the requests is critical. A retransmission mechanism is required in case any requests are lost. The use of UDP is thus limited and may cause severe problems. Another problem in the use of RTSP is the absence of a mechanism for system recovery. Thus, RTSP implementation requires some other fail-safe method or session control option.

APPENDIX B

VIDEO COMPRESSION ESSENTIALS

B.1 Introduction to Video compression

Video communications almost always rely on compressed video streams. In fact, transmission of raw (uncompressed) video streams is impractical due to the excessive amount of bandwidth needed. Moreover, computer processing and memory limitations often impose serious constraints on transmission rates. Representation of video streams in compressed form is therefore required for efficient video communication systems.

The terms "Video Compression" or "Video Encoding" usually refer to the procedure of reducing the amount of data needed for representing a digital video source for transmission and storage. The video compression is done directly at the source side of the digital video content. On the opposite side, the complementary operation is performed and it is named "Decompression" or "Decoding", referring to the operation which recovers the original digital video signal starting from a compressed representation, prior to display. Data compression - as a general concept - can be of two different types:

- **Lossless compression**. Refers to a class of algorithm for data compression that works in a way such that a perfect reconstruction of the original data can be obtained from the compressed bit-stream. It is mainly used when the fidelity to the original is fundamental (i.e. text document, executable programs, etc.).
- **Lossy compression**. Refers to data encoding methods that compresses data by discarding definitively some unnecessary information. The procedure aims

> to minimize the amount of data that needs to be held, handled, and/or transmitted. It is most commonly used to compress multimedia data (audio, video, and still images), especially in video communication applications (s.a. streaming media, internet telephony, etc.).

Digital video data requires a large amount of memory resources, for storage and manipulation, and bandwidth for transmission. Nowadays systems commonly pose some constraints on these resources which make video compression a mandatory step for all the most useful and used applications (Digital TV Broadcasting, Internet Video Streaming, Mobile Video Streaming, etc).

The consumer applications represent a very large market in continuous expansion. The revenues involved in digital TV broadcasting are considerable. Video coding is then an essential player of the game, for example, higher is the number of high-definition TV channels that a provider is able to allocate in the available transmission bandwidth, greater will be its success and its economic gain with respect to competitors. Furthermore, final users are getting more involved with video technologies and are able to discern the quality and the performance of video-based applications, driving the need for video coding technology improvements.

Hundreds of scientific papers have been proposed, suggesting innovative techniques for improving one or more elements of the video codec. In reality, commercial video coding applications tend to use a limited and well defined set of techniques which have been already standardized. Working with standardized video coding formats and algorithms makes inter-operability between encoders and decoders of different manufacturers possible. Then, several are the benefits of standardized video coding formats compared with non-standard proprietary solutions.

The last couple of decades have been characterized by the emersion of numerous video compression standards. All of them have been released by technical organizations and industrial corporations. The main organizations involved in the standardization activities include the International Standards Organization (ISO)[1] and International Telecommunications Union(ITU)[2]. The first digital video standard, named H.120 [80], has emerged in 1984. Since then, the ITU-T Video Coding Expert Group (VCEG) and the ISO/IEC Moving Picture Expert Group (MPEG) have joined forced for the development of new standards. The VCEG was mainly focused on standards for communication applications [81]-[82] while MPEG on high quality

[1] Link: http://www.iso.org/iso/home.html

[2] Link: http://www.itu.int/en/Pages/default.aspx

applications for storage [83] and video broadcasting applications [84]. Notwithstanding the different focuses, VCEG and MPEG designed together one of the most successful and widely used standard: the MPEG-2. Afterwards, the two organizations continued working independently, the former on the improvement of the H.263 standard first and on the design of the H.26L after, the latter on the development of the well know MPEG-4. The evolution of compression standards generated by MPEG and H.26X are very closely related. Many of the techniques adopted by MPEG's latest compression standard borrow from recent developments in H.26X's latest release, and vice versa. For these reasons, in late 2001 we assisted to a new fusion between VCEG and MPEG and to the creation of the Joint Video Team (JVT). The common scope was - and still is - the development of a standard designed to be integrated into networks, with a much higher coding efficiency compared with its predecessor, with improved network adaptation and simple syntax specifications. In may 2003 the H.264/AVC (Advanced Video Coding) Recommendation [85] has been approved by the ITU-T and similarly has occurred within ISO/IEC. Since this moment, the JVT has worked on the definition of the H.264/AVC standard with its extensions and, recently, is working on the forthcoming standard HEVC (High Efficiency Video Coding). The Recommendation H.264 Advanced Video Coding, published in March 2009 by the JVT, defines a format or syntax for compressed video and a method for decoding this syntax to produce a displayable video sequence. The standard document does not actually specify how to encode digital video - this is left to the manufacturer of a video encoder - but in practice the encoder is likely to mirror the steps of the decoding process. It builds on the concepts of earlier standards such as MPEG-2 and MPEG-4 Visual and offers the potential for better compression efficiency, i.e. better-quality compressed video, and greater flexibility in compressing, transmitting and storing video.

The H.264/AVC is at the moment the state-of-the-arts standard in the video compression field. A complete description of the standard is out of the scope of this written, the interested reader is then kindly addressed to [86],[11].

B.2 Standard H.264/AVC Overview

H.264 Advanced Video Compression is an industrial standard for video coding that defines a format or syntax for compressed video and a method of decoding this syntax. It provides a set of tools or algorithms that can be used to deliver efficient, flexible and robust video for a wide range of applications, from low-complexity, low bit-rate mobile video applications to high-definition broadcast services. H.264/AVC has been developed to address a large range of applications, bit rates, resolutions, qualities, and services; in other words, H.264/AVC intends to be as generically applicable as possible. In order to maximize the interoperability while limiting the complexity, targeting the largest deployment of the standard, the H.264/AVC specification defines *profiles* and *levels*. Profiles and levels together specify restrictions on the bit streams and minimum bounds on the decoding capabilities, making possible to implement decoders with different limited complexity, targeting different application domains. Encoders are not required to make use of any specific set of tools but only to generate bit streams which are compliant to the correct profile and level combination.

A profile is a subset of the coding tools. In order to achieve a subset of the complete syntax, flags, parameters, and other syntax elements are included in the bit stream that signal the presence or absence of syntactic elements that occur later in the bit stream. All decoders compliant to a certain profile must support all the tools in the corresponding profile.

A level is a specified set of constraints imposed on values of the syntax elements in the bit stream. These constraints may be simple limits on values or alternatively they may take the form of constraints on arithmetic combinations of values. Each level specifies upper bounds for the bit stream or lower bounds for the decoder capabilities, e.g., in terms of picture size, decoder processing rate, size of the memory for multi-picture buffers, video bit rate, and motion vector range. In H.264/AVC, the same level definitions are used for all profiles defined.

Different profiles have been included in the standard for covering the heterogeneity of receivers and application scenarios. The profile most widely used is the *Main Profile* (MP) which provides a good tradeoff compression performance/computational complexity. The *Baseline Profile* (BP) - used for mobile applications - targets low-cost applications with limited computational resources. The High Profile (HP) is the primary profile for broadcast and disc storage applications, particularly for high-definition television applications (this is the profile adopted into HD DVD and Blu-ray Disc, for example). Other profiles have been defined, for more detailed information on the H.264/AVC profiles and levels, refer to Annex A of [85].

Before going more in detail in the encoding/decoding main steps, it is useful to understand the structure of the H.264 syntax. Fig. B.1 show a syntax structure overview. At the beginning, the H.264 sequence consists of a series of packets or Network Adaptation Layer Units, NAL Units or NALUs including parameter sets containing key information and slices (coded video frames or parts of them). At the next level, a slice is organized in coded macroblocks (MB), each containing compressed data corresponding to a (16x16) block of displayed pixels in a video frame. At the end, a macroblock contains type information describing the particular choice of methods used to code it, prediction information (such as coded motion vectors or intra prediction mode information) and coded residual data.

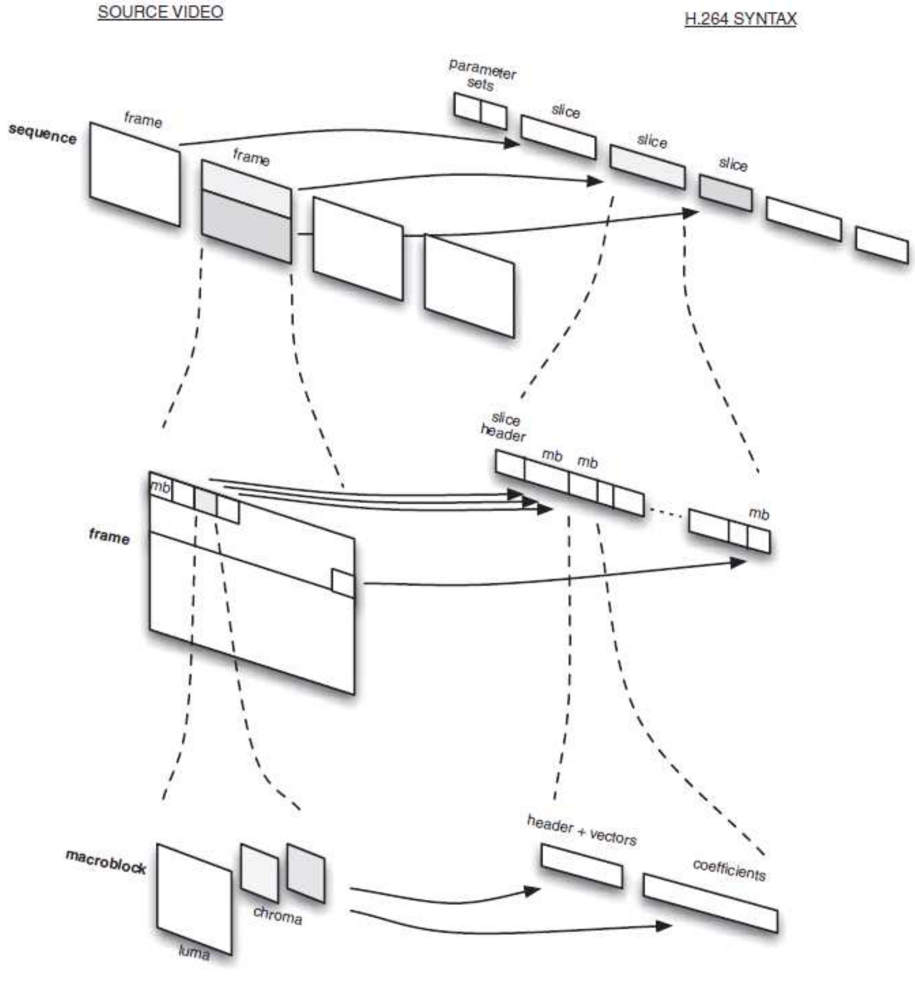

Figure B.1: Overview of the H.264 Syntax [11].

An H.264 video encoder carries out **prediction**, **transform** and **encoding** processes to produce a compressed H.264 bitstream. An H.264 video decoder mirrors the encoder's processes carrying out **decoding**, **inverse transform** and **reconstruction** to produce a decoded video sequence. A sequence of original video frames or fields is encoded into the H.264 format and results in a bitstream which represents the video in compressed form. This compressed bitstream is stored/transmitted and the original video flow can be reconstructed by decoding the video sequence. Being the H.264/AVC a lossy compression format, generally the decoded version is not identical to the original one. As you know, the H.264/AVC standard does not specify how to encode the input video sequence, but only how the output bitstream of a compatible video encoder has to be organized in order to be "readable" from any H.264 decoder. The video sequence given as input, composed by video frame, is segmented in macroblocks (MB) of 16x16 displayed pixel. The macroblocks are from this moment on the units to be processed. In the encoder, a prediction macroblock is generated and subtracted from the current one to form a residual. The residual is then transformed, quantized and encoded. In parallel, the quantized data are re-scaled and inverse transformed and added to the prediction macroblock to reconstruct a coded version of the frame which is stored for later predictions. In the decoder, a macroblock is decoded, re-scaled and inverse transformed to form a decoded residual macroblock. The decoder generates the same prediction that was created at the encoder and adds this to the residual to produce a decoded macroblock. Figures B.2-B.3 give a typical outline of H.264 encoder and decoder respectively. Let's describe in more detail the main processes of encoder and decoder

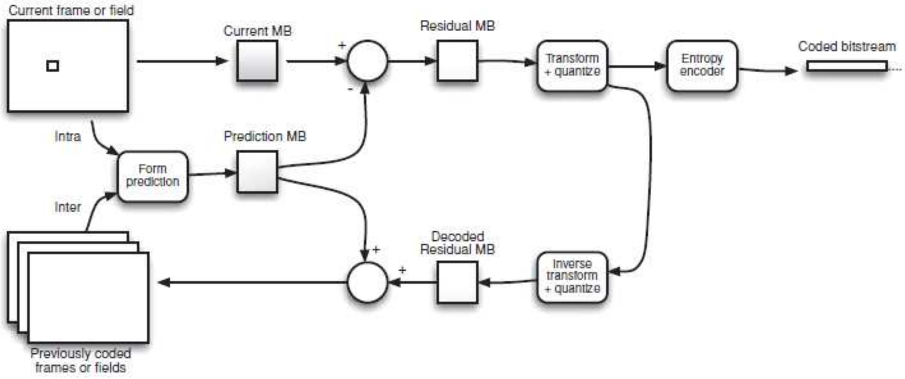

Figure B.2: H.264 encoder[11].

procedures:

- **Encoder Side**
  - *Prediction*

    A *prediction* of the current MB is generated based on previously-coded

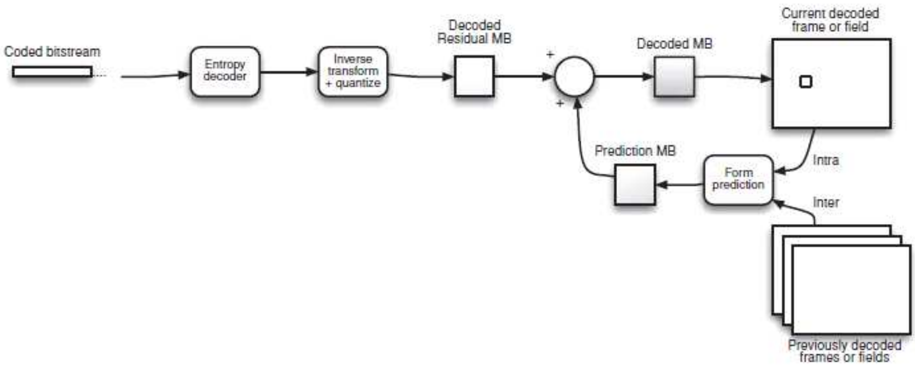

Figure B.3: H.264 decoder[11].

data. In case of intra prediction (I-frame) values of the previously-coded neighbouring pixels are extrapolated to form a prediction of the current MB. Intra prediction uses block sizes of (16x16) and (4x4) to predict the MB from surrounding, previously coded pixels within the same frame (Fig. B.4). On the contrary, inter prediction (P- and B- frames) uses MBs of frames which have already been encoded and transmitted. As block sizes a range from (16x16) down to (4x4) is used to predict pixels in the current frame from similar regions in previously coded frames, which may occur before or after the current frame in display order (Fig. B.5). At this point, the prediction is subtracted from the current MB, generating the so called *residual.* Prediction methods supported by H.264 are very flexible, enable accurate predictions resulting in efficient video compression.

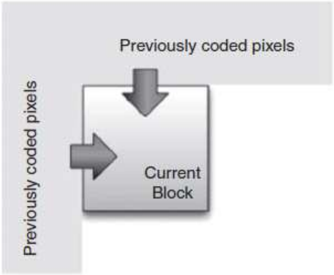

Figure B.4: Intra Prediction [11].

- *Transform & quantization*

A set of coefficients is achieved applying an integer transform on blocks of

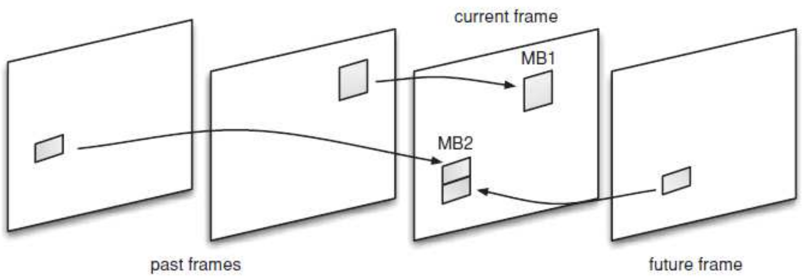

Figure B.5: Inter Prediction [11].

(4x4) or (8x8) pixels. These coefficients are used as a weighting factor for a standard basic pattern. When combined, the weighted basis patterns re-create the block of residual samples. The block of transform coefficients is then quantized using a Quantization Parameter (QP) which influences the precision of the transform coefficients. High values of QP result is high compression at the expenses of poor quality; instead, low QP values result in better reconstructed quality at the expenses of low compression.

- *The bitstream encoding*

  With previous operations, a number of values to be encoded in a compressed bitstream has been generated. In addition to the quantized transform coefficients also other information are now well known, s.a. how the decoder should re-create the prediction, which tools have been used and further information on the complete video flow. By means of variable length coding (VLC) and/ or arithmetic coding a binary flow is achieved. The encoded bitstream can then be stored and/or transmitted.

- **Decoder Side**

  - *Bitstream decoding*

    The H.264-complaint bitstream after transmission or storage is given as input to an H.264 decoder. The syntax elements along with the other information are recovered by means of the opportune decoding technique.

  - *Rescaling & inverse transform*

    The quantized transform coefficients are re-scaled. Each coefficient is multiplied by an integer value to restore its original scale. For obvious reasons, the recovered coefficients are similar but not identical to the originals. The re-scaled coefficients are used as weights of the standard basic patterns to re-create each block of residual data. These blocks are combined together to form a residual MB. The reconstructed blocks are similar but not identical to the original block due to the forward quantization process.

– *Reconstruction*

For each MB and accordingly to the prediction type applied to the MB itself, the decoder generates a prediction as done by the encoder. This prediction is added to the decoded residual for reconstructing a macroblock ready to display (Fig. B.6).

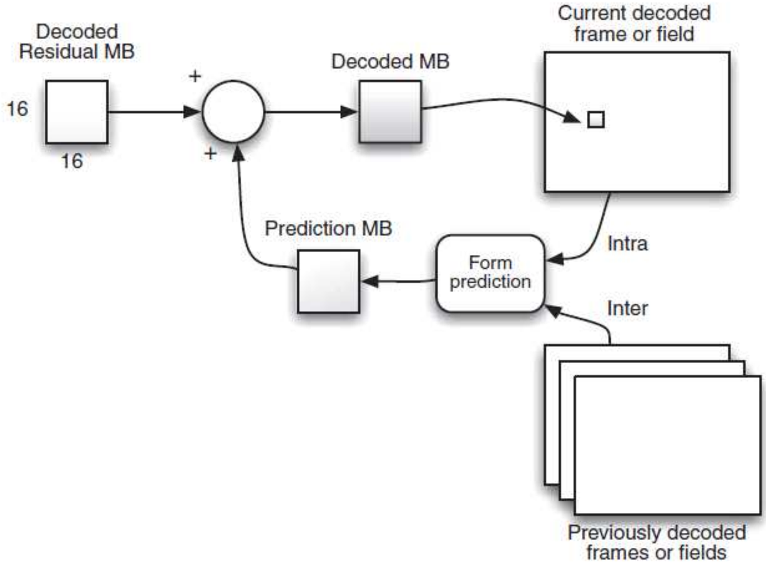

Figure B.6: H.264 Reconstruction [11].

B.2.1 H.264/AVC Standard Extensions

The given description of the H.264/AVC standard is related to the standard version excluding its further extensions. In reality, the video coding industry continues to grow up quickly demanding ubiquitous services and applications. Platforms and delivery mechanisms for video applications are experiencing an evolution phase characterized by an increasing expectation of available contents on any platform from mobile to HD and 3D displays, over any network including broadcast, internet, mobile, etc. For facing with these needs, the standard itself has evolved since 2003.

The main driver of this evolution has been the increasing need for coding the same original content at different bandwidths and display resolutions. This led to the development of the Scalable Video Coding (SVC) extension to H.264, standardized as H.264 SVC. Scalable coding, also called layered coding, is suitable for transmission over noisy channel since the more important layers (e.g., the base layer) can be better protected and sent over a channel with better error performance. Scalable coding is also used in video transport over variable-bit rate channels. When the channel bandwidth is reduced, the less important enhancement layers may not be transmitted. It is also useful for progressive transmission, which means the users can get rough representations of the video fast with the base layer and then the video quality will be refined as more enhancement data arrive. With the most recent diffusion of 3D technologies and 3D Televisions, another trend towards creation and delivery of multiple views of the same scene developed. To this extent, also tools for multiview video coding have been standardized as H.264 MVC. SVC supports efficient coding of video in such a way that multiple versions of the video signal can be decoded at a range of bitrates, spatial resolutions and/or temporal resolutions or frame rates. By jointly coding multiple versions, it should be possible to deliver them in a more efficient way than the alternative of coding and transmitting each version separately. SVC can be employed in various applications s.a. Archiving and Storing (storing a sequence as a scalable bitstream allows the fast recovery of a low-quality preview of the video sequence), Multiple Decoders (A scalable bitstream can efficiently support a wide range of decoding capabilities) and Graceful Degradation (Scalable coding offers a mechanism for maximizing the quality at a particular point in time for a specific decoder). On the other end, multiview applications require coding of multiple, closely related video signals (different views of the same scene). Similarly to SVC, Multiview Video Coding (MVC) exploits the correlation between these views to deliver efficient compression. Also for MVC, several are the potential applications, such as stereoscopic and auto-stereoscopic TVs as well as free-viewpoint applications, immersive teleconferencing and gaming.

Before going more in detail giving an overview of these two extensions, it is important to clarify that their direct competitor is represented by the **simulcast** transmission, characterized by single layer (SL) bit-streams. In this case, for delivering multiple version of a video sequence, each of them has to be encoded independently, for example with H.264/AVC or other standards. This means that, if we want to ensure the same video content to three different decoders (or clients) with different capabilities, the original video stream has to be encoded and transmitted three times, with the consequent waste of bandwidth and computational resources. Moreover, the adaptation of a single stream can be achieved through transcoding, currently used in multi-point control units in video conferencing systems or for streaming services in 3G systems. Hence, a scalable video codec has to compete against these alternatives [41]. In Fig. B.7, an exemplary SVC scenario is shown.

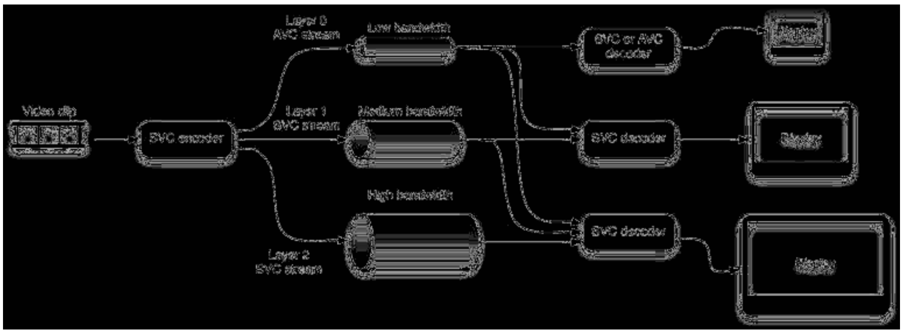

Figure B.7: SVC Scenario [11].

B.2.1.1 Scalable Video Coding (SVC)

Scalable Video Coding (SVC) extends the capabilities of the original standard and is incorporated as Annex G of the H.264/AVC standard [41],[87]. A reference software implementation - Joint Scalable Video Model Software - is available[3]. SVC delivers multiple coded versions of a video clip using a lower overall bit-rate in comparison with the simulcast scenario. This is achieved by exploiting correlations between different version of the same sequence coded at different operation points. As a general concept, a scalable bit-stream is that it is composed by multiple substreams which are generated in function of underlying sub-stream but which allow to remove parts of it still resulting in another valid bitstream for some target decoders. An SVC encoder generates a bit stream organized in "layers", each of which is an independent sub-stream providing incremental refinement with respect to lower layers. Independently form the number of layers/sub-streams composing the bit stream, each SVC flow presents a common base layer, the lowest one, whose correct reception and decoding is fundamental for decoding the video stream. On the top of the base layer, in the following indicated as BL, one or more enhancement layer can

[3] Joint Scalable Video Model software, http://ip.hhi.de/imagecomG1/savce/downloads/SVC-Reference-Software.htm

be provided. To decode the base layer a single-layer decoder, e.g. H.264 would be enough. The produced video sequence will typically have a low-quality/resolution. Instead, an SVC decoder is needed for decoding higher quality/resolution operating points achievable by decoding - in addition to the BL - the enhancement layer (layers). The SVC coding process exploits redundancy between sequences coded at different resolutions or qualities. In order to do that, it uses prediction mechanisms for generating additional layers from a base layer and the eventual lower layers. In this way, it should be possible to achieve the same displayed result as the simulcast system at a reduced bandwidth cost [11].

In Annex G of the H.264/AVC standard three main scalability types have been defined:

1. **Spatial scalability**.

   With spatial scalability, the applications can support users with different resolution terminals. The original content is first down-sampled by spatial decimation to obtain a lower resolution and then is encoded as base layer. The EL is generated starting from the decoded version of the base layer. The decoded BL video signal is up-sampled by spatial interpolation, then weighted and combined with the motion-compensated prediction from the enhancement layer. The selection of weights is done on a macroblock basis and the related information is sent as a part of the EL bit-stream. Each layer has a good degree of flexibility in the selection of other parameters, s.a. frame rate and size. BL and EL are then sent together over the channel. At the decoder, the BL is decoded to obtain the lower resolution video. Then, it is interpolated and weighted and added to the motion-compensated prediction from the enhancement layer. For example, let's suppose to have a spatial scalable video flow with two layers. An input video frame F is down-sampled to produce a low-resolution version $\widehat{F}$. The down sampled frame is now encoded as BL. Once decoded it will provide a low-resolution fame $\widetilde{F}$. The decoded low resolution frame is now up-sampled and used as reference for the prediction operation. The prediction so-obtained is encoded as EL. A decoded will follow a similar path for decoding the full resolution video frame (decoding the BL, up- sampling, adding prediction). Clearly, this process may be repeated to give additional layers with progressive spatial resolution. Spatial scalability can be achieved using some tools already available in H.264/AVC standard, without requiring extensions. Nevertheless, while coding an EL macroblock some modifications are needed in function of which has been the prediction mechanism used in the BL (or lower layers). H.264/SVC enables additional prediction modes for improving coding performance in comparison with the H.264:

- Reference Layer Up-Scaling: used for Intra blocks, the reference layer is scaled to the same resolution as the current layer and used as an extra prediction reference.
- Base Mode. Use the prediction choices from the corresponding reference layer macro block. Only a residual is sent in the enhancement layer, as prediction parameters the ones of the reference layer are used.
- Motion vector prediction from the reference layer. The EL MB partition is predicted using Inter prediction with the same reference picture indices as the corresponding reference layer MB and motion vector differences (MVD) created by using as predictors the up-scaled motion vectors of the reference layer.
- Residual prediction. The EL residual is predicted from the reference layer residual.

A typical application scenario of spatial scalability is the delivery of TV channels in standard (SDTV) and high-definition (HDTV). The first one can be encoded to form the base layer; all standard quality receiver will decode this layer only. An enhancement layer can provide the HDTV resolution. HD receivers will decode both layers.

2. **Temporal scalability**. In temporal scalable coding, the base layer is coded at a lower frame rate. The decoded base layer pictures provide motion-compensated predictions for encoding the enhancement layer. This kind of scalability has been designed for video services demanding different temporal resolutions, i.e. wireless video communications that may require to drop the video frame rate in case of poor channel conditions. Temporal scalability can be achieved using the P-and/or B-slice coding tools available in H.264/AVC, supported in both Main and High Profiles. This means that it can be achieved without any modification to the core of the H.264/AVC standard. Let's suppose to have a temporal scalable video flow with two layers. BL frame rate is fps and consists of coded frames 0,4,8, etc. EL frame rate is fps and consists

of frames 2,6,10,etc. An H.264/AVC decoder can decode BL only - to produce a sequence which a frame rate of fps - or BL and EL to produce an higher rate sequence at $2 * \widetilde{fps}$ frames per second. As for the spacial scalability case, also this process may be extended to give additional layers. It is worth noticing that, each additional layer is flexible in the choice of parameters s.a. the frame rate to use, etc.

3. **Quality scalability**. In quality (or SNR) scalable coding, the base layer is coded at a low visual quality (high QP values). The decoded base layer is then re-encoded at higher quality (low QP values). Spatial and temporal resolution are unchanged. The H.264/SVC extension supports two types of quality scalability:

    - Coarse Grain Scalability (CGS). It is a special case of spatial scalability in which the up/down-sampling factor is 1, hence the EL has the same resolution than the reference layer. Lower QP values are used for encoding the EL, granting an higher quality. All the spatial scalability tools cited above can be applied for predicting the EL from the reference layer. With this approach, each layer has a different quality and then a different bit-rate.
    - Medium Grain Scalability (MGS) allows the extraction of sub-streams at a wide range of bit-rates from a scalable bit-stream with a small number of quality layers. Any NAL unit in an EL may be discarded, still providing a fully decodable bit-stream. In this way, a variety of output bit-rates can be provided. Using MGS, selected Enhancement Layer NALUs may be discarded to provide sub-streams at a progressively lower bit-rate. Some constraints are introduced in the motion compensation prediction operation.

Therefore, different scalability types can be combined into hybrid coding schemes, i.e. spatial-temporal. Fig.B.8 shows a graphical overview of the different scalability types discussed above.

For further reference and an extensive explanation of the Scalable Video Coding extension, the reader is kindly advised to [41]. Readers interested to SVC performance in comparison with traditional AVC are invited to refer to [88].

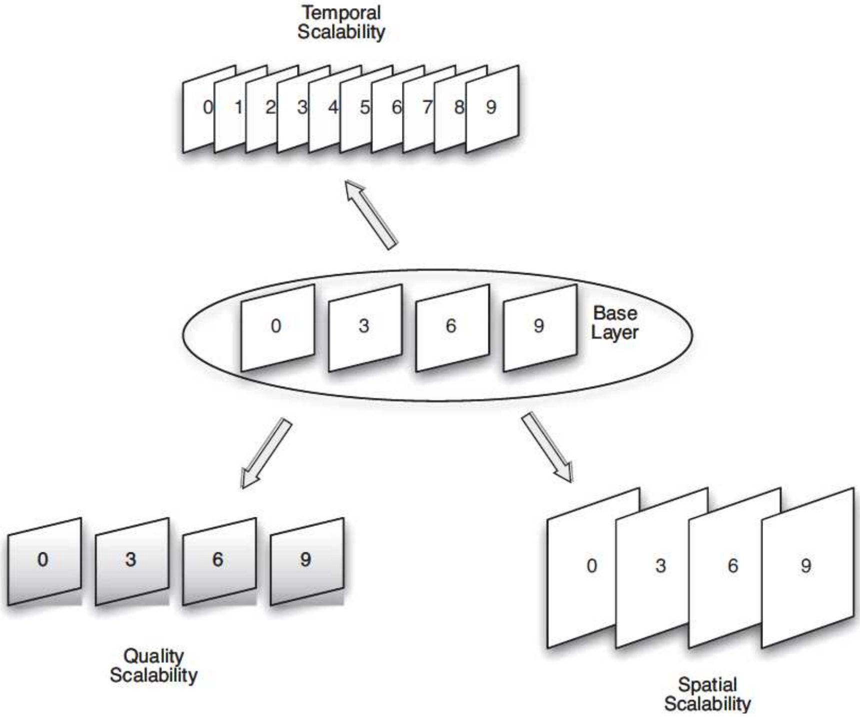

Figure B.8: Scalability types overview [11].

B.3 Multi-view Video Coding (MVC)

H.264 Multiview Video Coding is incorporated as Annex H into a draft revision of H.264/AVC [50]. A reference software is available [89]. A multi-view video is a collection of multiple videos capturing the same scene from different perspectives. Each view consists of a series of frames or fields that may be coded as a separate H.264/AVC stream. The size of the multiview flow increases with the number of cameras/views, hence efficient compression techniques are essential for its storage and transmission. As the video data originate from the same scene, exploiting similarities among the multiview video images is the key to efficient compression. These similarities can be classified into two types:

1. **Inter-view similarities**, within views captured by adjacent cameras.

2. **Temporal similarities**, within temporally successive frames of the same view.

Several motion compensation techniques have been developed for exploiting temporal similarities, based for example on block-matching techniques, variable block size, and many others. Regarding disparity compensation techniques, the simplest approach is based on block matching - like for motion compensation - that does not require knowledge of the geometry of the underlying 3D objects. More advanced approaches to disparity compensation are depth-image-based rendering algorithm which synthesize an image as seen from a given viewpoint by using the reference texture and depth image as input data. These techniques are really performing but rely on depth images, which are difficult to estimate. Indeed, also hybrid solution exists. Practical multiview compression schemes exploit the similarities (redundancies) with either predictive coding or with sub-band coding. Predictive coding schemes encode multiview video in a sequential way. As for general prediction mechanisms, pictures can be coded as Intra or Inter picture. Intra pictures are coded independently without referencing any other images or views. Inter pictures depend on one or more reference pictures previously encoded. Also combination of the two are possible, enabling the so called "hierarchical prediction". In sub-band coding schemes, all images to be encoded are subject to a sub-band decomposition that is followed by quantization and entropy-coding of its coefficients. Such schemes do not require sequential processing of images and offer more flexible multiview video representations. As in predictive coding, the sub-band decomposition exploits similarities among views by motion and disparity compensation.

Annex H to H.264/AVC specifies a number of additions to the basic H.264 syntax to support MVC, including:

- Sequence Parameter Set: specify views and anchor or key picture references.
- Reference Picture List: structured to include support for inter-view prediction.
- NAL Unit order: modified to allow the use of a Prefix NALU, containing extra information about the Base view. This special Prefix NAL Unit may be discarded by an AVC decoder that is not MVC-compatible, so that the base view may still be decoded.
- Picture numbering and reference indices: modified to support multiple views.

For further references on the topic the reader is invited to [11],[90],[91] and herein references.

CONCLUSIONS AND FUTURE DEVELOPMENTS

This PhD thesis has addressed the design and the performance analysis of cross-layers protection techniques aimed at enhancing video signal robustness when transmission takes place over noisy and loss-prone channels.

More in detail, in Chapter 1 an overview of the state-of-the-art solutions to be employed for facing typical transmission chains' impairments has been provided. Different classes of error-correction methodologies have been presented and their applicability has been discussed in function of the elementary data units to be processed and/or of the stage of the transmission chain in which they act. The provided overview has demonstrated that the error-control problem in broadcasting and streaming applications is still an open issue and, albeit a multitude of performing solutions already exists, there is still room for further analysis.

Following from the state-of-the art analysis, in Chapter 2 the performance evaluation of a new compelling error-protection scheme applied on layered encoded video flows has been presented. The proposed solution, called LA-FEC UI, optimizes the joint application of SVC as a source encoder, the LA-FEC mechanism based on an ad-hoc extension of Raptor and RaptorQ encoding structures as a channel encoder and a clever, albeit straightforward, scheme of long time interleaving, as an additional protection mechanism against long error bursts. The applicability and the advantages of this design have been proved in real-system scenarios.

In Chapter 3, a novel reliable transmission technique, called Multi-Dimensional Multi-Layer Aware Forward Error Correction (MDLA-FEC), has been introduced. The MDLA-FEC scheme can be applied as a protection mechanism of multidimensional layered encoded media. The decoding procedure has been mathematically described in function of the transmission system infrastructure employed for an exemplary view setup. The complex inter-layer dependency structure of the media stream has been described and it has been exploited in the decoding procedure for enhancing the robustness of the media stream against transmission losses.

In order to account also for the real nature of the phenomena affecting the transmission link and to easy the design of error-protection mechanisms, in Chapter 4 a channel model able to emulate error bursts of fixed length has been proposed. These fixed-size bursts are generate as a consequence of the Repetitive Electrical Impulse Noise (REIN) typically affecting DSL links which are widely employed to serve as last mile technology for IPTV services and not only. A combinatorial analysis has been conducted in order to identify all the possible error distribution combinations which may occur on a FEC protected data block and which still lead to decodable data blocks. Finally, a mathematical formula for calculating in a really fast and accurate way the decoding probability of ideal FEC codes under REIN influence has been introduced. The accuracy of the model has been proved in comparison with extensive empirical simulation campaigns results, while its efficiency has been proved in terms of runtime performance.

Finally, in Chapter 5, a novel metric for objective video quality evaluation, applicable to loss-affected uncompressed video sequences, has been introduced. The motivations at the base of my researches in this topic are manifold. First of all the possibility of testing new upper layer FEC scheme designs in terms of video quality, rather than in terms of decoding probability in presence of unknown frame losses. Indeed, this technique allows to detect the position of lost frames within a received video sequence in a domain in which sequence/frame numbering, time-stamps and similar tools are not available and finally, to enable the full sequence reconstruction by means of sequence alignment recovery mechanisms.

In the appendixes the most important video communication networks and protocols have been briefly introduced and the state-of-the art H.264/AVC source encoder along with its extensions has been described.

Although in this PhD. some steps have been taken in the context of reliable video transmission techniques design, the possible developments in this research field are potentially limitless. Due to the extreme variety of applications, receivers and underlying transmission systems available, there are different challenging paths that can be followed for achieving a valid contribution. One of the most interesting evolutions is represented by the extension of the proposed LA-FEC UI to any possible number of enhancement layers, in a such a way that a 3D free viewpoint video stream can be encoded in a multi-layer/multi-dimensional stream and delivered in broadcast for serving potentially any kind of receivers, ranging from mobile low-quality terminals, to HDTVs, to 3D Freeview point devices. Some contributions have been provided by the author in this direction, but there is still room for further improvements. Another challenging issue is represented by the adaptation of the channel model proposed in Chapter 4 to the multi-layer case and to the not-ideal FEC codes case. This way, the model will result in a suitable tool for testing FEC scheme performance matching the transmission scheduling actually employed, in order to guarantee the accuracy of the results in real application scenarios.

PERSONAL PUBLICATIONS

[1] C. Hellge, V. Pullano, M. Hensel, G.E. Corazza, T. Schierl, and T. Wiegand, "Mobile TV with long Time Interleaving and Fast Zapping," *2012 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pp. 623–628, 2012.

[2] V. Pullano, C. Hellge, M. Hensel, G.E. Corazza, and T. Schierl, "Application Layer FEC with Long Time Interleaver and Fast Tune-in for Mobile Satellite TV Services," *2013 International Conference on Computing, Networking and Communications (ICNC) Workshop on Computing, Networking and Communications*, 2013.

[3] C. Hellge, M. Hensel, V. Pullano, T. Schierl, G.E. Corazza, and T. Wiegand, "Long Time Interleaving with Fast Zapping by Layer-Aware RaptorQ and SVC," In preparation for submission to *IEEE Transaction on Vehicular Technologies*, 2013.

[4] V. Pullano, R. Skupin, G.E. Corazza, C. Hellge, and T. Schierl, "Modeling effects of impulse noise on Application-Layer FEC in DSL channels," Accepted to *the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting 2013*, 2013.

[5] V. Pullano, A. Vanelli-Coralli, and G.E. Corazza, "PSNR evaluation and alignment recovery for mobile satellite video broadcasting," *6th Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC)*, pp. 176–181, 2012.

[6] V. Pullano, C. Hellge, G.E. Corazza, and T. Schierl, “A mathematical Modeling of Multilayer Video Stream FEC Decoding for Broadcasting and IPTV services,” In preparation, 2013.

BIBLIOGRAPHY

[7] F. Zhai, Y. Eisenberg, and A.K. Katsaggelos, “Joint Source-Channel Coding for Video Communications,” in *Handbook of Image and Video Processing*, Al Bovik, Ed. Elsevier Academics Press, 2nd edition edition, June 2005.

[8] C. Hellge, D. Gómez-Barquero, T. Schierl, and T. Wiegand, “Layer-Aware Forward Error Correction for Mobile Broadcast of Layered Media,” *IEEE Transaction on Multimedia*, vol. 13, no. 3, pp. 551–562, June 2011.

[9] Alan C. Bovik, *The Essential Guide to Video Processing*, Academic Press, USA, 2009.

[10] Colin Perkins, *RTP: Audio and Video for the Internet*, Addison Wesley, June 2003.

[11] Iain E. Richardson, *The H.264 Advanced Video Compression Standard*, John Wiley & Sons, Ltd, UK, 2010.

[12] Claude E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[13] P. Elias, “Coding for noisy channels,” *IEEE Transactions on Communications*, , no. 5, pp. 37–46, March 1955.

[14] R. M. Fano, “A heuristic discussion of probabilistic decoding,” *IEEE Transactions on Information Theory*, vol. IT-9, no. 2, pp. 64–73, April 1963.

[15] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[16] D. Shu Lin and J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice Hall, 1983.

[17] I. S. Reed and G. Solomon, “Polynomial Codes Over Certain Finite Fields,” *SIAM Journal of Applied Math.*, vol. 8, pp. 300–304, 1960.

[18] ETSI EN 302 304, Digital Video Broadcasting (DVB), *Transmission System for Handheld Terminals (DVB-H)*, ETSI, November 2004.

[19] Frédéric Didier, “Efficient erasure decoding of Reed-Solomon codes,” *CoRR*, vol. abs/0901.1886, 2009.

[20] R. G. Gallager, “Low Density Parity Check Codes,” *MIT Press*, 1963.

[21] M. Luby, “LT Codes,” *Proceedings of the IEEE Symposium on the Foundations of Computer Science*, pp. 271–280, November 2002.

[22] A. Shokrollahi, “Raptor Codes,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[23] M. Luby, A. Shokrollahi, M. Watson, and T. Stockhammer, *Raptor Forward Error Correction Scheme for Object Delivery*, IETF - RFC 5053, October 2007.

[24] 3GPP, *Multimedia Broadcast/Multicast: Protocols and Codecs*, TS26.346.

[25] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, and L. Minder, *Raptor(Q) Forward Error Correction Scheme for Object Delivery*, IETF RMT - RFC 6330, August 2011.

[26] M. Luby, M. Watson, T. Gasiba, T. Stockhammer, and W. Xu, “Raptor codes for reliable download delivery in wireless broadcast systems,” *in Proc. of IEEE Consumer Communications and Networking Conf.*, vol. 1, pp. 192–197, 2006.

[27] R.E. Van Dyck and D.J. Miller, “Transport of wireless video using separate, concatenated, and joint source-channel coding,” *Proceedings of the IEEE*, vol. 87, pp. 1734–1750, October 1999.

[28] Yao Wang, Ya-quin Zhang, and Joern Ostermann, "Error Control in Video Communications," in *Chapter 14 in Video Processing and Communications*, Al Bovik, Ed. Prentice Hall PTR, 1st edition edition, 2001.

[29] ITU-R Recommendation BT.500-11, *Methodology for the subjective assessment of the quality of television pictures*, International Telecommunication Union, 2002.

[30] ITU-T Recommendation P.911, *Subjective audiovisual quality assessment methods for multimedia applications*, International Telecommunication Union, 1998.

[31] H. R. Wu and K. R. Rao, "Video quality testing," in *Chapter 4 of Digital Video Image Quality and Perceptual Coding*, P. Corriveau, Ed. 2006.

[32] ITU-T Recommendation J.144, *Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference*, International Telecommunication Union, 2004.

[33] ITU-R Recommendation BT.1683, *Objective perceptual video quality measurement techniques for standard de?nition digital broadcast television in the presence of a full reference*, International Telecommunication Union, 2004.

[34] S. Winkler, "Video Quality Measurement Standards - Current Status and Trends," *In Proceedings of ICICS 2009*, pp. 1–5, December 2009.

[35] Y. Wang, "Survey of Objective Video Quality Measurements," *Technical Report WPICS-TR-06-02*, February 2006.

[36] S. Winkler, *Digital Video Quality - Vision Models and Metrics*, John Wiley & Sons, 2005.

[37] 3GPP TS 22.246, *3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS) user services; Stage 1 (Release 10)*, 3GPP, 2011.

[38] M. Rezaei, I. Bouazizi, V.K.M. Vadakital, and M. Gabbouj, "Optimal channel changing delay for mobile tv over dvb-h," in *In Proc. of IEEE International Conference on Portable Information Devices (PORTABLE'07)*, 2007, pp. 1–5.

[39] J. Maisonneuve, M. Deschanel, J. Heiles, W. Li, H. Liu, R.Sharpe, and Y. Wu, "An overview of IPTV standards development," *Broadcasting, IEEE Transactions on*, vol. 55, no. 2, pp. 315–328, 2009.

[40] B. Sayadi, Y. Leprovost, S. Kerboeuf, M.L. Alberi-Morel, and L. Roullet, "MPE-IFEC: An enhanced burst error protection for DVB-SH systems," *Bell Labs Technical Journal*, vol. 14, no. 1, pp. 25–40, 2009.

[41] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, September 2007.

[42] ISO/IEC JTC1/SC29/WG11 MPEG2012/m23962, "Report of CE on D1-FEC: Delivery 1-Layer AL-FEC," San Jose, USA, February, 2012.

[43] A. Morello, V. Mignone, P. Burzigotti, and G. Vitale, "Upper Layer FEC for DVB-SH: Technical Solutions and Performance," *IBC Conference*, September 2007.

[44] M. Ibnkahla, Q.M. Rahman, A.I. Sulyman, H.A. Al-Asady, Jun Yuan, and A. Safwat, "High-speed satellite mobile communications: technologies and challenges," *Proceedings of IEEE*, vol. 92, no. 2, pp. 312–339, February 2004.

[45] A. Vanelli-Coralli, G.E. Corazza, G.K. Karagiannidist, P.T. Mathiopoulosl, D.S. Michalopoulost, C. Mosquera, S. Papaharalabost, and S. Scalise, "Satellite Communications: Research Trends and Open Issues," *International Workshop on Satellite and Space Communications (IWSSC'07)*, pp. 71–75, 2007.

[46] TMSSP0263r3, *IPDC File Delivery over DVB-SH: Simulation and Evaluation Framework*, February 2008.

[47] DVBTM-3731, *Implementation Guidelines for DVB-SH (DVB TM-SSP252)*, DVB.

[48] F. Pérez Fontan, M. Vázquez-Castro, C. Enjamio Cabado, J. Pita Garcia, and E.Kubista, "Statistical Modeling of the LMS Channel," *IEEE Trans. on Vehicular Technology*, vol. 50, no. 6, November 2011.

[49] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *Information Theory, IEEE Transactions on*, vol. 42, no. 6, pp. 1737–1744, Nov 1996.

[50] Joint Video Team Document JVT-AD007, *Editors Draft Revision to ITU-T Rec. H.264 — ISO/IEC 14496-10 Advanced Video Coding*, February 2009.

[51] C. Hellge, T. Schierl, and T. Wiegand, "Multidimensional layered forward error correction using rateless codes," *IEEE International Conference on Communications (ICC'08), Beijing, China*, May 2008.

[52] Technical Report TR-100, *ADSL2-ADSL2plus performance test plan*, March 2007, Broadband Forum.

[53] D. Levey and S. McLaughlin, "The statistical nature of impulse noise interarrival times in digital subscriber loop system," *Signal Processing*, vol. 82, no. 3, pp. 329–351, March 2002.

[54] M. Luby, T. Stockhammer, and M. Watson, "Iptv systems, standards and architectures: Part ii - application layer fec in iptv services," *IEEE Communication Magazine*, vol. 46, no. 5, pp. 94–101, 2008.

[55] C. Perkins M. Ellis, D. Pezaros, "Performance analysis of al-fec for rtp-based streaming video traffic to residential users," *Proceedings of 2012 IEEE 19th International Packet Video Workshop, Munich, Germany*, pp. 1–6, May 2012.

[56] DVB Blue Book A115, *DVB Application Layer FEC Evaluations*, May 2007, Available at: http://www.dvb.org/technology/standards.

[57] Bas Ven Den Heuvel, "Vsdl2 should also withstand "pein" impulse noise," *ETSI STC TM6*, February 2007.

[58] ETSI TS 102 585, Digital Video Broadcasting (DVB), *Digital Video Broadcasting; System Specifications for Satellite Services to Haldelh Devices (SH) below 3GHz*, ETSI, April 2008.

[59] ETSI EN 302 583, Digital Video Broadcasting (DVB), *Digital Video Broadcasting; Framing Structure, Channel Coding and Modulation for Satellite Services to Haldelh Devices (SH) below 3GHz*, ETSI, March 2008.

[60] ETSI TS 102 584, Digital Video Broadcasting (DVB), *DVB-SH Implementation Guidelines*, ETSI, December 2008.

[61] U. Reimers and A. Morello, "DVB-S2: the second generation standard for satellite broadcasting and unicasting," *International Journal of Satellite Communications and Networking*, vol. 22, no. 3, pp. 249–268, 2004.

[62] G. Faria, J. Henriksson, E. Stare, and P.Talmola, "Digital broadcast services to handheld devices," *Proceeding of IEEE*, vol. 94, no. 1, pp. 194–209, 2006.

[63] E.N. Gilbert, "Capacity of a burst-noise channel," *Bell System Technical Journal*, vol. 39, pp. 1253–1256, 1960.

[64] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell System Technical Journal*, vol. 42, pp. 1977–1997, 1963.

[65] U. Horn, K. Stuhlmuller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Signal Processing:Image Communication*, vol. 15, no. 1-2, pp. 77–94, 1999.

[66] P. Almstrom, M. Rabi, and M. Johansson, "Networked state estimation over a Gilbert-Elliot type channel," *Joint 48th IEEE Conference on Decision and Control*, December 2009.

[67] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G.J. Sullivan, "Rate constrained coder control and comparison of video coding standards," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 688–703, July 2003.

[68] B. Girod, "What's wrong with mean squared error?," *Digital images and human vision, Andrew B. Watson*, pp. 207–220, 1993.

[69] International Telecommunications Union, *ITU-R Recommendation BT.601-5: Studio encoding parameters of digital television for standard 4:3 and wide-scree 16:9 aspect ratios*, 1995.

[70] ANSI/T1E1.4/94-007, *Asymmetric Digital Subscriber Line (ADSL) Metallic Interface*, August 1997.

[71] J. M. Cioffi, "Asymmetric Digital Subscriber Lines," in *Chapter 34, Communications Handbook*, J.D.Gibson, Ed. CRC Press in cooperation with IEEE Press, 1997.

[72] L.M. Surhone, M.T. Tennoe, and S. F. Henssonow, *ANSI T1.413*, VDM Publishing, 2010.

[73] This C. Has, *G.992.1 (G.dmt) Draft Recommendation.*

[74] ETSI, *Global System for Mobile Communications.*

[75] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications*, Internet Engineering Task Force - RFC 1889, January 1996.

[76] D. Hoffman, G. Fernando, V. Goyal, and M. Civanlar, *RTP Payload Format for MPEG1/MPEG2 Video*, Internet Engineering Task Force - RFC 2250, January 1998.

[77] S. Wenger, M.M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, *RTP Payload Format for H.264 Video*, Internet Engineering Task Force - RFC 3984, February 2005.

[78] J. van der Meer, D. Mackie, V. Swaminathan, D. Singer, and P. Gentric, *RTP Payload Format for Transport of MPEG-4 Elementary Streams*, Internet Engineering Task Force - RFC 3984 - Proposed Standard, 2003.

[79] Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, and H. Kimata, *RTP Payload Format for MPEG-4 Audio/Visual Streams*, Internet Engineering Task Force - RFC 3016, 2000.

[80] ITU-T H.120, *Codec for Videoconferencing using Primary Digital Group Transmission*, International Telecommunication Union, 1984:1988, version1:version2.

[81] ITU-T H.26a, *Video Codec for Audiovisual Services at px 64kbit/s*, International Telecommunication Union, 1984:1988, version1:version2.

[82] ITU-T H.26b, *Video Codec for Audiovisual Services at px 64kbit/s*, International Telecommunication Union, 1995:1998:2000, version1:version2:version3.

[83] ISO/IEC, MPEG1-ISO/IEC 11172-1:1993/Cor1:1996/Cor2:1999 Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s: System, 1999.

[84] ISO/IEC, *ISO-IEC Recommendation 13818: Generic coding of moving pictures and associated audio information.*

[85] ITU-T and ISO/IEC JTC1, *Advanced Video Coding for Generic Audiovisual Services*, International Telecommunication Union, 2003.

[86] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[87] ETSI EN 302 304, Digital Video Broadcasting (DVB), *Coding of Audiovisual Objects- Part 10: Advanced Video Coding,* also ITU-T H.264 — ISO/IEC 14496-10:2009, 2009, *Advanced video coding for generic audiovisual services.*

[88] ISO/IEC, *ISO-IEC JTC1/SC29/WG11 N9577 SVC Verification Test Report*, Joint Video Team, January 2007.

[89] Joint Video Team Document JVT-AC207, *WD 3 Reference Software for MVC*, October 2008.

[90] M. Flierl and B. Girod, "Multiview Video Compression," *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 66–76, November 2007.

[91] Y. Chen, K. Wang, K. Ugur, M. Hannuksela, J. Lainema, and M. Gabbouj, "The Emerging MVC Standard for 3D Video Services," *EURASIP Journal on Advances in Signal Processing*, 2009.

ACKNOWLEDGMENTS

This thesis has reported the scientific outcomes of my PhD. and it is now the time to talk about the personal and human side of my experience. Listing all the person who have contributed to my professional and personal evolution during this three years period is a really difficult task and for this reason I would like to apologize in advance if I forget anyone. First of all, my deepest gratitude goes to Prof. Giovanni Emanuele Corazza and Prof. Alessandro Vanelli-Coralli, whose main lesson has been the importance of acquiring the autonomy and the self-confidence needed to remain standing in between this ”strange storm” called life. I owe my thanks to all the guys and girls of the Digicomm Group, with whom I have shared not only a simple office but much much more. The list of Digicommers is really too long but I want to thank to all of you guys....for the technical, and sometimes philosophical, discussions regarding engineering topics but mainly for laughing together, for sharing lunch breaks, for listening to me even when I was ”unbearable”. The Digicomm group is a big group to be part of and, out of everything, it has given me some special friends that I would really like to thank: Claudio Palestini for teaching me that being self-confident is the first step for going further, Alberto Candreva for his huge culture, for the rum-based evenings and for teaching me a lot of small and strange things that I will never forget, Giulio Gabelli for his extreme patience and reassuring smile and last, but absolutely not least, Lina Deambrogio, a guide, a friend, another sister... my ”Jiminy Cricket” as I love to call her.

My research activity has gained a considerable added value thanks to the 11-months period spent at the premises of the Fraunhofer HHI in the Image Processing Department, Multimedia Communications Group in Berlin. I am sincerely indebted with Dr. Thomas Wiegand and Dr. Thomas Schierl for giving me this huge opportunity and for their valuable guide. A really special thank goes to Cornelius Hellge, who has taught me more than any book or paper in the field, Estibaliz Guinea

Torre, for her friendship and sharing with me the "difficult task" of being women in a men-only office, Manuel Hensel for all our crazy mathematical delirium, Robert Skupin for the wonderful collaboration and for all the shared coffees and cigarettes, Tobias Mayer for teaching me that a language called Phyton exists and for sharing lunch breaks far away from "UDK"...
Finally, many thanks also to Yago, Ralph, Valeri, Karsten, Mauricio, Sergio for making the work place a funny place to be.
I would like to acknowledge also my international reviewers, Prof. Pascal Frossard and Dr. Dejan Vukobratovic, for their precious and greatly appreciated comments and suggestions.

Also my personal life sphere deserves some gratitude. I want to thank my friends Consuelo, Roberta and Valentina for being always available, for making me smile, for supporting my choices. My deepest gratitude goes to my family, my mom Aurora for her sweetness, my dad Aldo for his strength, for his tenacity and persistence in being the best mentor ever, my sisters, Francesca and Novella and my brother Mario, for following me in every moment despite the large distances. Finally, there is a last person to whom I absolutely owe my heartfelt gratitude, my boyfriend Rosario, for being the most solid reference point of this last years of my life, for not letting me give up, for being a glimmer of light in the dark moments, for walking together in the same direction....