UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

**Design and Optimization of Protection Strategies
Based on the Pro-MPEG COP3 Codes
for Time-sensitive Multimedia Streams**

Tesis Doctoral

César Díaz Martín
Ingeniero de Telecomunicación
2017

DEPARTAMENTO DE SEÑALES,
SISTEMAS Y RADIOCOMUNICACIONES

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

**Design and Optimization of Protection Strategies
Based on the Pro-MPEG COP3 Codes
for Time-sensitive Multimedia Streams**

Tesis Doctoral

Autor:

César Díaz Martín
Ingeniero de Telecomunicación
Universidad Politécnica de Madrid

Director:

Julián Cabrera Quesada
Doctor Ingeniero de Telecomunicación
Profesor del Departamento de Señales,
Sistemas y Radiocomunicaciones
Universidad Politécnica de Madrid

TESIS DOCTORAL

**Design and Optimization of Protection Strategies
Based on the Pro-MPEG COP3 Codes
for Time-sensitive Multimedia Streams**

Autor: César Díaz Martín

Director: Julián Cabrera Quesada

Tribunal nombrado por el Sr. Rector Magnífico de la Universidad Politécnica de Madrid, el día de de 2017.

Presidente: D.

Vocal: D.

Vocal: D.

Vocal: D.

Secretario: D.

Realizado el acto de defensa y lectura de la Tesis el día de de 2017 en

Calificación:

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO

Resumen

Durante los últimos años, la cantidad de servicios relacionados con la transmisión de vídeo que se proporcionan a través de redes IP ha ido aumentando de manera continua. El motivo principal de este incremento es la serie de ventajas que este tipo de redes presentan: ubicuidad, integración y sincronización sencillas entre servicios, aumento de la capacidad de interacción, etc. Sin embargo, las redes IP no constituyen un buen medio por el que distribuir vídeo. Por ello, para aumentar su fiabilidad se emplean típicamente mecanismos de gestión de la calidad de servicio (QoS), en particular en entornos de transmisión centralizados en el servidor y donde el servicio presenta condicionantes temporales fuertes. En estos escenarios, los códigos de protección uniforme Pro-MPEG COP3 se emplean de forma habitual, debido a su capacidad para lidiar con ráfagas de errores y su baja complejidad. Sin embargo, su eficacia decrece al empeorar las condiciones del canal de comunicación. Por otro lado, las estrategias de protección que tienen en cuenta la importancia desigual de los distintos elementos que componen el flujo de vídeo suelen obtener mejores resultados. Por contra, suelen ser notablemente más complejos que los códigos Pro-MPEG COP3.

En esta tesis se exploran estrategias que permitan mejorar la eficacia de los códigos Pro-MPEG COP3 sin comprometer su sencillez y conservando su funcionamiento básico. Las estrategias propuestas representan la generalización y optimización de dichos códigos, entre otras formas posibilitando que puedan proveer protección desigual al flujo de paquetes de datos.

En primer lugar se proponen dos extensiones a los códigos Pro-MPEG COP3 estándar. La primera de ellas permite utilizar hasta tres niveles de profundidad de entrelazado para incrementar su adaptabilidad a las condiciones del canal de comunicación y la decorrelación de errores. La segunda es un marco de actuación de baja complejidad para la protección desigual de la información que permite distribuir los recursos disponibles entre diferentes grupos de paquetes en función de la relevancia de estos. Para ello, el método propuesto permite usar no solo una matriz de generación por bloque de protección, como ocurre en el estándar, sino varias matrices con distintas dimensiones, lo que permite emplear distintas tasas de código a paquetes de distinta importancia.

A continuación se presenta un procedimiento para la optimización de la selección de las configuraciones más apropiadas en función de las características específicas del sistema dentro del marco de protección desigual descrito anteriormente. Esta estrategia de protección se basa en la hibridación de dos metaheurísticas comunes: recocido simulado

(simulated annealing -SA-) y búsqueda tabú (tabu search -TS-). Los procedimientos internos de ambas estrategias han sido modificados para encajar en las características del escenario de protección de la transmisión considerado, para así posibilitar la obtención de soluciones subóptimas en entornos en los que se imponen limitaciones temporales especialmente duras. Esta estrategia toma a su entrada una medida de la importancia de los paquetes que componen una ventana de observación, los valores de los parámetros que definen el comportamiento del canal y la cantidad de recursos disponibles, y encuentra en tiempo real configuraciones de protección cercanas a la óptima, en términos de número de matrices a utilizar y sus dimensiones.

Finalmente presentamos un modelo de distorsión sencillo pero que tiene en cuenta un contexto muy amplio de cada uno de los paquetes considerados. El modelo propuesto estima la contribución de cada paquete a la distorsión total esperada que se introduce en la secuencia presentada al usuario. Esta contribución, que representa la relevancia real del paquete, se obtiene empleando características de alto nivel del flujo de vídeo codificado y del comportamiento del canal de comunicación. El valor final no solo tiene en cuenta la degradación que provocaría la pérdida de este paquete y la probabilidad de que esto ocurra, sino que también considera el efecto que la posible pérdida de paquetes en cuadros/tiras de referencia tiene en la capacidad de distorsión del paquete en cuestión.

Abstract

In the recent years, an ever-increasing portion of the services related to video data transmission are supplied through IP networks, as they present numerous advantageous characteristics: ubiquity, easy service integration and synchronization, a far higher possibility for interactivity and many more. However, IP networks are not particularly well suited for video distribution. Thus, QoS management mechanisms are commonly employed to increase their reliability, particularly in server-driven, time-sensitive transmission scenarios. In these scenarios, the Pro-MPEG COP3 codes are commonly used, due to their capability to deal with packet loss bursts and to their very low complexity. However, their performance decreases with the packet loss rate. On the other hand, protection strategies that are aware of the uneven relevance of the different packets in the video data stream typically provide better performance, since they are capable of better distributing and using the available protection resources. However, their complexity usually exceeds that of the Pro-MPEG COP3 codes.

This thesis examines strategies to improve the performance of the Pro-MPEG COP3 codes without compromising their simplicity and preserving their encoding and decoding procedures. These strategies aim at a generalization and optimization of these codes, through, among other ways, enabling that they can be used to protect unequally the data packet stream.

First, we propose two extensions to the standard Pro-MPEG COP3 codes. The first one is an equal error protection (EEP) scheme that allows the use of up to three interleaving depths to boost further channel adaptation and error decorrelation. The second one is a low-complexity unequal error protection (UEP) framework that allows to allocate unequally the available protection resources among different sets of data packets regarding their relevance. To do that, the proposed approach enables the use of not only one, as in the standard Pro-MPEG COP3 codes, but a number of matrices of dissimilar dimensions per protection block. The use of this extension allows to apply uneven code rates to unevenly important data units.

Then, we present a procedure for optimizing the selection of suitable configurations to protect data when using the proposed UEP framework. This optimization strategy is based on the hybridization of two very well known metaheuristics: simulated annealing (SA) and tabu search (TS), whose core procedures are modified to fit the characteristics of the considered scenario, and so find near-optimal solutions when strict time restrictions apply. This strategy takes as input the importance of the video data packets and the

available resources, and finds in real time near-optimal configurations to protect them, in terms of number of matrices to be used and their dimensions.

Finally, we introduce a full-context-aware, yet lightweight distortion model. The proposed approach estimates the contribution of any given video packet to the resulting overall expected distortion of the sequence, and thus, its actual relevance, by taking into account upper-level characteristics of the encoded video streams and the behavior of the communication channel. This contribution is estimated considering not only the image degradation that its loss will potentially introduce in the sequence and the likelihood of this event occurring, but also the effect that the potential loss of data in reference frames/slices have on the actual importance of that packet.

Acknowledgments

Gracias, gracias y más gracias. A toda esa gente que de uno u otro modo ha participado, aportado, contribuido o apoyado en algún punto del desarrollo de esta tesis. Y a los que me han animado, soportado, consolado o escuchado (o fingido hacerlo lo suficientemente bien) en tiempos de desesperación y de aumento de la probabilidad de lanzamiento de toalla.

En particular, cómo no empezar agradeciendo a Julián, santo varón, todo el esfuerzo y paciencia que ha puesto encima de la mesa durante todos estos años. Si he llegado hasta aquí hoy ha sido en muy buena parte tanto por su guía y colaboración como por su habilidad para parar muchos de mis desvaríos investigativos y/o literarios.

Gracias a Narciso y a Fernando, por su disponibilidad, por sus consejos, por sus gestiones y por tanta y tanta ayuda, a F., por mostrarme la lengua más académica y por todas esas direcciones IP, y a Luis, siempre atento a la vida.

Grachemile al vector Encomienda ($\langle \overrightarrow{\text{Fili, Jesús, Raúl}} \rangle$), por sus innumerables y valiosísimas lecciones en las prestigiosas Universidades de la Vida, de la Música, de la Cocina, del Atleti..., a Dani, por llevarme desde lo más profundo de Linux a lo más alto del Cerro del Telégrafo, a Pablo, como mínimo por aquel Sigur Rós casi *a cappella* y por tantos y tantos céntimos ahorrados, a Carlos Roberto, por sacarme a trotar un soleado día de enero, a Massi, por su bubububububuh y por darle tanto brillo al bonito barrio de Tetuán, a Rafa, por su *flow*. Y a Jon, Carlos C., Susana, Tomás, Esther, Maykel, Gianlu, Sasho, Marcos, Virginia, Ana, Sergio, Richard, Lara, Marta... Tanta gente y tan estupenda.

Gracias a tantos amigos de hoy y siempre: del parque, de la Escuela, de aquel Stuttgart mítico... Imposible no mencionar explícitamente al gran Kino, que se merece lo menos tótem y medio por tantos años de amistad, compañía y ánimos.

Gracias a mis padres y al insuperable Dr. Maki, por estar siempre, siempre y siempre ahí.

Y gracias a Diana, que lo es todo para mí. Y, de todo, lo mejor.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives	2
1.3	Structure	3
1.4	Contributions	4
2	Scenario	7
2.1	What goes on in IP networks	7
2.1.1	The shift to a new environment	7
2.1.2	Delay and jitter	7
2.1.3	Data corruption and data loss	8
2.1.3.1	Channel impairments	8
2.1.3.2	Transmission channel behavior	9
2.2	The fight over video data transport: TCP vs UDP	9
2.3	Who controls the network?	11
2.4	The UDP/RTP combo	13
2.5	QoS management: providing video quality	13
2.5.1	Traffic management mechanisms	14
2.5.2	Error control mechanisms	14
2.5.2.1	Automatic Repeat Request (ARQ)	14
2.5.2.2	Forward Error Correction (FEC)	16
2.5.2.3	Hybrid Automatic Repeat Request (HARQ)	18
2.6	Channel modeling	18
2.6.1	Description of the behavior of the channel at upper layers	18
2.6.2	General model: finite-state Markov chain model	19
2.6.3	One-state model: memoryless, Bernoulli model or binary symmetric channel (BSC) model	21
2.6.4	Two-state models	21
2.6.5	Models for fixed-length bursts	24
2.7	Data relevance and unequal error protection (UEP) schemes	24
2.7.1	Encoded video data is not equally important	24
2.7.2	Beyond frame level	26
2.7.3	Error propagation and data importance	27

2.7.4	Distortion models and data prioritization	29
2.7.5	Real importance of data units	30
3	Protection scheme	33
3.1	Standard Pro-MPEG COP3 codes	33
3.1.1	Introduction	33
3.1.2	Encoding	34
3.1.3	Decoding	35
3.1.4	Complexity	37
3.1.5	Performance and motivation for code enhancement	37
3.1.6	Adaptation to the protection of two-layered video streams	38
3.1.6.1	Layered video stream	38
3.1.6.2	Layer-aware forward error protection	38
3.1.6.3	Description of the scenario	39
3.1.6.4	Description of the proposal	39
3.1.6.5	Evaluation of the adaptation	41
3.2	EEP enhancement of Pro-MPEG COP3 codes	41
3.2.1	Procedure	41
3.2.2	Complexity	43
3.2.3	Evaluation of the scheme	43
3.2.4	Adaptation to the protection of two-layered video streams	44
3.3	UEP enhancement of Pro-MPEG COP3 codes	45
3.3.1	Procedure	46
3.3.2	Proof of concept	47
3.3.3	Signaling	52
3.3.3.1	Mask signaling method	54
3.3.3.2	Explicit signaling method	54
3.3.3.3	Shared-load signaling method	55
3.3.3.4	Comparison of the signaling methods	56
3.4	Conclusions	58
4	Unequal error protection scheme optimization	61
4.1	Introduction	61
4.2	Problem description	63
4.2.1	Optimization problem formulation	63
4.2.2	Solution space of the subproblem with configurations of matrices	64
4.3	Proposed optimization procedure	65
4.3.1	Solution space reduction	66
4.3.2	Proposed hybrid simulated annealing metaheuristic	68
4.3.2.1	Description of the base methods	68
4.3.2.2	Optimization procedure	70
4.3.3	Summary	74
4.4	Experiments and results	75
4.5	Conclusions	83

5	Distortion model	85
5.1	Introduction	85
5.2	Distortion model	86
5.2.1	General description	86
5.2.2	Previous work	87
5.2.3	Selection of the measurement of distortion	88
5.3	Proposed distortion model	90
5.3.1	Motivation	90
5.3.2	Expected distortion introduced in a video sequence	90
5.3.3	Expected distortion introduced by a frame ($\bar{D}_f$)	91
5.3.3.1	Distortion introduced without considering the references ($D_{\text{indep},f}$)	94
5.3.3.2	Impact of a simultaneous errors in a reference frame	95
5.3.3.3	Formalization of the impact of the reference frames	97
5.3.3.4	Likelihood of occurrence of a group of combinations in a frame ($P(\mathbf{t}_f^{p_f, g_f})$)	99
5.3.3.5	Likelihood of occurrence of a group of combinations in the reference frames ($P(\mathbf{t}_{R_f}^{p_{\text{ref}}, g_{R_f}})$)	99
5.3.3.6	Example of use of the proposed distortion model	100
5.4	Experiments	102
5.4.1	Performance in terms of accuracy	102
5.4.2	Performance in terms of processing time	106
5.4.3	Performance when used jointly with UEP extension to the Pro- MPEG COP3 codes	107
5.5	Conclusions	110
6	Conclusions and future work	111
6.1	Conclusions	111
6.2	Future work	113
	Bibliography	115

Nomenclature

3D-HEVC	3D High Efficiency Video Coding
ABL	Average Burst Length
ACK	Acknowledgement
ACO	Ant Colony Optimization
AL-FEC	Application-Layer Forward Error Correction
ARQ	Automatic Repeat reQuest
AVC	H.264/Advance Video Coding (MPEG-4 - Part 10)
BER	Bit Error Rate
CDN	Content Delivery Network
CTU	Coding Tree Unit
DCCP	Datagram Congestion Control Protocol
DIBR	Depth-Image-Based Rendering
DiffServ	Differentiated Services
DSL	Digital Subscriber Loop
FAST TCP	Fast AQM (Active Queue Management) Scalable TCP
FEC	Forward Error Correction
FTV	Free viewpoint television
GA	Genetic Algorithm
GLS	Guided Local Search
GOP	Group Of Pictures
HARQ	Hybrid Automatic Repeat Request

HbbTV	Hybrid Broadcast Broadband TV
HEVC	H.265/High Efficiency Video Coding (MPEG-H - Part 2)
HTTP	HyperText Transfer Protocol
HVS	Human Vision System
IntServ	Integrated Services
IP	Internet Protocol
IPTV	Internet Protocol TV
LA-FEC	Layer-aware FEC
LCU	Largest Coding Unit
LDPC	Low-density parity-check
M2TS	MPEG-2 - Transport Stream
MB	Macroblock
MDS Codes	Maximum Distance Separable Codes
MPEG	Moving Picture Experts Group
MPEG-2	H.262/MPEG-2 Part 2
MSE	Mean Square Error
MV-HEVC	Multiview High Efficiency Video Coding
MVC	Multiview Video Coding
NAT	Network Address Translation
nD	n-dimensional
OTT	Over The Top
PEIN	Prolonged Electrical Impulse Noise
PLR	Packet Loss Rate
PSNR	Peak Signal-to-Noise Ratio
PSO	Particle Swarm Optimization
QoE	Quality of Experience
QoS	Quality of Service

CONTENTS

QP	Quantization Parameter
QUIC	Quick UDP Internet Connections
REIN	Repetitive Electrical Impulse Noise
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
RTT	Round-Trip Time
SA	Simulated Annealing
SCTP	Stream Control Transport Protocol
SHINE	Single High-level Impulse Noise
SHVC	Scalability extension of HEVC
SLA	Service-Level Agreement
SPD	Selective Packet Discard
SSIM	Structural SIMilarity
SVC	Scalable Video Coding
TCP	Transport Control Protocol
TS	Tabu Search
UDP	User Datagram Protocol
UEP	Unequal Error Protection
VoD	Video on Demand

Chapter 1

Introduction

1.1 Motivation

The world of video transmission has undergone a huge change in the recent years: an ever-increasing portion of services involving video distribution, previously delivered through long-established RF-based methods, are now supplied using the Internet Protocol (IP) [1] suite over packet-switched networks. According to Cisco, globally, 82 % of all consumer IP traffic will be video traffic by 2020, from up to 70 % in 2015 [2] and slightly over one-third in 2009 [3]. In absolute numbers, it will grow over three times in the following four-year period.

The reasons behind this paradigm shift are manifold, most of them related to key advantageous characteristics associated with these networks. Namely, they are almost ubiquitous and multiservice, and make multidirectional communications possible. These features enable: easier integration and synchronization with services of the same or different nature; interactivity with the user (selection of content and of the time of viewing, use of trick modes, etc); a greater level of personalization (profiling, insertion of ads, etc); infrastructure sharing and the consequent cost reduction; a certain degree of democratization of video production and distribution (providing some video-related services do not require huge investments any more), etc.

In some cases, the whole group of services provided under the same system (e.g. Internet Protocol TV (IPTV) [4]) are provided through IP networks. In others, only a subset of the services in a pack are delivered via IP, as a complement to the services supplied through traditional means, in what constitutes a hybrid solution to provide an enhanced user experience (e.g. second screen services through Hybrid Broadcast Broadband TV (HbbTV) [5]).

Within this expanding IP-based video transmission paradigm, two main approaches coexist: a server-driven one, usually built around IPTV services, which is typically backed by protocols UDP and RTP, and a client-driven one, referred to as OTT, that uses protocols TCP and HTTP. Each of them is conditioned by both inherent and external factors that may account for advantages or drawbacks when setting up a specific video streaming service. Server-driven systems are commonly designed to guarantee a minimum

level of performance, including image quality and latency. However, to accomplish this goal, there needs to be some degree of external intervention in the network, ranging from absolutely managing it to simply applying an upper-level control technique. On the other hand, client-driven strategies can more easily adapt to the behavior of the communication channel, usually unmanaged, and to terminal characteristics, but cannot provide a minimum, constant quality performance to users. Nevertheless, the frontiers between the two are vanishing and pure server- or client-driven schemes are increasingly rarer to find. The causes are both economical (the deployment of managed networks require huge investments) and technical (the continuous improvement of networks, and particularly of the last mile, the difficulty of operating a large number of clients with their own control logic, etc.).

To overcome the impairments caused during transmission (latency, information losses, and others), QoS techniques are commonly introduced. These mechanisms are far more developed in the case of server-driven setups, since, as already indicated, they are intended to provide a certain level of performance. In these cases, typical QoS strategies consist in application-level traffic management and error control mechanisms. Whereas the first group of techniques is almost exclusively applied in managed, private networks, as, among other things, they violate net neutrality, the second group is more versatile and can be finer-grained designed to better fit the particulars of a given service or stream and adapt to a specific communication channel.

In this thesis, we focus on server-driven, real-time video transmission scenarios, where error-correcting mechanisms are employed to enhance QoS. In these scenarios, the Pro-MPEG COP3 codes are very well appreciated and extensively used, due to their capability to cope with error bursts and to their extremely low complexity. However, their performance decreases as channel conditions grow less favorable. On the other hand, protection strategies that are aware of the unequal importance of the different elements that make up the video data stream (regarding the distortion that they are expected to introduce in the resulting video sequence) usually provide better performance, since they enable a more appropriate distribution of the available protection resources, which results in a reduction of the distortion caused by channel impairments. However, they are typically far more complex than the Pro-MPEG COP3 codes.

1.2 Objectives

In light of these elements, this thesis first aims at studying the inherent characteristics of the Pro-MPEG COP3 codes, with a special focus on their weaknesses when performing in transmission scenarios subject to strict conditions. This analysis is subsequently used to explore strategies for the enhancement of these codes. These strategies aim at improving their performance without compromising their simplicity and preserving their structural encoding and decoding methods. Moreover, they target a generalization and optimization of these codes, through, among other ways, enabling that they can be used to protect unequally the data packet stream.

Therefore, regarding the enhancement of the codes, the first objective of the thesis

is to delve into their internal procedures with the purpose of boosting existing working mechanisms. This task aims at increasing their capability to adapt to the specific conditions of the communication channel with minimal changes and increase their overall recovery capacity. The second objective of this work is to generalize the Pro-MPEG COP3 codes by creating a low-complexity framework in which the available protection resources can be unequally distributed among data packets depending on their relevance, as a means to better use those resources and reduce the resulting degradation of the sequence presented to final users. In particular, we target at a protection mechanism based on standard Pro-MPEG COP3 codes in which a sequence of video data packets is input into the protection module, these packets are grouped in accordance to their uneven relevance, and they are protected using unequal code rates. All of this without compromising the nature of the original codes and without using extra resources.

Next, given that the use of the previously described framework might lead to be able to consider a considerable number of feasible configurations per group of packets, the process of searching for suitable configurations might not be able to be carried out using brute-force algorithms, as this could exceed imposed time requirements. Therefore, another objective of this work is to design a procedure that enables the selection of near-optimal protection configurations whereas simultaneously complying with strict time requirements. This optimization method must consider and incorporate inherent characteristics of the scenario and of the utilized protection scheme.

Finally, regarding the distortion models that are employed to estimate the importance of the packets in the stream, we have detected that most of the available approaches do not completely meet the requirements of the considered scenario, either because they are not accurate enough, because they need access to the source content or because the time that it takes to process them exceeds real-time limitations. Thus, the last objective of the thesis is to design a distortion model that is accurate enough to make the most of the unequal protection scheme described before, that can be carried out by just using information that can be very easily fetched from the packet stream, and that does not take much time to process so as to comply with strict time conditions.

1.3 Structure

In addition to the present introduction, this thesis comprises another five chapters and an appendix whose contents are briefly outlined next.

In Chapter 2, we introduce the general scenario, where many ideas and concepts related to audiovisual content transmission are described in depth and the main actors of the system are identified and sketched. First, the main difficulties of delivering video over IP networks are described, jointly with the different transmission strategies that have been put into practice to make video information finally get throughout the transmission network to the client, represented by the transmission and upper-level protocols that are employed, and the implications of using these tools in terms of network capacity usage, latency, etc. Then, we present the main actors of the system. First, the principal QoS management strategies are outlined. Second, the behavior of communication channels is

analyzed, and the general Markov chain model is introduced, jointly with simple but the most useful, and so broadly utilized, particularizations of this model. Finally, we analyze the main characteristics of packeted encoded video streams, along with a description on how channel errors propagate throughout the stream depending on the location of the error and on the coding prediction structure, which utterly determines the video data interdependence within the stream and with that the level of importance of each element.

In Chapter 3, we propose two schemes based on the Pro-MPEG COP3 equal error protection codes. The first one represents an equal error protection extension to the standard codes, allowing the use of not only one or two but a third dimension of interleaving depth for channel error decorrelation. The second one is an unequal error protection strategy that organizes video data packets depending on their relevance. These packets are then protected using unequal code rates, so allowing higher recovery rates of more important video data packets. The benefits of both schemes are analyzed through respective analyses and proofs of concept in a variety of scenarios.

In Chapter 4, we propose a method for optimizing the selection of suitable configurations for the protection of video data when using the UEP scheme proposal presented in the previous chapter. This optimization strategy is based on the hybridization of two very well known metaheuristics: simulated annealing (SA) and tabu search (TS), whose basic procedures are modified to fit the characteristics of the considered scenario, and so find near-optimal configurations when strict time restrictions apply. The performance of the proposed method is analyzed in terms of time, resulting distortion error and other parameters.

In Chapter 5, we explore the characteristics of distortion models already present in the literature and the features that we judge of key importance for the considered scenario, both regarding the inherent characteristics of the model and of how that distortion should more suitably be measured. We then describe our full-context-aware, yet non-intrusive and straightforward approach, where the packet interdependence as a result of intra- and inter-frame/slice prediction is amply considered. The proposed packet-level distortion model is finally compared to other adapted models of dissimilar features.

Finally, Chapter 6 includes the conclusions to this thesis.

1.4 Contributions

The contents of this thesis have been partially published in several peer-reviewed international conferences and journals as detailed next:

- Protection scheme
 - The equal error protection (EEP) extension to the standard Pro-MPEG COP3 codes was presented in [6]. Moreover, it also includes the adaptation of both the standard and the EEP cases to the protection of two-layered video streams.
 - The unequal error protection (UEP) framework for the protection of any given sequence of unequally-important video data packets was published first in [7].

This paper was presented as a proof of concept of our approach, as a means to validate it and explore its potential benefits.

- In [8], we presented a collection of methods to enable properly signalize which data packets are protected by which repair packets for arbitrary data-to-repair-packet connection setups.
- Protection scheme optimization
 - A first version of the optimization procedure to select suitable configurations of the developed UEP strategy can be found in [9]. This initial version focused on the reduction of the solution space and introduced the varying scale neighborhood structures included in the final approach. A complete version of the proposed hybrid metaheuristic was presented in [10]. This final version includes all the modifications performed on the core procedures of simulated annealing and tabu search to consider imposed time restrictions and explore the solution space as desired.
- Distortion model and data prioritization
 - An early version of a complete unequal error protection strategy was initially outlined in [11], and was later on described in depth in [12]. This strategy employed a primal version of the subsequently significantly more deeply developed distortion model.
 - A first advanced version of an adaptive distortion model was published in [13]. The model presented in this paper is aware of the frame prediction structure but considers only to a limited extent the interdependence relations between data units.

Chapter 2

Scenario

2.1 What goes on in IP networks

2.1.1 The shift to a new environment

As mentioned in Chapter 1, the number of video transmission services provided through IP networks is rapidly increasing year by year. There exist good reasons for this paradigm shift: ubiquity, far easier service integration and synchronization (e.g. second-screen applications), the possibility for interactivity and many more. However, IP networks are not a particularly well-suited environment for video distribution.

Indeed, IP networks were designed to provide best-effort delivery: a connectionless communication mechanism by which single data packets are addressed and routed independently along the network, just considering the information they carry in that respect. This behavior translates into an unreliable transmission service, where data packets can be lost, or delivered corrupted, duplicated, out of order or out of time. Moreover, the overall latency can vary significantly from one data packet to another (jitter).

Why does all of this happen? The causes of IP network transmission unreliability are addressed next.

2.1.2 Delay and jitter

The time that it takes a data packet to travel across the network from the transmitter to the receiver, called end-to-end or network delay, is fairly variable, and potentially unlimited. It results from the sum of delays associated with several independent contributing sources:

- Transmission delay: time that takes a network node (sender or intermediate networking devices -routers or others-) to push a data packet onto a link.
- Propagation delay: time that takes one piece of information of a data packet to cover the distance between one node and the following one.

- Processing delay: time that takes a node to process the routing information of a data packet. It can also include error verification.
- Queuing delay: time spend by a data packet in a queue at a node while waiting for other frames to be transmitted.

A data packet goes across an integer number of links and through the same number minus one of nodes. The described delays vary depending on the link (propagation, transmission time), the node (processing, queuing times) and external causes (e.g. system failures, imposed traffic management mechanisms, etc.). So, the network delay of a data packet is computed as the sum of all the times associated with all the links and nodes traversed by that packet on its way from the sender to the receiver, where we assume that nodes use store-and-forward transmission, that is, that the whole packet is received and checked before starting to send it forward. So, it is very likely that the network delay of different data packets will differ, not only because delays may change over time, but also due to the fact that, with high probability, they will go over different paths (links and nodes). This situation may result in out-of-order or out-of-time packet delivery.

2.1.3 Data corruption and data loss

2.1.3.1 Channel impairments

There exist a wide number of phenomena that can affect negatively the data transmitted over IP networks. These events are of very different nature and take place at different abstraction layers. Thus, they impair the bistream differently. At a physical level, the data flow is strongly influenced by occurrences like linear and nonlinear distortion, random noise, impulse noise (e.g. Repetitive Electrical Impulse Noise -REIN-, Prolonged Electrical Impulse Noise -PEIN-, Single High-level Impulse Noise -SHINE- in Digital Subscriber Lines -DSL- [14, 15]), interferences from active and passive devices¹, fading (e.g. multipath-induced and shadow fading in wireless propagation media)... The result at this level is the one's complement of one or more bits in the stream, as shown in Fig. 2.1. On the other hand, at upper levels, events affect whole data packets. Those events basically involve the rejection of data packets at intermediate routers or at the receiver, due to network congestion², transmission timeouts (an excessively delayed packet in the end becomes a lost packet), or the detection of corrupted data packets (i.e., containing bit errors not corrected at lower levels³). The particularity of these errors is that the considered structure of information, the data packets, never appear corrupted: they appear either absolutely unaltered or lost. These type of errors are called erasures. An example is depicted in Fig. 2.2.

¹Such as poorly shielded household devices or appliances related to lightning activity.

²If a packet arrives at a node whose buffer is full, it is discarded.

³For instance, some types of impulse noise at DSL loops result in long bursts (up to tens of milliseconds) that can affect a large number of consecutive bits and even spread across several packets.

Figure 2.1: Example of channel errors (physical level)

Figure 2.2: Example of channel erasures (upper level)

2.1.3.2 Transmission channel behavior

As a consequence of the latter, the behavior of a communication channel regarding the impairment of data during the transmission process can be described differently regarding the abstraction level considered. Since the behavior of a communication channel is not deterministic but stochastic, this description is carried out resorting to statistical measures. So, at lower abstraction levels, this is usually done in terms of the varying signal-to-noise ratio or bit error rate (BER), obtained as a result of the analysis of the comparison between the transmitted bits and the ones received. At upper layers, however, this is typically done in terms of the varying packet loss rate (PLR), obtained as a result of the analysis of the received packets (no comparison is needed in this case, as it is known where the erasures are).

In this thesis, we focus on the elements of the transmission system belonging to upper abstraction layers. Therefore, we will only consider the behavior of the channel from the latter point of view.

2.2 The fight over video data transport: TCP vs UDP

With the purpose of making data transmission reliable, a transport layer protocol, Transport Control Protocol (TCP) [16, 17, 18], was incorporated as a complement to IP. TCP provides connection-oriented communication, reliable delivery, and flow and congestion control, which solves many of the problems stemmed from the internet layer: all the data packets sent from the transmitter are received unaltered and ordered. Nevertheless, TCP solves these problems at the expense of introducing others; mainly: increased transmission latency and jitter and throughput limitation. So, for those cases where the problems derived from the use of TCP were more damaging to the targets pursued by the application transmitting through the IP network than those inherent to that network, or they could be tackled differently in the upper layer, a new transport layer protocol was conceived: User Datagram Protocol (UDP) [19]. UDP is a very simple, connectionless, unreliable protocol that enables end-to-end message transportation with

a minimum of extra mechanism. UDP does not introduce any error, flow or congestion control procedures and therefore the problems derived from the best-effort approach of the internet layer remain. Although some additional transport layer protocols have been defined later on with the aim of joining advantages of TCP and UDP and reducing their drawbacks, like Datagram Congestion Control Protocol (DCCP) [20], Stream Control Transport Protocol (SCTP) [21], FAST AQM Scalable TCP (FAST TCP) [22, 23], or Quick UDP Internet Connections (QUIC) [24], TCP and UDP are overwhelmingly the two protocols more used nowadays for message delivery over packet-switched networks. Thus, we focus our analysis on the TCP/IP and UDP/IP protocol combos.

Both TCP and UDP have been profusely used to distribute video-related services. The selection of one or the other basically depends on the restrictions imposed by the service in question and the characteristics of the network, and how they match with the features of the two protocols:

- Regarding time and quality, services with no strict real-time requirements —e.g. video on demand (VoD), most broadcastings (often including live events)—, or none at all (video content downloading), can perform nicely with TCP, which is commonly used jointly with the application-level protocol Hypertext Transfer Protocol -HTTP- [25] or, in some technologies, with a proprietary streaming protocol running on top of it [26]. In this cases, a sufficiently large reception buffer virtually ensures the unaltered, continuous playback of video sequences, as it will be able to absorb delivery delay fluctuations (due to changing network conditions and necessary retransmissions), and variable TCP-managed send rates. For determining the size of the buffer, not only the network latency, but also the bitrate of the video to be transmitted must be considered, assuming that we have enough channel throughput and the user's terminal device is capable of properly handling the required bitrate. Provided a rather stable connection in a well-dimensioned network, if a large buffer is utilized, a target video quality can be guaranteed to a great extent, however at the expense of a perceptible delay (sometimes of up to tens of seconds).

Many video providers opt for a multi-bitrate representation of the encoded sequence, so that the system can select among the different bitrate options depending on the channel conditions and the buffer occupancy. This type of strategy is known as adaptive bitrate streaming. In most of the implementations used today (e.g. MPEG-DASH [27], Apple HTTP Live Streaming [28], Microsoft Smooth Streaming [29]), the resulting stream is segmented into a number of small independent chunks, so as to facilitate switches between different encodings and the use of smart control algorithms. By using this technique, the possibility of video freezes and rebuffering is reduced, whereas using smaller reception buffers, thus speeding up video loading times. Naturally, in these cases, the video delivered to the user will with high probability be of a changing quality, which might be inappropriate, if the service provider targets at supplying rather constant a video image quality level to a great number of users (e.g. IPTV).

However, if those services demand lower end-to-end latency —e.g. videoconferencing, live broadcasting of certain events, online gaming—, TCP is no longer a suitable option, and UDP is necessarily brought into play. To fulfill the requirements of the service, the reception buffer is now very short. However, if UDP is used alone, video image quality requirements will most probably not be met, due to data packet drops. In these cases, application layer protocols and external techniques are needed to ensure a target quality level.

In light of these analysis, we see that, if no other elements are considered, setting up a service in many cases means establishing a tradeoff between delay and quality.

- Regarding the possibility of setting end-to-end multicast connections for streaming live content, UDP is the only choice, as TCP only provides point-to-point connections. Multicasting can be extremely helpful, as it means saving a huge amount of bandwidth, but in reality it is only used in private, managed IP networks. In the public, non-managed internet, multicasting is still an open matter, due to technical and economic issues. There exist, nevertheless, strategies to lighten the amount of bandwidth used for live broadcasting, like the use of Content Delivery Networks (CDNs) or other techniques following similar principles: single unicast connections created from a central server to an edge one, from where in turn a number of unicast TCP or unicast/multicast UDP connections are set.
- Regarding communications for which no feedback channel is available, TCP is also not an option, as it is based on the transmission of ACK messages to acknowledge data packet transfers.
- Regarding network management, contrary to UDP, HTTP/TCP traffic easily bypasses firewalls, Network Address Translation (NATs) gateways and proxies. For this reason, TCP is usually preferred to UDP for networks the provider has no control over.

2.3 Who controls the network?

Besides controlling the service or set of services that are supplied, the service provider may also be in charge of managing the content that is offered, the network over which this content is delivered and the devices that are needed to display the received content⁴. The characteristics of the deployed video delivery ecosystem, which includes the selection of the transport protocol, are heavily conditioned by the degree of involvement of the operator in these elements, particularly regarding content distribution.

In that respect, until a few years ago, video streaming was only possible through expressly set-up networks. Traditional telecom service providers (e.g. AT&T, Deutsche Telekom, Telefónica) would make huge investments to deploy closed, proprietary, dedicated networks that could be tailored and managed to fit specific circumstances and

⁴For example, IPTV systems require the use of dedicated set-top boxes to access and decode the delivered video content.

requirements: number and type of users -for instance, premium vs non-premium-, target Quality of Service (QoS) and Experience (QoE), etc. That would result in fairly predictable, controllable, well-dimensioned, high-speed networks, in which video-related services hardly need to compete for resources with other type of services. Under these conditions, quality requirements regarding both mean time between errors and image quality can be more easily met than in pure best-effort delivery networks, as the starting error rate (channel's inherent PLR) is lower and the behavior of the channel is less variable, and so can be anticipated to a greater degree, making error and congestion control mechanisms more efficient and effective. This is the environment in which IPTV services are provided, as they usually require complying with strict quality conditions⁵. In this context, UDP is typically selected as the reference transport protocol for content broadcasting, since it allows to conform with latency restrictions, and implementing multicasting and bypassing networking devices (NATs, proxies...) is not a problem since the service provider owns the network. On the other hand, when it comes to providing VoD services in managed networks, both UDP and TCP represent suitable options, as they usually face less strict time conditions and are delivered through unicast connections.

However, thanks to the continuous improvement of networks, and particularly of the last mile, broadband connections making possible real video streaming (in contrast to previous video transmission strategies like progressive download or pseudostreaming) over the public Internet are now accessible to an increasing number of users. In this context, Over-the-top (OTT) video streaming strategies which are possible without the involvement of the service provider in the network control, have attracted increasing attention. Indeed, services providers like Netflix, Hulu, BBC iPlayer or Amazon Instant Video account today for some of the largest shares of overall download traffic. This technology presents some very appealing features from both technical and non-technical perspectives: adaptability to changing network and client conditions, reliable data delivery, economical deployment, a greater degree of democratization in terms of the possibility to start up new services, etc. However, they also present important drawbacks: the difficulty to provide appropriate QoS and QoE in relation to both image quality and latency simultaneously, and effectively implement multicasting [30]. Both pull-based (i.e., delivery driven by the client) OTT video services, increasingly based on the adaptive streaming paradigm, and push-based (that is, controlled by the server) mostly rely on the HTTP/TCP combo for transport.

Nevertheless, both strategies are converging little by little, as a variety of factors seem to indicate. Many traditional video providers do now offer OTT video content through their platforms, either as a quality-limited 'freemium' service associated to the IPTV system, as a means to provide VoD within their owned networks or through the public Internet, or covering some portion of the transmission chain⁶. Furthermore, the unmanaged network's capacity keeps increasing, multicasting in the open Internet will

⁵IPTV conditions, typically specified in terms of throughput, delay and jitter, error rate and service availability, are usually included in standard contracts called service-level agreements (SLA), that express the terms under which services are provided.

⁶For instance, multicasting over UDP is used to distribute video content to the edge of the CDNs and from this point is transferred via OTT strategies to final users.

likely be a reality at some point, and a majority of households in many countries will be able to access Internet through high-speed links in the near future.

However, for now (and the coming years), the main means to provide QoS regarding both image quality and latency is through the server-driven UDP-based paradigm. In this thesis, as mentioned in the introductory chapter, we focus on scenarios where video-related services are provided under strict delay restrictions (e.g. live event broadcasting or high-quality videoconferencing), and so the mentioned video delivery strategy is selected. In the following sections, then, the elements that characterize the selected mechanism are described.

2.4 The UDP/RTP combo

As already mentioned, due to UDP's simplicity (it only provides a checksum to verify data integrity, and source and destination port numbers), it is usually accompanied by an application layer protocol so as to provide meaningful information on the ongoing data transfer. Real-time Transport Protocol (RTP) [31], a protocol specifically designed for the end-to-end, real-time transfer of media, is typically employed to this task. RTP does not provide flow or error control, like TCP does, but facilitates them through sequence numbering and timestamping. Moreover, RTP is highly adaptable to the streaming scenario. There exist different profiles to be used in regard to the particular environment or class of applications (for the case under study, the most suitable is the RTP audio video profile (RTP/AVP) for real-time audio and video conferencing and communication applications [32]). Furthermore, the payload format field indicates the specific format of the data carried in the packet, which enables a finer-grained data management. For instance, within the RTP/AVP profile: MPEG1/MPEG2 and MPEG2-Transport Stream (M2TS) [33], H.264/AVC [34], H.265/HEVC [35]... Finally, it also allows the implementation of user-specific header extensions.

In addition, RTP does not usually work alone, but it is commonly used jointly with other two protocols: Real Time Streaming Protocol (RTSP) [36] and RTP Control Protocol (RTCP) [31]. RTSP is utilized to establish and control media sessions between end points and control streaming media servers. It is client-driven and typically uses TCP to ensure reliable control message delivery. RTCP provides non-intrusive quality monitoring of the RTP session it is associated to (providing information like the packet loss rate, jitter or round-trip time -RTT-), and control information, and is used for the synchronization of media streams. This information is partly obtained from RTP's sequence numbers and time stamps.

2.5 QoS management: providing video quality

Since neither UDP nor RTP provide any flow or error control mechanisms, they need to be explicitly included at the application level, so that target image quality is provided to final users [37]. First, traffic management techniques are introduced to control both the flows allowed into network links and the distribution of the available bandwidth among

the present flows. In this way, network congestion can be prevented to a greater degree and therefore the channel conditions under which the considered services are provided can be enhanced [38]. These tools are almost exclusively applied in managed, private networks, as they violate net neutrality. Second, error control mechanisms are included to deal with the errors (in this case, packet losses) that finally occur during the transmission of video. These techniques are described in the following subsections.

2.5.1 Traffic management mechanisms

Packet traffic is handled in regard to implemented application-aware (coarse-grained bandwidth management mechanisms) or flow-level (fine-grained) policies [39]. Main techniques include: traffic shaping and policing [40], differentiated services (DiffServ) [41], integrated services (IntServ) [42], and selective packet discard (SPD) [43].

2.5.2 Error control mechanisms

The main groups of techniques are listed and described next:

2.5.2.1 Automatic Repeat Request (ARQ)

- **Description**

The receiver sends ACK messages to the sender every time a data packet arrives to indicate its correct reception. The sender, for its part, sets a timer for each packet sent. If no ACK messages associated with the data packet in question arrives within a fixed timeout, the sender assumes that this packet has not arrived at the receiver and it needs to be retransmitted.

There exist three main basic instances of ARQ: Stop-and-wait ARQ, Go-Back-N ARQ (similar to that used by TCP), and Selective Repeat ARQ. These techniques, whose performance is outlined in Fig. 2.3, heavily differ in the operations made during a the timeout associated with a packet and after this timeout is exceeded, resulting in a dissimilar reception buffer size, bandwidth utilization and overall throughput.

- **Pros and cons**

The main advantage is that the extra bandwidth needed to put ARQ into practice is relatively low, particularly when the number of lost packets out of the total sent is not so high. In addition to the bandwidth required to acknowledge all packets that arrive correctly (including those received after being resent), the extra overhead due to ARQ performance includes all resent packets. Therefore, it is made up of a constant part, due to ACK messages, plus a variable portion that grows with the channel PLR.

An important drawback is the delay. In the same way as TCP, transmission reliability comes at the price of requiring a sufficiently large reception buffer that can absorb packet delivery delay fluctuations, and so virtually ensure continuous playback (unless network conditions significantly worsen). The required buffer size (of at least the equivalent to

Figure 2.3: ARQ instances

up to several RTTs) might be unacceptable for many real-time video data streaming applications. Still, some of these applications use ARQ-like implementations in which receivers also set timers to control timeouts that depend on the maximum delay that the application in question can accept. In addition, negative acknowledgment (NACK) messages are employed to save bandwidth. Nevertheless, contrary to standard ARQ implementations, these cannot guarantee reliable packet delivery.

Additionally, a reverse channel is always necessary when implementing this error control mechanism. Besides communications through one-way communication links, where these techniques cannot be used, they cope poorly when transmitting to a large number receivers in multicast, as each receiver has to be dealt with independently, since the packets that are lost differ in each case.

2.5.2.2 Forward Error Correction (FEC)

- **Description**

The FEC error control mechanism, also known as channel coding, works as follows:

1. At the sender side, a series of operations are performed on the original information (the original data is encoded) to add redundancy. The output of these operations is transmitted to the receiver.
2. The decoder at the receiver uses the received data to detect and subsequently correct transmission errors to the best of the code's capacity.

The original data might be part unaltered of the encoder output. In that case, the code is called systematic. In contrast, codes not directly embedding the original data in the encoded data stream are called non-systematic. Schematic examples of both types are shown in Fig. 2.4 and Fig. 2.5. Systematic codes are usually preferred over non-systematic codes for two main reasons:

1. Receivers do not need to perform any decoding operations for using successfully received original data.
2. Receivers not supporting channel coding can still work, obviously at the expense of not being able to enhance transmission reliability.

The data units that FEC codes use are sets of bits called symbols. Symbols can involve from individual bits to whole transmission packets. The characteristics and capacity of a FEC code, and even the possibility of applying it, highly depend on the symbol length and the abstraction layer to which they are intended. In this regard, FEC codes working with short symbols (typically octets) are usually employed at lower layers to deal with bit errors, whereas channel codes working with large symbols (whole transmission packets) are applied to cope with packet losses at upper layers. In this thesis, we work with and will always allude to FEC codes working with whole transmission packets at the application layer, usually referred to as Application-Layer FEC (AL-FEC) codes.

Figure 2.4: Systematic FEC code

Figure 2.5: Non-systematic FEC code

There exist two main types of FEC codes: block codes and convolutional codes. Both are characterized by parameters k and n , $n > k$, where k represents the number of original data symbols input to the encoder at a given time step, and n is the number of output symbols of the encoding process at that same time step. The principal difference between the two categories is that whereas for block codes the n symbols exclusively depend on the those k inputs, for convolutional codes the n symbols also depend on the m inputs previous to the k ones. Parameter m then represents the memory of the convolutional code. In both cases, $n - k$ is the redundancy added to the original data, and $r = k/n$ the FEC code rate. Although convolutional codes are asymptotically optimal with respect to parameter m , the complexity of the encoder and the decoder also exponentially increase with this parameter

• Pros and cons

FEC mechanisms do not require a reverse channel, and so they can be successfully implemented in scenarios where retransmissions perform badly or cannot even work (e.g. one-way or multicast communications).

The utilization of channel coding comes at the price of adding a constant extra bandwidth, even if no errors occur. In addition, FEC techniques cannot guarantee total reliability. They can, however, if properly designed considering the characteristics of the scenario and the inherent capacity of the code, increase this reliability to satisfy target quality criteria: overall average image quality, mean time between artifacts. . .

Finally, FEC adds a delay that equals the time dedicated to the encoding and decoding operations plus that of the receiver waiting for all the packets required to perform the

decoding algorithm at some point. Nevertheless, in the context of real-time applications, FEC codes are selected and adjusted to ensure that this delay is low enough to meet the latency restrictions imposed by the application. So, it is certainly significantly lower than the one introduced because of retransmissions

2.5.2.3 Hybrid Automatic Repeat Request (HARQ)

- **Description**

This option is a combination of the two error control techniques described above: FEC and ARQ. At the receiver, the FEC code is applied first to correct any packet losses that might have occurred. If for any reason, the code's capacity is exceeded and a given packet or set of packets cannot be recovered, ARQ is then applied to request the retransmission of the required packets to make the recovery operations work or directly to obtain the packets that cannot be rebuilt.

- **Pros and cons**

HARQ outperforms what the used FEC code and ARQ mechanism would do alone. On the one hand, when compared to the channel code, this technique provides total transmission reliability. On the other hand, if compared to pure ARQ, it requires a significantly lower number of retransmissions, which might result extremely helpful in scenarios with a large number of clients.

Despite these advantages, this techniques shares some of the drawbacks inherent to FEC and ARQ. Both a constant bandwidth due to redundancy plus a variable one due to retransmission is added. Furthermore, the overall delay increases, as it includes both that on account of retransmissions and the one due to FEC generation, and packet waiting and recovery.

2.6 Channel modeling

2.6.1 Description of the behavior of the channel at upper layers

Figure 2.6: Example of trace of lost packets

The description of the behavior of a channel is typically performed through the analysis of real traces of received packets, which basically are a series of binary numbers

indicating whether a packet is successfully received (on time and unaltered) or lost (not received, received out of time, or corrupted but not repaired in lower layers). The former is normally represented by “0” and the latter by “1”. Within a trace, different patterns can be distinguished. The characteristics and frequency of these patterns in a given network are used for describing the behavior and subsequently modeling that network. In particular, the following patterns can be identified:

- Gap: set of consecutive successfully received packets between two lost packets. A gap of n packets can be represented as 10^n1
- Cluster: set of consecutive lost packets between two successfully received packets. A cluster of n packets can be represented as 01^n0
- Error burst: set of consecutive packets starting and ending with lost packets where the local PLR is greater than a given threshold. An error burst can therefore include both successfully received and lost packets, i.e., a combination of gaps and clusters.

An example of a real trace is included in Fig. 2.6.

2.6.2 General model: finite-state Markov chain model

The behavior of the channel can then be modeled considering the way in which erasures occur, that is, how the PLR varies during the period of video transmission. With the designed models, this behavior can be predicted and simulated to a considerable extent, particularly in the long term. This knowledge is of great importance when selecting and dimensioning key elements in the communication system, or adjusting their parameters: source encoder (e.g. frame prediction structure, quantization parameter (QP), error resilient mechanism), channel encoder (e.g. code, scheme of protection), transmitter (packet sending arrangement, i.e., the fashion in which packets are delivered)...

The modeling of the channel’s behavior is generally carried out with finite-state Markov chains with several properties (expressed below). These models consist of N_S discrete states. State i , where $i \in \{1, \dots, N_S\}$, has stationary probability π_i , where $0 < \pi_i < 1$, and an associated error probability p_i , where $0 \leq p_i \leq 1$. The stationary probability of a state expresses the likelihood of being at any given moment in that state, if the previously visited states are unknown. The error probability in a state equals a given value of PLR. Moreover, the transition between states is expressed through a set of N_S^2 variables arranged in a transition matrix $\mathbf{P}$:

$$\mathbf{P} = \begin{bmatrix} p_{11} & \cdots & p_{1N_S} \\ \vdots & \ddots & \vdots \\ p_{N_S1} & \cdots & p_{N_SN_S} \end{bmatrix} \quad (2.1)$$

where p_{ij} , with $0 \leq p_{ij} \leq 1$, expresses the probability of moving from state i to state j (of course, $i, j \in \{1, \dots, N_S\}$), i.e., $p_{ij} = P(X_n = j \mid X_{n-1} = i)$, where X_t denotes the

Figure 2.7: Chain model: example with three states

state at time t . The addition of all the transition probabilities going out from any state must equal 1. That is:

$$\sum_{j=1}^{N_S} p_{ij} = 1, i \in \{1, \dots, N_S\} \quad (2.2)$$

An example with $N_S = 3$ is presented by means of a state diagram in Fig. 2.7.

Properties

The Markov chains considered to model the behavior of communication channel fulfill a number of properties, enumerated next:

1. They must fulfill the so-called Markov property: the probability of arriving at a certain state only depends on the immediately preceding state, and not on previously visited ones. In this way:

$$P(X_n = x_n \mid X_1 = x_1, \dots, X_{n-1} = x_{n-1}) = P(X_n = x_n \mid X_{n-1} = x_{n-1}) \quad (2.3)$$

where $x_1, \dots, x_n \in \{1, \dots, N_S\}$.

2. They are stationary: the distribution of probabilities does not change with time.
3. They are irreducible: it is possible to get to any state from any state after a number of transitions.
4. They are aperiodic: returning to a given state is not obligatorily after a fixed number of transitions.

The stationary probabilities are linked to the transition probabilities and can be obtained from them through solving the following linear system:

$$\pi_k = \sum_{i=1}^{N_S} \pi_i p_{ik}, k \in \{1, \dots, N_S\} \quad (2.4)$$

$$\sum_{i=1}^{N_S} \pi_i = 1 \quad (2.5)$$

In each state, errors are generated as independent events, regarding its error probability. After generating an event at a state, the system moves to a different state or stays in the same one, considering the transition probabilities. At this state, a new event is generated, and so on. The overall PLR is related with the error probability and the stationary probability of all states through:

$$\text{PLR} = \sum_{i=1}^{N_S} p_i \pi_i \quad (2.6)$$

The most utilized models to shape the behavior of wired networks are next described regarding the number of states considered.

2.6.3 One-state model: memoryless, Bernoulli model or binary symmetric channel (BSC) model

In this model, the simplest one, the PLR is assumed to be constant. Therefore, only one state is needed. Even though it might be accurate enough to describe how the channel behaves in the backbone of the network, where the transmission of different packets constitute independent events, it does not consider its behavior at the end of the line, where DSL and wireless links are likely employed, and errors are, with high probability, correlated.

2.6.4 Two-state models

Figure 2.8: Two-state model

These models consist of two states with dissimilar error probability. The notation usually employed presents one of the states as the good state (state G) and the other one as the bad state (B), as shown in Fig. 2.8. The error probability in state G is lower than the one in state B, $p_G < p_B$. Since it has more than one state and they have unequal error probabilities, it is able to capture the bursty nature of the network, when considered from end to end. Two-state models generate a geometrical distribution of the duration of good and bad phases, which can very well approximate the length of the error bursts in wired links.

The transition matrix is as follows:

$$\mathbf{P} = \begin{bmatrix} 1 - p_{GB} & p_{GB} \\ p_{BG} & 1 - p_{BG} \end{bmatrix} \quad (2.7)$$

The overall PLR, as previously mentioned related with the error probability and the stationary probability of states G and B, can be computed as:

$$\text{PLR} = p_G \pi_G + p_B \pi_B \quad (2.8)$$

where:

$$\pi_G = \frac{p_{BG}}{p_{BG} + p_{GB}} \quad (2.9)$$

$$\pi_B = \frac{p_{GB}}{p_{BG} + p_{GB}} \quad (2.10)$$

There are different models regarding the number of a-priori assumptions:

1. Gilbert model

This model, named after E. N. Gilbert, who proposed it in 1960 [44], considers that packets always arrive successfully in state G, that is, $p_G = 0$. Moreover, Gilbert suggested a procedure to derive the value of the remaining three independent parameters (p_B , p_{GB} and p_{BG}) from the likelihood of four different realizations within the real traces, as long as the observed traces are long enough: $P(X_n = 1)$, $P(X_n = 1 \mid X_{n-1} = 1)$, $P(X_n = 1 \mid X_{n-2} = 1, X_{n-1} = 0)$, and $P(X_n = 1 \mid X_{n-2} = 1, X_{n-1} = 1)$. First, three intermediate variables are defined in terms of these likelihoods:

$$a = P(X_n = 1) \quad (2.11)$$

$$b = P(X_n = 1 \mid X_{n-1} = 1) \quad (2.12)$$

$$c = \frac{P(X_n = 1 \mid X_{n-2} = 1, X_{n-1} = 1)}{P(X_n = 1 \mid X_{n-2} = 1, X_{n-1} = 0) + P(X_n = 1 \mid X_{n-2} = 1, X_{n-1} = 1)} \quad (2.13)$$

The value of the three parameters is then obtained as follows:

$$p_{BG} = 1 - \frac{ac - b^2}{2ac - b(a + c)} \quad (2.14)$$

$$p_B = \frac{b}{1 - p_{BG}} \quad (2.15)$$

$$p_{GB} = \frac{ap_{BG}}{p_B - a} \quad (2.16)$$

There also exist a simplified version of the Gilbert model, in which the events in state B are generated with a fixed probability, usually $p_B = 1$ (all packets are assumed to be lost, and so error bursts and clusters coincide), or $p_B = 0.5$. The value of the remaining parameters that characterize this model can be obtained observing all restrictions by means of much easier expressions as the ones presented above. First, the overall PLR and the average burst length (ABL) need to be computed. The wanted values are derived from the value of the latter parameters through:

$$p_{BG} = \frac{1}{ABL} \quad (2.17)$$

$$p_{GB} = PLR \frac{p_{BG}}{p_B - PLR} \quad (2.18)$$

Its simplicity and good performance make it one of the most employed models to model communication channels with memory, especially for wired networks. In addition, the overall PLR and the ABL are quite frequently the only statistics available regarding the behavior of the channel, usually obtained through a backward channel, for instance by means of RTCP packets. Therefore, the simplified Gilbert model often constitutes the best approximation possible to the real behavior of the communication channel.

2. Gilbert-Elliott model

This model, proposed by Elliott in 1963 [45], is a generalization of Gilbert's proposal. In this case, no a-priori limitations operate. In this model, packets can be lost in both states. However, assuming a low p_G , errors in state G can be seen as independent. This way, a wider number of phenomena are contemplated in the model. Those leading to independent loses, like those that take place in the network's backbone, are considered in state G, whereas those leading to highly correlated errors are considered in state B. This model, however, constitutes a more complex model and the determination of the parameters' value a much more complicated task.

2.6.5 Models for fixed-length bursts

Figure 2.9: Fixed-length burst model

These models consist of $L + 1$ states, where L is the length of error bursts in the channel. This is depicted in Fig. 2.9. One of the states is characterized as “good”, whereas the rest are bad states. As in previous cases, the stationary probability of the good state is far lower than that of the bad states. Furthermore, they typically consider that no packets are successfully received during error bursts, as they are intended to simulate phenomena that usually take place in last mile links, specifically xDSL’s impulse noise, that lead to losing a number of consecutive packets [14, 15, 46]. In that case: $p_{B,i} = 1$, $i \in \{1, \dots, L\}$.

2.7 Data relevance and unequal error protection (UEP) schemes

2.7.1 Encoded video data is not equally important

The encoding of video is a process by means of which a raw video sequence is converted into a compressed format that uses a lower number of bits than the original representation. This process is carried out through exploiting sequence spatial and/or temporal correlation: basically, when a group of pixels in a frame is found sufficiently similar to an already inspected (and encoded) one (in the same frame -spatial redundancy- or in previous or subsequent ones -temporal redundancy-), it is not represented in the new format by means of the original values of its pixels, but through indicating the group of pixels it is similar to (by means of a motion vector) and the difference between the values of the pixels of the two groups (known as residual or prediction error). So, the group of pixels to be encoded uses the previously inspected one as reference and depends on it for the subsequent decoding process. This procedure is called differential coding and is performed throughout the whole sequence. Thus, most parts of the resulting bitstream are heavily dependent on other parts. The advantage of this process mainly is the number of bits saved to represent a piece of information. The main drawback is the vulnerability that this process creates: an error (modification from the originally encoded information or loss) in the reference information might lead to incorrectly decoding all the data that depends on it.

Traditionally, video sequences are encoded according to previously established prediction/coding structures. These structures specify the reference frames of every frame in a set of consecutive frames called group of pictures (GOPs). Frames are of different types depending on their references, as described next⁷:

- **I-frames (intra-coded frames)**: they are encoded only using information belonging to themselves. Hence, they exploit spatial redundancy, but not temporal. Their encoding and decoding processes do not rely on that of other pictures in the sequence. Since they only exploit spatial redundancy, they usually require more bits to encode than other frame types.
- **P-frames (predicted frames)**: they are encoded using information belonging to previous frames and, depending on the standard, also to themselves. Therefore, they exploit temporal correlation and, in some standards, also spatial redundancy when needed. They typically require less bits than I-frames to encode.
- **B-frames (bi-directional predicted frames)**: they are encoded employing information belonging to previous and subsequent frames and, depending on the standard, also to themselves. So, in the same way as P-frames, they exploit temporal redundancy and, in some standards, also spatial correlation when required. As they can refer to a greater number of frames for inter prediction, they usually require less bits than P-frames to encode.

Figure 2.10 shows two examples of prediction structures. The first one (Fig. 2.10a) corresponds to a coding structure where all the references of the frames in the GOP are also contained in the GOP. This is a closed GOP. In the second one (Fig. 2.10b), some of the references of the frames in the GOP are in a different GOP. This is an open GOP.

Standards prior to H.264/MPEG-4 Part 10 (Advance Video Coding -AVC), such as H.262/MPEG-2 Part 2 (MPEG-2) or H.263, limited the number of references of P-frames to only one and those of B-frames to just two, a preceding one and a subsequent one. Furthermore, B-frames could not be used as reference to other frames.

The arrival of standard AVC, however, made some of these concepts more vague. For instance, prediction was not any longer made necessarily at frame level but internally, at slice level. One frame is composed of one or more slices, each of which is encoded independently potentially using different references. They thus can be of different type (I, P, B...), in accordance to the descriptions above, plus some new types, called switching slices. Furthermore, B-frames started to be able to be employed as reference to other frames through the use of hierarchical picture structures, so enabling temporal scalability.

For both systems, frame- and slice-based, the basic processing unit has been, up to AVC, the macroblock (MB). That is, a frame/slice consists of a number of MBs. A

⁷For the sake of clarity and simplicity, in this document, we use the words 'frame' and 'picture' without distinction. That is, we assume that progressive video is used. However, a frame might be composed of not only one, but two pictures if interlaced video is used. In this case, pictures are known as fields. Most of the descriptions, models and techniques that are based on frames presented in this document can be translated into fields if this is required by the specifics of the scenario.

Figure 2.10: Examples of GOPs

MB is a rectangular set of pixels (typically 16x16, which in the most common chroma subsampling format, 4:2:0, is in turn made up of 16x16 samples of the luma component and 8x8 samples of each of the two chroma components) that can be further divided into transform blocks and prediction blocks. The former ones are encoded with respect to information in MBs of the same frame/slice and so exploit spatial redundancy. The latter are encoded using as reference MBs in other frames or slices in other frames and so exploit temporal correlation. In the newest video compression standard, H.265/MPEG-H Part 2 (High Efficiency Video Coding -HEVC), MBs has been replaced by coding tree units (CTUs), also known as largest coding units (LCUs). CTUs are basically a generalization of MBs with the idea of obtaining greater video compression rates.

2.7.2 Beyond frame level

In addition to traditional video coding schemes, new formats, extensions and tools have been introduced to face a variety of scenarios and types of content, creating new ways of interdependency besides inter- and intra-frame/slice. For instance, layered video coding (e.g. Scalable Video Coding -SVC- or -SHVC) provides support for different types of scalability: temporal (frame rate), spatial (frame resolution), quality (same resolution but at different image qualities) or color gamut (color space). To that end, it enables the encoding of video in a number of inter-dependent layers: a base layer and additional (one or several) enhancement layers above it, each of them transmitted in a separate substream. The base layer is the only one that can be decoded independently. The result of decoding this layer is a video sequence of a limited quality. Each enhancement layer depends on the lower ones for decoding. Decoding these layers in the precise order in regard to these dependencies, beginning with the one just above the base layer, will increase the quality of the resulting video sequence, up to a similar one to that that can be achieved using non-layered video coding [47, 48]. This coding scheme is very helpful to provide video-related services to sets of users with devices of potentially extremely

Figure 2.11: Example of error propagation at frame level in a closed GOP. The red star indicates the frame where the error has taken place and the blue ones the frames affected by this error, besides the one where the error occurs.

different capabilities (from handheld appliances to 4K resolution TV sets) connected through a large variety of access networks. Each client will be able to display video sequences of a quality equal to the one that results from decoding the base layer plus a number of enhancement layers, depending on its capabilities.

Additionally, in those cases related to three-dimensional video and free viewpoint television (FTV), both regarding the multiview (e.g. Multiview Video Coding -MVC- or Multiview High Efficiency Video Coding -MV-HEVC) and the Depth-Image-Based Rendering (DIBR) (e.g. 3D High Efficiency Video Coding -) paradigms, frames/slices in one view can be encoded not only referring to frames/slices in the same view, but also to frames/slices in other ones, either texture or depth images. In this fashion, spatial redundancy can be further exploited, as the views captured by neighboring cameras are in many cases very similar.

2.7.3 Error propagation and data importance

As mentioned before, a simple bit error or an isolated packet erasure in the information that is used as reference by some encoded data might lead to incorrectly decoding both the reference information and the dependent one. But not only that: all the data that in turn use the latter one as reference will also be affected by the original loss, and so on. Hence, errors are said to propagate through the bitstream, potentially affecting large portions of the sequence, both spatially and temporally. Naturally, the propagation of errors takes place at the level of basic encoding units (MBs, CTUs...), as this is where intra- and inter-prediction actually happens. However, it is reflected at coarser-grained encoding levels, which include these basic units. So, for instance, in traditional one-layered video sequences, an error in data that belongs to a reference frame might impact all the frames inside and outside the GOP that were encoded employing that data as reference. For example, as shown in Fig. 2.11, in a video sequence encoded following the prediction structure in Fig. 2.10a, an error in the first P-frame might affect all video frames in the GOP, except for the I-frame, whose encoded data does not depend on any information belonging to the P-frame. This effect can be seen in Fig. 2.12, which shows the visual artifacts provoked by the loss of only one data packet. In the first example (Seq. #2), the lost packet belongs to an I-frames, whereas in the second one (Seq. #3), the lost packet carries informations of a P-frame. Regarding layered video, an error in the information belonging to one layer might impact not only in the layer in question, but

Figure 2.12: Seq. #1: Original sequence; Seq. #2: Sequence corrupted because a packet carrying I-frame information was lost; Seq. #3: Sequence corrupted because a packet carrying P-frame information was lost.

in all the layers above it. This effect occurs in all encoding schemes at the levels (data structures) through which they can be characterized: MBs/CTUs, frames/slices, layers, views..., which states that data structures in the bitstream are of different importance regarding the scope and degree of degradation that their corruption/loss might cause in the resulting decoded content presented to users. For instance, considering only the frame level, the number of frames affected by an error in the information related to each frame, and coding structures like the one presented in Fig. 2.10a, the I-frame is more relevant than any P-frames, whose importance in turn relies on the position they hold along the GOP, but anyhow are more important than the B-frames.

2.7.4 Distortion models and data prioritization

QoS control mechanisms in transmission systems can greatly benefit from the unequal treatment of data depending on the importance of the transported content, not only regarding the traffic class (e.g. by prioritizing video streaming services over email services), but the elements within a class [49]. Indeed, regarding traffic control, data prioritization strategies allow for intra-class management of data packets. This can be used, for instance, to implement finer-degree DiffServ or SPD strategies, respectively by creating classes according to the relevance of video data packets and providing packet-forwarding properties according to these classes (e.g. low-latency transmission for important classes and best-effort delivery for less important ones) [50], or by discarding less important packets in case of network congestion [51]. Regarding error control mechanisms, ARQ strategies can save bandwidth by only retransmitting important data packets, or opt for a multiple retransmission scheme in which more relevant packets are retransmitted a number of times to increase the probability of on-time arrival and less relevant packets are not retransmitted [52]. Finally, FEC-based protection schemes can improve the delivered image quality by distributing protection resources according to the importance of data: the more important a video data packet is, the more resources are devoted to protect it. In this way, more important data packets are more likely to be recovered in case of loss [53]. This strategy is known as unequal error protection (UEP).

As aforesaid, the loss of different data units of the bitstream affect unequally the sequence finally decoded and presented to the user. Therefore, a precise knowledge of the degradation caused by these errors is key to determine the real importance of each unit and so boost the effectiveness of the selected QoS control technique. This process requires simulating the error or combination of errors under analysis, decoding the impacted sequence and measuring the consequential quality degradation. This assessment can be carried out through a full-reference (FR), a reduced-reference (RR) or a no-reference (NR) technique, according to whether the original (reference) images are used entirely, partly or not utilized at all. Moreover, this process can be performed according to methods that move from complete objectiveness (e.g. mean square error -MSE) to total subjectiveness (e.g. rates given by viewers based on their perception), depending on how much they take into consideration the human vision system (HVS). However, in either case, obtaining the exact distortion that all the possible combinations of errors will cause and so associate an exact distortion value to each data unit in the stream is a vast task,

almost unmanageable in practice. The reason is manifold. First, the number of possible combinations of errors that affect interdependent data units increases enormously with the amount of units considered. Second, the process of computing the resulting distortion is usually very time consuming. And third, there still exist imponderables that depend on different aspects of the transmission chain, like the behavior of decoder when facing information starvation or the error concealment technique applied to cope with data losses.

Thus, the relevance of the different data units that make up the encoded video stream is typically not directly computed but estimated by means of a distortion model. Basically, such a model relates key features that characterize the video data units at a particular level to the distortion that will be potentially introduced in the decoded sequence if these elements are not available at the decoder. In addition to the features of the data unit in question, it is also important to incorporate information about its context. For instance, it is useful to consider the prediction structure used to encode that unit. This is illustrated through the following example: two data units transporting the same information belonging to an I-frame are not equally important if different GOP lengths are employed, as the impact of losing that information is of different duration. So, the unit whose loss provokes the more-lasting effect is more important. Furthermore, the likelihood of losing the data unit, or, more generally, the behavior of the communication channel, is also relevant to estimate the importance of data units, and therefore, to be included in the considered distortion model.

2.7.5 Real importance of data units

The real importance of a data unit does not only rely on the relevance of the information it carries (measured as the distortion that it would be introduced if it were lost singly), but on its actual contribution to the overall degradation. That means that in order to properly assess the importance of a data packet, it is not enough to know the impact of its individual loss and the likelihood of occurrence of this loss. It is also necessary to know the likelihood of losing the other interdependent units and how the loss of these other packets modulates the actual impact of the loss of the packet in question. The reason is that there exists a correlation factor that considers the effect on the overall distortion of potentially simultaneously losing interdependent data packets. This factor can be positive, negative or zero, depending on the position that those lost packets hold in the packet stream and the relation of dependency between the frames/slices they belong to. For instance, the distortion that is introduced if two packets carrying information of the same frame/slice are lost is lower than that of adding up the distortion that results from losing them individually. The reason is that a masking effect resulting from the loss of synchronization occurs in the first place, and so the effect of the loss of the second packet in the first case does not equal that of its individual loss: a negative correlation takes place.

Modeling the actual contribution of each data packet in the stream with respect to that of the other interdependent packets is then a key task that strongly depends on the encoding and transmission scenarios, as these elements utterly determine the

interdependences between data units. This job is tackled very differently in the different approaches present in the literature, as will be seen in Chapter 5, and represents a clear trade-off between simplicity and accuracy.

Chapter 3

Protection scheme

3.1 Standard Pro-MPEG COP3 codes

3.1.1 Introduction

This AL-FEC codes were introduced by the Pro-MPEG Forum in its Code of Practice 3 r2 [54] (Pro-MPEG COP3 codes), and were later on standardized by the SMPTE in its specification 2022-1 [55]. Since then, they have been included in a number of standardization documents, including the DVB standard for IPTV (ETSI TS 102 034 [4]) and documents of various organizations: ATIS, ETSI, ITU-T, the Open IPTV Forum (who transferred its technical activities to the HbbTV Association in 2014) [56]... They have been widely deployed, both alone and jointly with a second code, due to their very appealing features [4]. Especially, their low complexity and their capability to cope with burst errors are very much appreciated in real-time streaming over packet-switch networks.

Figure 3.1: Standard Pro-MPEG COP3 codes repair packet generation. Example with four rows ($D = 4$) and five columns ($L = 5$).

3.1.2 Encoding

In the standard Pro-MPEG COP3 codes, data packets are first organized in matrices of D rows and L columns. Packets are arranged in sequence number order, filling the matrix row by row, from the column most to the left to the one most to the right. Protection packets can then be generated both row-wise and column-wise by XORing the associated data packets, as illustrated in Fig. 3.1. The parity packets generated row-wise are suited to deal with independent packet losses, and column-wise-created FEC packets are intended to cope with burst errors, as they are basically the result of applying an interleaving step prior to performing the XOR operations. So, the number of data packets per FEC block, k , equals $D \cdot L$, whereas the number of repair packets depends on whether one or both dimensions are employed: D repair packets if only the row dimension is used, L repair packets if only the column dimension is employed, and $D + L$ repair packets if both dimensions are used.

Naturally, the code rate of the codes, r_{FEC} , and depends on the selected configuration, that is, on the values of parameters D and L , and on which dimensions are finally used. So, the code rate for the three possible options is:

$$r_{\text{FEC}} = \begin{cases} L/(L + 1) & \text{if only rows are used} \\ D/(D + 1) & \text{if only columns are used} \\ D \cdot L/(D \cdot L + D + L) & \text{if both dimensions are used} \end{cases}$$

The recovery capability of the codes heavily depends on the selected protection configuration. Nevertheless, the set of configurations that can be used is restricted by the characteristics of the scenario. Particularly, the range of values that parameters D and L can take is limited by: (i) the maximum tolerable latency, jointly with the packet transmission arrangement, and the video bitrate, which determine the maximum value of k , (ii) the minimum allowed FEC code rate, which determines the available FEC bandwidth, that is, the maximum number of parity packets that can be generated and transmitted, and (iii) the restrictions imposed by the standardizing document (in terms of block size and number of columns) or the range of values supported by the receivers that implement the solution [54, 57]. The limitations contained in the standardizing documents have nevertheless evolved over time with the characteristics of video transmissions, and should not represent an obstacle for deployment [4]. The reason for these changes is twofold: greater flow bitrates allow handling bigger protection blocks in the same period of time, and error sources might cause longer error bursts, then requiring matrices with more columns. Furthermore, some documents explicitly mention that this clause shall not apply and allow appliances to support values outside the indicated range [4]. Finally, many times, the values of D and L are limited externally, e.g. through the established signaling procedure [58]. Considering these conditions, the actual values that D and L take, and the decision on which dimensions to employ, are often selected after an optimization process that depends on the channel behavior.

Figure 3.2: Example of column-wise data packet recovery.

Figure 3.3: Example of row-wise data packet recovery.

Figure 3.4: Example of column-row-column data packet recovery.

3.1.3 Decoding

The procedure to recover lost packets depends on which dimensions are used to generate FEC packets. If repair packets are generated only column-wise, the recovery algorithm loops through the columns of the matrix, selecting those where only one data packet is lost and the repair packet has been received. The available data and repair packets of these columns are XORed, resulting in the reconstruction of the lost packet. An example of the recovery of lost data packets provided this configuration can be seen in Fig. 3.2. On the other hand, if repair packets are generated only row-wise, the same procedure takes place, but performed on rows. An example of lost data packet recovery under this configuration is depicted in Fig. 3.3. Finally, if repair packets are generated both column- and row-wise, the algorithm performs a maximum of three loops. First, it loops through one of the dimensions. Second, through the other. And third, again through the first one. In every loop, the corresponding XOR operations are carried out at each column or row containing only one lost packet. Examples showing the recovery of a combination of lost data packets are provided in Fig. 3.4 and Fig. 3.5. In the example in Fig. 3.4, the algorithm starts up looping through the columns, whereas in the example in Fig. 3.5, it first deals with the rows.

Figure 3.5: Example of row-column-row data packet recovery.

Figure 3.6: Examples of code recovery failure.

The reason to select and perform XOR operations only at columns and rows containing only one lost packet is explained next. At each column, the code performs like an ideal channel code of parameters $(D + 1, D)$. In the same way, at each row, the code performs like an ideal channel code of parameters $(L + 1, L)$. That means that at each column or row, if only one data packet is lost (and the corresponding repair packet is available), the algorithm will always be able to recover it. However, if two or more packets (data or repair) are lost, the code will not manage to recover them. The configuration through which parity packets are generated in both dimensions enables the algorithm to swing between them so as to find rows and columns where only one packet is lost and rebuild it. By doing so, more rows and columns that contained two or more lost packet at first will contain only one lost packet afterwards, thus enabling its recovery.

Therefore, the combinations of lost packets where the standard Pro-MPEG COP3 codes fail to recover all lost data packets are those for which there exists no way to accomplish that all packet losses are the only missing ones in the column or row at some point of the algorithm. If repair packets are only generated column-wise, this situation takes place if at least one of the columns contains more than one missing packet. In the case of the configurations where parity packets are only created row-wise, this situation occurs when at least one row contains two or more lost packets. If a configuration using

both dimensions is selected, this condition happens when at least two columns contain two or more missing packets that are in turn part of the same rows. These situations are respectively shown in Fig. 3.6.

These situations are likely to happen, for the first and third type of configurations, if the number of columns is not high enough so as to properly deal with error bursts, or if the likelihood of occurrence of two or more error bursts affecting the same matrix is too high. In these cases, there are too many columns with two or more missing packets, thus lost packets cannot be recovered. In the case of the second type of configuration, the occurrence of any error burst of two or more packets suffices to overcome the codes. The second configuration is then, as mentioned before, only suitable against isolated packet losses, i.e., uncorrelated errors that are rare enough so as not to affect the same rows.

In the considered networks, the channel is essentially bursty, which means that configurations where FEC packets are only created row-wise are of little use, as the likelihood of errors affecting two or more packets in the same row is very high. Between the other two types of configurations, even though the configurations where the two dimensions are employed perform better than the ones where repair packets are only created column-wise, the latter are usually selected [4]. The reason is that in channels in which errors occur in bursts, the extra recovery capability that brings also using row-wise-generated packets does not generally make up for the considerable bitrate increase that this option entails.

3.1.4 Complexity

Pro-MPEG COP3 codes are characterized by its extremely low complexity with respect to both the redundancy generation algorithm and the encoding and decoding processes (computational complexity). Unlike other state-of-the-art codes, like Raptor or RaptorQ codes [59], redundancy is generated through the straightforward one-step process described above, which very much eases their implementation and deployment. Concerning this process, the encoding computational complexity equals $L - 1$ symbol, i.e. packet-level, XOR operations per row-wise repair packet and $D - 1$ symbol XOR operations per column-wise repair packet. Thus, the encoding complexity deeply depends on the shape of the matrix, growing linearly with k in the worst case. In the decoding process, always less than $D \cdot (L - 1) + L \cdot (D - 1)$ symbol XOR operations are needed to recover all the lost packets in a protection block in the worst case, if both dimensions are used. Therefore, the decoding computational complexity is also linear with k in the worst case. If, instead of symbol operations, we consider bit XOR operations, all the referred number of operations are to be multiplied by the length of the longest data packet involved.

3.1.5 Performance and motivation for code enhancement

The performance of the Pro-MPEG COP3 codes has been conscientiously evaluated for a variety of scenarios in different documents [60, 57]. This analysis shows that they work very satisfactorily in terms of the necessary redundancy to reach a target PLR from the inherent loss ratio of the channel, as long as the latter is sufficiently low. However, they

become less efficient as network conditions turn more severe, which might lead to an inadequate protection of the video stream and so to intolerable levels of degradation of the video presented to final users.

The previous conclusions have motivated the search for enhancements to the standard Pro-MPEG COP3 codes, that could increase their efficiency at high channel PLR, whereas preserving the principles of their redundancy generation algorithm in terms of procedure and complexity.

3.1.6 Adaptation to the protection of two-layered video streams

In this section, we discuss the introduction of a Layer-aware FEC (LA-FEC) scheme [48] based on the Pro-MPEG COP3 codes to protect two-layered multimedia streams.

3.1.6.1 Layered video stream

A layered video stream is the result of applying layered video coding to encode a video sequence. This video stream contains a number of inter-dependent layers, a base layer and additional (one or several) enhancement layers above it, each of them transmitted in a separate substream. The base layer is the only one that can be independently decoded. The result of decoding this layer is a video sequence of a limited quality. Each enhancement layer depends on the lower ones for decoding. Decoding these layers in the precise order in regard to these dependencies, beginning with the one just above the base layer, will increase the quality of the resulting video sequence, up to a similar one to that that can be achieved using non-layered video coding [47, 48]. Layered video coding can provide different types of scalability: temporal (frame rate), spatial (frame resolution), and quality scalability (same resolution but at different image qualities).

This coding scheme is very helpful to provide video-related services to sets of users with devices of potentially extremely different capabilities (from handheld appliances to 4K resolution TV sets) connected through a large variety of access networks. Each client will be able to display video sequences of a quality equal to the one that results from decoding the base layer plus a number of enhancement layers, depending on its capabilities.

3.1.6.2 Layer-aware forward error protection

In contrast to traditional UEP strategies, where protection packets are generated independently for each layer, LA-FEC schemes follow the dependency between the different video layers to improve channel decoding capability [48].

In addition to the data packets belonging to a given enhancement layer, LA-FEC schemes use packets of lower layers to generate the repair packets associated to that enhancement layer. In this way, the probability to rebuild lost packets that belong to lower, and thus, more important layers, increases, as this procedure creates more ways to perform this recovery. Furthermore, this method does not affect the enhancement layer in question. There might be situations where lost packets belonging to that enhancement

layer cannot be rebuilt because of being tied to non-recoverable lost packets of lower layers, due to exceeding the code's recovery capability. If a traditional UEP scheme is used instead, the lost data packets of that enhancement layer might be able to be recovered. However, they will turn useless if the data packets belonging to lower layers cannot be recovered, as they depend on them. Therefore, both types of situations result in the same video quality. Nevertheless, in the majority of possible combinations of received and lost packets, the use of LA-FEC increases the probability of recovering lower-layer data packets, thus reducing that of having available but useless data packets of upper layers. Hence, all layers see a benefit in using LA-FEC.

3.1.6.3 Description of the scenario

In the considered scenario, two layers are employed to encode video sequences: the base layer and one enhancement layer. We also assume that users are divided into two different subsets, regarding network and device capabilities. Users in the first group only receive the base layer of the layered video stream with an associated quality Q_1 . The second group obtains both layers with an associated quality $Q_2 > Q_1$. In order to guarantee the target quality, the layered video is protected by means of an LA-FEC scheme.

3.1.6.4 Description of the proposal

Figure 3.7: Example of packet arrangement of the proposed LA-FEC scheme for two-layered multimedia protection. Purple is for base layer packets and pale blue for enhancement layer packets.

Our proposal is a straightforward scheme that operates as follows. First, the base layer is protected on its own applying the Pro-MPEG COP3 codes. The repair packets created cannot depend on enhancement layer data, as this information is distributed to both sets of users. The redundancy generated guarantees the aimed Q_1 . Next, enhancement layer protection is created. To this end, two steps take place. In the first one, the code parameters are optimized using only packets belonging to the enhancement layer. In the second one, the base layer video packets are uniformly distributed among the different dimensions of the enhancement layer. Finally, repair packets are generated XORing not

 (a) (Bitrate enh. layer / bitrate base layer)
= 0.45

 (b) (Bitrate enh. layer / bitrate base layer)
= 0.35

 (c) (Bitrate enh. layer / bitrate base layer)
= 0.25

 (d) (Bitrate enh. layer / bitrate base layer)
= 0.15

Figure 3.8: Evaluation of the performance of the proposed LA-FEC scheme with standard and enhanced Pro-MPEG COP3 codes. Parameters: FEC latency = 400 ms; Channel PLR = 10^{-2} ; Error distribution = REIN model (8 ms); Bitrate base layer = 4 Mbps.

only the corresponding enhancement layer packets, but also the associated base layer packets. So, the decoding capabilities of both layers are increased and quality Q_2 can be obtained. Figure 3.7 shows an example of the application of the proposed LA-FEC scheme. As for the one-layer video case, an overall minimum FEC code rate is imposed. With the purpose of fulfilling the quality requirements of the users that receive just the base layer, it is assumed that the required code rate of the LA-FEC code used for the protection of this layer meets the overall one. Consequently, the minimum FEC code rate that can be used for the protection of the enhancement layer also equals the imposed overall value.

Moreover, unlike for the protection of non-layered multimedia, where the number of data packets within a FEC block is not important as long as it fulfills the imposed constraints, an extra restriction appears in the layered video case: the need that the number of data packets in a base layer protection block and the number of data packets

in an enhancement layer protection block need to meet the same relation as their bitrates do, in order not to lose synchronization. To that end, padding packets shall be added if needed to complete the matrices.

3.1.6.5 Evaluation of the adaptation

Figure 3.8 presents the results derived from applying the proposed LA-FEC scheme to the protection of a set of two-layered videos. The difference between the videos employed in the simulations relies on the bitrate of the enhancement layer, which is respectively 0.45, 0.35, 0.25 and 0.15 times the bitrate of the base layer, which is of 4 Mbps. Results are shown in terms of the percentage of blocks that contain at least one uncorrectable error. What is depicted is the value obtained when applying the best code that the following imposed limitations allow: a maximum FEC latency of 400 ms, the bitrate of the respective layer, and the minimum FEC code rate presented in the x-axis in the graphs.

First, we can see in the graphs that, as expected, the results of both strategies always improve as the conditions imposed become less restrictive, that is, if the minimum FEC code rate decreases and the bitrate of the enhancement layer gets greater (assumed a constant maximum FEC latency). Nevertheless, the results obtained after applying the LA-FEC scheme are significantly better compared to the ones obtained when the two layers are protected independently, especially for the base layer. Furthermore, the difference of performance between the two schemes also grows with the available protection resources. This proves that the LA-FEC scheme makes better use of the available protection resources. The reason behind both tendencies is that in addition to devoting more resources to the protection of data packets, it also becomes more likely to find more and more suitable protection configurations, a characteristic that intensifies in the case of the LA-FEC proposal.

3.2 EEP enhancement of Pro-MPEG COP3 codes

With the purpose of increasing the efficiency of standard Pro-MPEG COP3 codes for high PLRs, an enhancement consisting in introducing an extra dimension of protection is presented. Thanks to this extension, not only one or two dimensions (row-wise protection or column-wise protection or both, as in the standard), but up to three can be used per protection block upon convenience. The insertion of a third dimension implies increasing the possible combinations of data packets within a FEC block to create repair packets, that is, of protection configurations, which allows a better adaptation to the channel conditions.

3.2.1 Procedure

In essence, as indicated in the standard document, the set of data packets that are used to generate a certain parity packet belonging to dimension d in the Pro-MPEG COP3

Figure 3.9: EEP enhanced Pro-MPEG COP3 codes parity packet generation. Example with $D = 4$, $L = 5$, $N_3 = 5$ and $G_3 = 4$. Repair packets in the third dimension are generated through XORing the data packets equally colored.

codes can be identified by just detecting: (i) the minimum sequence number among all of them; (ii) the number of packets in this set, N_d ; and (iii) the sequence number gap between any two consecutive data packets, G_d , as it remains constant. For the row-wise-generated parity packets, the two latter variables respectively are $N_1 = L$ and $G_1 = 1$, and, for the column-wise-created ones, respectively $N_2 = D$ and $G_2 = L$. Based on this straightforward observation, we propose allowing the addition of an extra dimension, that is, generating a third set of parity packets with parameters N_3 and G_3 . An example is shown in Fig. 3.9.

For a certain k , the new variables N_3 and G_3 need to fulfill the following constraint to fit the matrix:

$$N_3 \cdot G_3 = k = D \cdot L$$

That means that, once D and L (or k) are specified, just a number of combinations of N_3 and G_3 are allowed. The introduction of a third dimension does not necessarily imply the addition of extra redundancy, but it increments the number of possible protection configurations to better utilize the available FEC bandwidth. That is, it simply enables new possible ways of distributing the protection through new combinations within the block of k data packets. Or, looking at it from a different perspective, to be able to reach the same quality level as the standard ones, employing less FEC overhead than the most suitable configuration that can be selected when working with only one or two dimensions.

Provided the number of data packets per protection block and the minimum FEC code rate, the extra parameters raise the number of possible combinations of data packets within a FEC block that can be employed to compute the repair packets. This action increases the capability of Pro-MPEG COP3 codes to adapt to a specific situation, whereas respecting the principles of the redundancy generation algorithm of the standard codes.

3.2.2 Complexity

In regard to the encoding computational complexity of the proposed enhanced codes, $N_d - 1$ XOR operations are carried out per repair packet in the d^{th} dimension. The decoding process is a bit more complicated to evaluate than in the standard case, as many possible realizations can take place. However, the number of XOR operations is always lower than $N_1 \cdot (N_2 - 1) + N_2 \cdot (N_1 - 1) + G_3 \cdot (N_3 - 1)$. Therefore, the computation complexity of both stages is linear with k in the worst case. In the same way as for the standard codes, if we consider the number of bit XOR operations, the total amount results from multiplying the referred number by the length of the longest data packet involved.

3.2.3 Evaluation of the scheme

Figure 3.10: Evaluation of the performance of standard and enhanced Pro-MPEG COP3 codes

For the evaluation of the proposal, we follow the criteria specified in reference stan-

dards [60, 57] to compare the performance of the standard Pro-MPEG COP3 codes with that of the enhanced ones. By means of a series of simulations, in which the conditions and limitations of the scenario are provided, the resulting points indicate the FEC overhead corresponding to the code configuration (dimensions to use, and values of N_1 , N_2 and N_3) that requires the lowest FEC bandwidth among those that fulfill the target quality requirement (mean time between blocks with errors equal to 4 hours). The actual parameter values of the most suitable configuration at each point are not shown to avoid confusion. The features and constraints of the scenarios are specified next:

- Channel error distributions: independent losses and burst losses (REIN model, 8 ms) with PLRs from 10^{-3} to 10^{-6}
- Additional latency due to FEC: 100 ms and 400 ms
- Video bitrates: 2 Mbps and 6 Mbps
- Packet sending arrangement (i.e. how data and repair packets are organized to be sent -see standards [60, 57]-): constant

The results obtained for the different configurations are depicted in Fig. 3.10. First, it can be observed that at low PLRs the proposed scheme does not present a better performance than the standard codes, as the latter are capable of satisfying the target criterion with very low extra bandwidth for any scenario (they perform very close to or meet the ideal code). However, as the PLR increases, our proposal gets more efficient than the standard Pro-MPEG codes for most of the configurations, specially, as expected, when the REIN error distribution applies, as new configurations can help boost error decorrelation. In the scenarios with the independent random error model, the gap between the two schemes is smaller. On the other hand, the performance of the enhanced Pro-MPEG COP3 codes improves, compared to the standard ones, as the number of data packets per FEC block grows. The reason is that the number and the quality of possible configurations also rises.

3.2.4 Adaptation to the protection of two-layered video streams

In this section, we use LA-FEC based on the proposed EEP enhancement to protect two-layered multimedia streams. The considered scenario is the same one as for the standard codes: the bitrate of the enhancement layer represents 0.45, 0.35, 0.25 and 0.15 times the bitrate of the base layer, which is of 4 Mbps, a maximum FEC latency of 400 ms is used. The only change is the inclusion of the possibility to generate repair packets of a third dimension. Figure 3.11 shows the results obtained under the described conditions. Again, the minimum FEC code rate is indicated in the x-axis of the graphs.

In the same way as with the standard codes, the proposed LA-FEC arrangement performs better than the traditional UEP scheme for almost all code rates (only for very high rates both perform alike). Moreover, although the performance of both of them improves as the imposed code rate decreases, the difference between the two strategies becomes greater.

 (a) (Bitrate enh. layer / bitrate base layer)
= 0.45

 (b) (Bitrate enh. layer / bitrate base layer)
= 0.35

 (c) (Bitrate enh. layer / bitrate base layer)
= 0.25

 (d) (Bitrate enh. layer / bitrate base layer)
= 0.15

Figure 3.11: Evaluation of the performance of the proposed LA-FEC scheme with standard and enhanced Pro-MPEG COP3 codes. Parameters: FEC latency = 400 ms; Channel PLR = 10^{-2} ; Error distribution = REIN model (8 ms); Bitrate base layer = 4 Mbps.

Finally, the number of impaired blocks as a result of applying the LA-FEC using the EEP enhanced codes are lower than that obtained employing the standard ones (Fig 3.8 and 3.11). This results further confirm the conclusions drawn in the previous section, which stated that EEP enhanced Pro-MPEG COP3 codes are significantly more efficient than the standard codes for high PLRs.

3.3 UEP enhancement of Pro-MPEG COP3 codes

The layer-aware adaptation presented before has two important limitations. On the one hand, all data packets in the same layer are assumed equally important (as mentioned in Chapter 5, this is a typical assumption made by LA-FEC and, in general, layer-level scheme designers). On the other hand, it is too specific, as it can exclusively be applied to

layered-video and does not cover other compress formats, including traditionally encoded one-layered video (e.g. MPEG-2, AVC, HEVC...). With the purpose of overcoming these limitations, we present in this section a UEP enhancement to the standard Pro-MPEG COP3 codes. This version constitutes a more general framework that is able to cover a much wider set of formats, virtually anyone that leads to any kind of prioritization of packets. The objective of the new extension is to protect the data in the encoded video stream at packet level, in accordance to their uneven relevance (this can be the result of any given data prioritization scheme) and without considering the specific encoding process. In this way, we will be able to increase the efficiency of the FEC codes, measured in terms of quality of the resulting video sequence, and not of packet recovery rate. The reason is that minimizing the degradation caused by packet losses on the decoded video, and not reducing the number of packets that are finally lost, is actually the ultimate goal of channel coding in the context of multimedia streaming. This quality improvement can be achieved through a smart distribution of redundancy among data, even at the expense of a lower overall packet recovery rate.

3.3.1 Procedure

Figure 3.12: Example of the distribution of data packets among several matrices for unequal protection. In the standard case (subfigure (a)), $k = 36$, $r_{\text{FEC}} = .25$, and repair packets are only generated column-wise, and so: $D = 4$ and $L = 9$. The presented UEP configuration (subfigure (b)) is made up of three matrices ($N_M = 3$), with matrix dimensions: $D_1 = 2$, $L_1 = 4$; $D_2 = 4$, $L_2 = 3$; $D_3 = 8$, $L_3 = 2$.

Our proposal takes as starting point the k video packets within the protection block, jointly with a set of k values that express the unequal importance of those packets, in terms of the potential distortion that their loss might introduce in the decoded sequence. These values are, as already mentioned, the result of applying a given data packet prior-

itization strategy to the packet stream. The new extension considers the arrangement of the k packets in sets of N_M matrices of dimensions $\{D_m, L_m\}, 1 \leq m \leq N_M$, in which all packets are protected and each one is included in just one matrix. The number of data packets that can be arranged in each matrix is $k_m = D_m \cdot L_m, 1 \leq m \leq N_M$. An example of a valid UEP configuration is shown in Fig. 3.12. The mapping between packets and matrices is done in the following fashion: the k_1 most relevant packets, according to the data prioritization scheme, are arranged in the first matrix, the following k_2 are arranged in the second one, and so on. Additionally, the parameter r_{FEC} is considered for limiting the overall number of repair packets that can be generated per block, and so:

$$\sum_{m=1}^{N_M} (D_m + L_m) \leq r_{\text{FEC}} \cdot k$$

Therefore, each possible combination of values of the variables $N_M, D_1, L_1, \dots, D_{N_M}, L_{N_M}$ constitutes a feasible configuration of the proposed protection mechanism. Each configuration is expected to introduce an overall distortion value that depends on the actual distortion values associated with the packets and on the channel's behavior. The most suitable configuration is the one that minimizes the overall expected distortion.

The number of feasible configurations relies on the value of variables k and r_{FEC} , and on the maximum number of matrices that a configuration can have, $N_{M_{\max}}$. This number can be potentially huge. Thus, in scenarios where strict time restrictions apply, suboptimal search techniques might be employed to find sufficiently good configurations whereas fulfilling latency requirements. This is further considered in Chapter 4.

Furthermore, since the most convenient packets to be protected together may not lay sequentially in the bitstream, we allow the layout of non-consecutive packets in the matrices. For that reason, the signaling of the data packets XORed to create the repair packets varies with respect to the standard case. In the latter, only the inclusion of the matrix dimensions is needed, as expressed in RFC 6015 [58]. However, if our proposal is used, data packets need to be explicitly signaled, either employing a mask or a similar mechanism. That might involve an increase in bandwidth occupation, as header extensions may be needed to identify all the data packets in the FEC block. Section 3.3.3 presents different approaches to deal with data-to-repair-packet signaling.

3.3.2 Proof of concept

In this section, we present the experiments and the subsequent analysis through which we discuss the benefits of our proposal. The evaluation consists in a direct comparison between the performance of the proposed extension when employing different number of matrices, and that of the standard codes, under a wide range of conditions.

The evaluation has been carried out considering that the faced communication channel is essentially bursty (e.g. DSL and wireless networks). Under this assumption, following the justification in Section 3.1.3, we only consider parity packets generated column-wise.

Function	f
Uniform	0
Linear	1
Quadratic	2
Cubic	3

 Table 3.1: Value of parameter f .

 Figure 3.13: Emulated packet distortion values. Example with $k = 20$.

Parameter	Values
N_M	1 to 4
R_{video}	2.5 Mbps, 5 Mbps
r_{FEC}	10 %, 20 %
k	20 to 400
d_p	Uniform, linear, quadratic, cubic
C	REIN, simplified G-E
p_e	10^{-2} , 10^{-1}

Table 3.2: Test parameters.

To make a convincing proof of concept, we have considered a great number of contexts, regarding the variety of parameters involved and their values: the number of matrices used, N_M , the video bitrate, R_{video} , the protection bandwidth overhead, r_{FEC} , the number of data packets per FEC block, k , the sets of packet distortion values with which to simulate different data prioritization scheme outcomes, d_p , $1 \leq p \leq k$, the model reflecting the behavior of the channel, C , and the PLR, p_e . The values used in the tests are detailed in Table 3.2.

The data prioritization strategy outcomes, that is, the distortion values associated to the different packets, d_p , $1 \leq p \leq k$, have been emulated employing the following normalized functions: uniform, linear, quadratic, and cubic. For all of them, data packets take values in the next fashion:

$$d_p = \left(\frac{k - p + 1}{k} \right)^f, 1 \leq p \leq k$$

where parameter f changes with the function, as shown in Table 3.1, thus resulting in the value relationships depicted in Fig. 3.13. When using the uniform function, all packets are assumed equally important. With the linear function, packets are given distortion values that relate in a linear fashion. This relationship is of course quadratic and cubic in the two remaining cases, respectively. The greater the value of parameter f is, the

(a) $R_{\text{video}} = 2.5$ Mbps, $r_{\text{FEC}} = 10\%$, $p_e = 10^{-1}$, d_p uniform.

(b) $R_{\text{video}} = 2.5$ Mbps, $r_{\text{FEC}} = 10\%$, $p_e = 10^{-1}$, d_p linear.

(c) $R_{\text{video}} = 2.5$ Mbps, $r_{\text{FEC}} = 10\%$, $p_e = 10^{-1}$, d_p quadratic.

(d) $R_{\text{video}} = 2.5$ Mbps, $r_{\text{FEC}} = 10\%$, $p_e = 10^{-1}$, d_p cubic.

Figure 3.14: Resulting distortion derived from using 1 to 4 matrices to protect the number of data packets indicated in the horizontal axis, subject to the same scenario conditions ($R_{\text{video}} = 2.5$ Mbps, $r_{\text{FEC}} = 10\%$, $p_e = 10^{-1}$), but assuming the different distortion function indicated for each graph. The use of dissimilar number of matrices is plotted in different colors and markers (see legend).

less uniform the importance of the video packets becomes.

Results are given, for each configuration and context, in terms of the overall expected distortion, which equals the summation of the expected distortion of all the packets in the protection block. This value is computed as the distortion value given by the distortion model, multiplied by the likelihood of losing that packet during transmission and not being able to recover it. Moreover, for a more suitable analysis of results, overall distortion values have been normalized with respect to a reference value. This value equals the best result obtained using the standard mechanism under the same conditions, allowing the bigger protection block considered in the simulations, 400 video packets. So, it becomes much easier to discern, based on the range between the different configurations, the benefit or damage derived from their use.

Figure 3.14 shows the results of the experiments for a specific context ($R_{\text{video}} = 2.5$ Mbps, $r_{\text{FEC}} = 10\%$, and $p_e = 10^{-1}$). Each graph has been generated using different packet distortion values. The horizontal axis expresses the number of packets available in the FEC block. The depicted relative distortion value is, for each value of k , the one obtained with the most convenient combination of matrix sizes, given the maximum allowed number of repair packets (computed as $k \cdot r_{\text{FEC}}$). As a reference, black vertical lines have been added to mark different FEC latency periods, concerning the time devoted to wait for data packets at the receiver before applying the decoding algorithm: 200, 400, 600, 800 and 1000 ms. So, one can check how large protection blocks can be when each of these latency restrictions apply, and how beneficial the proposed UEP extension is, compared to the standard one, under each time limitation.

In light of these data, we can claim that the results obtained when using the proposed extension outperform the ones with the standard case, provided that the protection scheme has enough capability to simultaneously (i) cope with burst errors, and (ii) provide unequal protection to video packets, regarding their importance. Having enough capability ultimately means having enough number of repair packets to be able to generate suitable configurations to those ends. This power completely depends on the specific context. This fact is represented in each graph, where two opposite behaviors can be readily spotted. If the number of data packets per protection block is too low, the results obtained when employing the UEP extension are worse than those obtained when applying the standard case. Moreover, the number of data packets that are required to overcome the standard codes increases with the number of matrices that are used, m . However, once the minimum value of k is reached (and so, of $k \cdot r_{\text{FEC}}$), the protection strategy is capable of introducing enough interleaving depth, and therefore, properly provide unequal protection.

So, if the context allows the generation of enough number of repair packets, we observe that the overall distortion decreases with the number of matrices used, as long as the data packets are not uniformly relevant. In fact, the more distinct the video packets are, considering the distortion value associated, the greater the gain is when increasing the number of matrices. Additionally, as can be seen in all graphs, the gain resulting from increasing the number of matrices per FEC block, i.e., using $m + 1$ matrices instead of m , gets smaller with m . That is, the more matrices in the configuration it is used to

Figure 3.15: Resulting distortion derived from using 1 to 4 matrices to protect the number of data packets indicated in the horizontal axis, subject to different scenario conditions (R_{video} , r_{FEC} , and p_e), but assuming the same distortion function. The use of dissimilar number of matrices is plotted in different colors and markers (see legend).

protect a group of packets, the less beneficial it is to switch to a configuration with one more matrix. Although, if the number of data and the number of repair packets are great enough, there will always be a distortion reduction.

Regarding the type of communication channel, we can see that, in the case of encountering a channel in which bursts are of a fixed length, as the one modeled with the REIN model, rather small FEC blocks suffice to be able to generate enough parity packets so as to be able to cope with bursts and provide adaptive protection. The reason is that a smaller number of columns is needed to ensure error decorrelation (equal or greater than the number of packets affected by a single burst). In channels where losses take place in bursts of variable length, as the one modeled with the simplified G-E model, a greater k is necessary to outperform the standard mechanism, as more columns per matrix are needed to reach error decorrelation. Those results can be spotted in the presented graphs, where systems using the proposed extension and facing channels with fixed-length bursts (non-red solid lines) outperform the standard technique and converge to a steady performance quicker than those utilizing the proposed extension but facing channels with variable-length bursts (non-red dashed lines).

The above mentioned graphs show a general tendency for the whole set of simulations. Indeed, altering the scenario conditions basically affects (i) the gain gap between the different configurations, and (ii) the influence of the size of the protection block.

A second figure, Fig. 3.15, has then been included with the aim of illustrating the specific effect of the rest of the parameters analyzed: R_{video} , r_{FEC} , and p_e . It contains four graphs where packet distortion values always present a quadratic relationship. The first one, Fig. 3.15(a), is just a reposition of the corresponding graph in Fig. 3.14. The rest of them present the results obtained after modifying the value of the indicated parameter with respect to the first one. In Fig. 3.15(b), we can observe that, if the video bitrate is higher, systems implementing the proposed extension require a greater FEC block to outperform the standard codes and converge to a steady performance. The reason is the following: more data packets are lost within a single burst, and thus, more parity packets are needed to effectively decorrelate channel error. This is not a problem in terms of time, as the number of packets received in the same period rises in the same proportion of the bitrate. However, it increases the complexity of the process to obtain the optimum configuration, as this increases the number of feasible configurations. As mentioned before, fast search techniques might be required to fulfill latency restrictions when the number of configurations is high.

On the contrary, Fig. 3.15(c) illustrates that greater protection bandwidth overheads allow the system employing the extension to use smaller FEC blocks to be able to outperform the standard case. In addition, Fig. 3.15(d) shows that, as p_e increases, the system requires fewer repair packets to outperform the standard technique. This conclusion is of special interest to us, as the main motivation of our work is to improve the performance of the Pro-MPEG COP3 codes for high PLRs, where these codes are weaker.

3.3.3 Signaling

Figure 3.16: Example of data-to-repair-packet setup.

When applying systematic Application-Layer Forward Error Correction (AL-FEC) codes to deal with channel impairments, one of the main design concerns is the amount of extra bandwidth devoted to the repair flow [61]. One key goal is then to avoid expending too much bandwidth on signaling which data packets are protected by which repair packets. To that end, there exist RFCs and other standardization documents that specify, both within and outside the FEC framework defined in the RFC 6363 [62], compact, efficient methods to better accommodate this information, regarding the particularities of each code. For instance, RFC 6681 [63] and RFC 6682 [64] incorporate all the required parameters for the receiver to replicate the Raptor/Raptor Q encoding scheme used to create the redundant flow, so that the packets involved are properly arranged and the recovery process can be carried out. With the same purpose and in a similar way, RFC 5510 [65] and RFC 6865 [66] describe a means to include Reed-Solomon codes necessary parameters.

In the case of the standard Pro-MPEG COP3 codes, the dimensions of the matrix are used to signalize the links between the two flows, as described, although only for column-wise created parity packets, in RFC 6015 [58]. Every repair packet allocates all the required information to unequivocally identify all the data packets that it protects: (i) the sequence number of the first data packet in the stream that it protects, called sequence number base (SN_{base}); (ii) the number of packets that it protects, which for row-wise generated parity packets equals the number of columns (L), and for column-wise generated FEC packets equals the number of rows (D); and (iii) the sequence number difference between any two consecutive protected data packets. For row-wise created repair packets, this difference equals one, as they protect consecutive packets in the stream. However, for column-wise generated parity packets, this difference equals L . Furthermore, if the proposed EEP extension is implemented, the signaling associated with the third dimension can be carried out through the same procedure: repair packets will include the following information: SN_{base} , number of packets protected (N_3), and sequence number difference between consecutive protected packets (G_3).

Nevertheless, these compact signaling mechanisms are only possible if the connections between repair and data packets respond to nonflexible, replicable connection setups. This is not the case, for instance, when the links are set randomly, or when they

correspond with exterior criteria, e.g., they are the result of packet-level distortion minimization problems within unequal error protection strategies, which is the case of the proposed UEP enhanced Pro-MPEG COP3 codes.

So, with the aim of using a signaling method that can fit the characteristics of the new scheme and of the transmission scenario, three different signaling approaches are evaluated. These approaches can actually be applied to any data-to-repair-packet connection setup, although in this section we provide an analysis focused on the specific characteristics of the UEP enhanced Pro-MPEG COP3 codes. The presented schemes are subsequently evaluated in terms of average signaling overhead per packet and overall bandwidth devoted to this task per protection block.

Figure 3.16 depicts an example of setup. This setup is used to illustrate how the presented signaling methods work.

3.3.3.1 Mask signaling method

Figure 3.17: Mask signaling method provided the setup in Fig. 3.16.

This is the only standardized procedure to signalize flexible connection setups, however only for protection blocks of up to 48 data packets [67]. What is described here represents an extension to this method. Each FEC packet carries the following signaling information to identify the data packets that it protects: (i) the minimum sequence number among the k data packets in the protection block ($SN_{base}^{overall}$) (2 octets); and (ii) a mask made up of k bits. Setting the p^{th} bit means that the repair packet was generated using the $(SN_{base}^{overall} + p - 1)$ -th data packet in the protection block ($\lceil k/8 \rceil$ octets, as the number of octets has to be integral). Data packets are left unaltered. Figure 3.17 shows this method for the setup in Fig. 3.16.

3.3.3.2 Explicit signaling method

Figure 3.18: Explicit signaling method provided the setup in Fig. 3.16.

Each repair packet transports: (i) the sequence number of all the data packets that it protects (2 octets per each of them). Figure 3.18 shows this method for the example in Fig. 3.16.

3.3.3.3 Shared-load signaling method

Figure 3.19: Alternative sequence numbering provided the setup in Fig. 3.16.

Firstly, this method generates an alternative, independent sequence numbering of the data packets protected by each FEC packet. To that end, the packet with the minimum real sequence number among the ones protected by a given repair packet receives an alternative base sequence number, which is incremented by one for the data packet in this same set with the second minimum real sequence number, and so on. This is depicted in Fig. 3.19. This procedure allows the consideration of the data packets in the same set as if they were located consecutively in the stream, enabling a compact signaling in the sense of the standard Pro-MPEG COP3 codes.

Figure 3.20: Shared-load signaling method provided the setup in Fig. 3.16.

Contrary to other methods, the signaling load needs to be shared between the FEC packets and the data packets. This can be easily carried out through a header extension that in no way compromises the systematic nature of the code, as the original content is not modified.

Figure 3.20 shows this method for the example in Fig. 3.16. Repair packets carry the following signaling information: (i) the minimum sequence number among the packets that it protects (2 octets); and (ii) the number of data packets protected (2 octets). Data packets include the following signaling information: (i) the sequence number of the repair packets that were created for its protection (2 octets per FEC packet); and (ii) the alternative sequence number derived from the new numbering (2 octets per FEC packet).

3.3.3.4 Comparison of the signaling methods

The presented schemes are now compared in terms of the overhead required to put them into practice. To this end, the space that is required in an RTP-FEC header to accommodate the necessary information is measured. In this comparison, three different measurements are considered: the average signaling overhead per packet, the average signaling overhead per packet carrying signaling information, and the overall bandwidth devoted to this task per protection block. It is assumed that all data packets are connected to one repair packets, like in the proposed UEP Pro-MPEG COP3 codes. An overall extra overhead for protection purposes of 15 % of the bitrate of the main flow is imposed.

Figure 3.21: Overhead devoted to repair-to-data-packets signaling per packet.

Figure 3.22: Overhead devoted to repair-to-data-packets signaling per packet carrying signaling information.

Figure 3.21 shows the evolution of the average bandwidth required per packet (data and repair) for packet signaling with the number of data packets in the protection block. Figure 3.22 presents this evolution but only considering those packets indeed carrying

Figure 3.23: Overall overhead devoted to repair-to-data-packets signaling per protection block.

signaling information. Finally, Fig. 3.23 depicts the evolution of the overall bandwidth with the number of data packets in the protection block.

As can be seen, the signaling load per packet required with the mask signaling method (Fig. 3.21 and Fig. 3.22) grows quasi-linearly (in a staircase manner) with the size of the protection block, which makes the overall signaling overhead grow exponentially with this number, as can be seen in Fig. 3.23. If the number of data packets in the block is kept low, this method is very well suited. However, it becomes less efficient as the number of data packets in the protection block increases.

On the contrary, the performance of the other two methods remains rather constant with the size of the protection block. Particularly, as can be seen, for a sufficiently high number of data packets, the explicit method requires the lowest signaling bandwidth per packet and overall (Fig. 3.21 and Fig. 3.23). However, this method might not handle it conveniently. As shown in Fig. 3.22, the load of a single FEC packet directly depends on the number of data packets protected by it, which might differ enormously from repair packet to repair packet. Thus, although it is averagely efficient, this behavior may lead to undesirable events, like exceeding the network's MTU.

Regarding the shared-load signaling method, we observe that it demands more resources than the latter, as depicted in Fig. 3.21 and Fig. 3.23. Nevertheless, it distributes the associated load more efficiently among packets (Fig. 3.22). Thus, the amount of extra information added to each packet is very low and fairly constant.

Finally, it can be concluded that, as already said, the method based on the mask is perfectly appropriate for small protection blocks. As this block size increases, the explicit signaling method becomes the most suitable in terms of the extra bandwidth required. However, particular scenario features might require a more smartly handled packet signaling, making the shared-load signaling method the best choice.

3.4 Conclusions

In this chapter, we have first introduced a Layer-Aware FEC scheme in which the Pro-MPEG COP3 codes are used to increase the QoS of two-layered video distribution. To that end, base layer data packets are protected both alone and jointly with those of the enhancement layer. The latter is carried out by means of distributing the base layer packets among the repair packets of the enhancement layer. With this straightforward step, the recovery capability of the protection scheme is significantly raised. To test the proposal, we have compared it to a standard scheme in which no layer dependency is considered. Simulation results show a relevant benefit in the introduction of the proposed protection mechanism.

Secondly, in light of the significantly low efficiency of the AL-FEC Pro-MPEG COP3 codes at relatively high channel's PLR, we have first presented an enhancement to these codes that consists in allowing the addition of a third dimension of parity packets. The introduction of extra parameters raise the number of possible protection configurations within a FEC block. This action increases the capability of the codes to adapt to the specific conditions of the communication channel. This is done through boosting error decorrelation, which leads to increasing recovery capacity. The good performance of the proposed enhanced Pro-MPEG COP3 codes have been proved through a set of tests, in which they have been compared to the performance of the standard ones under standard network conditions. Additionally, we have adapted the new extension to the protection of two-layered video, in the same fashion as for the standard case, obtaining a noticeable gain when compared to the traditional approach and to when the standard Pro-MPEG COP3 codes are employed.

Next, we have described a framework based on the Pro-MPEG COP3 codes to unequally protect any given sequence of unequally-important data packets, regardless of the employed encoding format. Our approach enables the use of a number of matrices of unequal dimensions to protect the data packets in the FEC block, which allows a better adaptation to the video stream, as different FEC code rates can be applied to different sets of data packets, depending on their importance. We have made a proof of concept of the new UEP Pro-MPEG COP3 codes through comparing its performance with that of the standard mechanism for a wide range of possible scenarios, depending on the parameters analyzed and their values. Evaluation results show that the proposed mechanism outperforms the standard one, as long as the protection scheme has enough capacity to simultaneously deal with burst errors and provide unequal protection to data packets. In addition, given the impossibility of using a compact means to signalize which data packets are protected by which repair packets in this kind of protection strategies, we have presented several procedures to enable arbitrary data-to-repair-packet connection setup signaling. These methods highly differ in the amount of bandwidth employed in signaling and how this overhead is distributed among data and repair packets. The carried-out simulations show that the best choice depends on the considered scenario and the preferences of the user.

The work presented in this chapter has resulted in several publications. Both the

EEP extension to the Pro-MPEG COP3 codes and the adaptation of the standard and the enhanced EEP cases to the protection of two-layered video streams were presented in [6]. The aforementioned proof of concept for validating the proposed unequal error protection (UEP) extension to the Pro-MPEG COP3 codes and exploring its potential benefits was published in [7]. Finally, the procedures to enable arbitrary data-to-repair-packet connection setup signaling was published in [8].

Chapter 4

Unequal error protection scheme optimization

4.1 Introduction

In the considered scenario, where video-related content is streamed through IP networks (e.g. television broadcasting, video on demand, voice over IP or videoconferencing), many of the provided services, particularly those that involve live event broadcasting or some kind of interactivity, are not only demanding in terms of quality, but also of latency. In these situations, the protection mechanisms that are selected to increase the reliability of the communication channel must perform in real time. Particularly, in the case of the UEP schemes, like the one proposed in Section 3.3, this restriction includes the time to allocate available resources among source data in regard to their importance. This procedure is carried out in a variety of ways, depending on the specifics of the transmission.

Indeed, in some UEP strategies, the distribution of resources responds to an a priori arrangement. In these cases, a pre-established fixed categorization of data and a rather constant distribution of data among these categories are habitually assumed [48, 12]. However, the distribution of resources is usually the result of an optimization problem, in which the optimal distribution is one in a set of feasible solutions. This problem is tackled differently regarding the characteristics of the protection mechanism (e.g. whether the distribution of resources is set before the transmission begins -a single instance that is solved offline-, or periodically along the course of the transmission -a series of instances that are solved on the fly), the level at which the stream is analyzed and at which resources are distributed (e.g., macroblock, slice, frame, layer, etc.), and the imposed restrictions (in particular regarding latency and computation complexity).

Regarding combinatorial optimization, strategies that work with rather small solution spaces or are not conditioned by strict restrictions, usually perform an exhaustive search to find the optimal solution [68, 69]. On the other hand, if the optimization problem fulfills the necessary requirements (e.g. continuous first partial derivatives, convexity, etc.), or a relaxation method can be applied, strategies commonly turn to exact optimization

algorithms to find the optimal solution (e.g. integer or mixed-integer programming). In particular, there exist numerous proposals posing optimization problems that can be formulated in terms of rate-distortion. Usually, in these strategies, channel coding is seen from a high level of abstraction and they do not aim at a further optimization of the code. Specifically, designers frequently chose maximum-distance-separable (MDS) channel codes (e.g. Reed-Solomon codes), that is, ideal codes, to that end. In these cases, the optimization problem is usually solved by the method of Lagrange multipliers [70, 71].

However, in many situations, protection schemes cannot include exact optimization algorithms to solve the optimization problem. This is typically the case when a white-box perspective is assumed, in which the designer aims at obtaining the best internal parameters values of the protection mechanism to optimally distribute the available resources. In addition, regarding combinatorial optimization, the characteristics of the scenario may not allow the use of brute-force search. In these circumstances, many proposals use iterative methods to find the optimal solution. The iterative methods can be either standard (e.g. iterative linear programming [72], branch and bound method [73]) or created ad hoc [74, 75].

Finally, in scenarios limited by hard restrictions, either regarding latency or computation complexity, with a large solution space, where neither exact optimization algorithms nor iterative methods can be used, metaheuristics are commonly employed to guide the search process [76]. These approaches obtain sufficiently good solutions, that is, solutions that may not be optimal but are considered good enough for the purposes of the problem, given the imposed restrictions. In this regard, designers mainly opt for search methods based on genetic algorithms (GA) [77, 78] or guided local search (GLS) [79, 80]. In principle, a broad set of metaheuristics are available to designers [81], from population-based nature-inspired approaches like evolutionary algorithms (e.g. the above mentioned GA [82]) or swarm algorithms (e.g. ant colony optimization –ACO– [83] or particle swarm optimization –PSO– [84]), to single-solution options like simulated annealing (SA) [85], tabu search (TS) [86], or the also already mentioned GLS [87]. Nevertheless, the selection of a metaheuristic highly depends on the actual context at hand, as its performance relies on how its characteristics fit in with those of the scenario and the imposed limitations. In the resource allocation problem that we consider, the time that takes the strategy to reach acceptable solutions and the simplicity of the scheme are of particular concern. The former is important so as to comply with real-time application latency requirements, whereas the latter responds to practical reasons, as the employed algorithm is intended to be incorporated in transmission modules to be deployed in a potentially high range of devices, from high-level computational capacity servers to computationally limited terminals. Of the available approaches, SA is one of the options that is better suited for the scenario: it can meet such rigid conditions regarding time and complexity, whereas still providing appropriate solutions. Although, depending on the context, it may be outperformed by other algorithms in the long run regarding how close final solutions are to the overall optimal one, contrary to other metaheuristics, SA is well known for being a "quick starter", that is, a method that is able to obtain good enough solutions in short periods of time [88, 89, 90]. Moreover, it is highly prone to problem-specific adaptations.

These are the reasons why this rather simple metaheuristic has been successfully widely applied to solve many combinatorial problems within and outside the area of communications [91, 92, 93]. In addition, with the aim of making this memoryless technique more effective, it can be hybridized with TS, which provides memory structures to enhance the search of solutions [94, 95]. TS encourages intensification or diversification upon convenience, and so helps guide the search to reach better solutions in very short time periods.

Consequently, we have designed a fast and robust optimization technique based on SA and guided by TS. This technique aims at finding near-optimal configurations of the UEP framework presented in Chapter 3 when strict time restrictions apply. In the considered protection strategy, the optimal configuration (i.e. the number of matrices and their size that minimize the overall expected distortion) is computed periodically, so that it adapts to the varying characteristics of both the packet stream and the communication channel. Moreover, the number of feasible configurations can be very high, as it grows along with the number of data and repair packets in the block. So, the described combinatorial problem, with a potentially large solution space, and very restricted processing time, is solved by the proposed hybrid procedure. The relative importance of the involved data packets is assumed to be known before starting up each new computation, and is the result of applying a given distortion model, like the one proposed in Chapter 5.

4.2 Problem description

4.2.1 Optimization problem formulation

As referred in Chapter 3, the problem is formalized in the following terms. Let N_M be the number of matrices in a configuration, and C_m and R_m respectively the number of columns and rows in matrix m , $1 \leq m \leq N_M$. It is assumed that matrix 1 protects the $C_1 \cdot R_1$ most relevant packets in the protection block, matrix 2 the following $C_2 \cdot R_2$, and so on.

The goal of the optimization problem is to find the most convenient protection configuration, s^{opt} , that is, the combination of values of the variables $N_M, C_1, R_1, \dots, C_{N_M}, R_{N_M}$ that minimizes the overall expected distortion introduced by the N_P data packets in the block, D_T . The general formulation of the problem is as follows:

$$\min_{N_M, C_1, R_1, \dots, C_{N_M}, R_{N_M}} D_T = \sum_{p=1}^{N_P} D_p \cdot P_p \quad (4.1)$$

where D_p is the distortion associated with losing packet p , which, as mentioned, results from applying the distortion model and is assumed to be known beforehand, and P_p is the likelihood of losing packet p . P_p not only depends on the behavior of the channel, but also on the selected protection configuration, and, more specifically, on the protection matrix where packet p is allocated.

The following conditions are imposed to the problem:

- With the purpose of preserving core characteristics and the extremely low level of complexity of the redundancy generation algorithm of the standard codes, we impose that all data packets are protected, and that every data packet is protected by one and only one matrix. Moreover, only the last matrix, matrix N_M , may need padding packets to complete its packet distribution, as the contrary would lead to unequal and incoherent levels of protection of different packets in the same matrix, considering their importance, and heavily increasing the number of possible solutions. These conditions are formalized as follows:

$$\sum_{m=1}^{N_M-1} C_m \cdot R_m + C_{N_M} \cdot (R_{N_M} - 1) < N_P \leq \sum_{m=1}^{N_M} C_m \cdot R_m \quad (4.2)$$

- Following the reasons in Subsection 3.1.3, it is assumed that parity packets are generated only column-wise:

$$\sum_{m=1}^{N_M} C_m = N_{\text{FEC}} \quad (4.3)$$

where $N_{\text{FEC}} = N_P \cdot ((1/r_{\text{FEC}}) - 1)$ is the number of repair packets, and r_{FEC} is the imposed minimum code rate.

Provided specific time restrictions, only solutions considering 1 to $N_{M_{\max}}$ matrices will be able to be tested, and so $N_M \in \{1, \dots, N_{M_{\max}}\}$. The overall optimization problem is then divided into $N_{M_{\max}}$ subproblems. Each subproblem works with configurations of different number of matrices and delivers the best solution that it has been able to find. So, at the end $N_{M_{\max}}$ candidate solutions to the overall optimization problem will be available, s_1^{opt} to $s_{N_{M_{\max}}}^{\text{opt}}$. The overall optimal solution, s^{opt} , will be the best one among them.

Furthermore, assuming the imposed conditions, it can be seen that each subproblem has a number of degrees of freedom that depends on the number of matrices of the configurations that it works with. In particular, if the configurations are made up of N_M matrices, the problem will have $2 \cdot (N_M - 1)$ degrees of freedom. This calculation considers the dimensions of all the matrices but one, whose size can be computed once the others have been set.

4.2.2 Solution space of the subproblem with configurations of matrices

Let Ω_{N_M} be the solution space of the subproblem, i.e., the number of possible combinations of values that fulfill the expressed restrictions, if N_M matrices are employed. Then, for given values of variables N_P and N_{FEC} , the cardinality of the solution space, $N_{\Omega_{N_M}}$, can be computed through the following iterative and recursive equation:

$$N_{\Omega_{N_M}}(N_P, N_{\text{FEC}}) = \sum_{i=1}^{N_{\text{FEC}}-N_M+1} \sum_{j=1}^{\left\lfloor \frac{N_P-N_{\text{FEC}}+i}{i} \right\rfloor} N_{\Omega_{N_M-1}}(N_P - i \cdot j, N_{\text{FEC}} - i) \quad (4.4)$$

where the base case of the recursive equation is:

$$N_{\Omega_1}(N_P, N_{\text{FEC}}) = 1 \quad (4.5)$$

Each iteration is used to set the number of columns and rows of the current matrix, and each recursion step considers all the feasible solutions once the dimensions of the current matrix are known. So, in the first recursion step, each iteration sets the values of i and j , which are the number of columns and rows of the first of the N_M matrices, respectively. In this way, two variables out of the $2 \cdot (N_M - 1)$ ones are set. The expression inside the summations then considers all the feasible combinations of sizes of the remaining $N_M - 1$ matrices, i.e., the feasible combinations of values of the remaining $2 \cdot (N_M - 2)$ variables. In the following recursion step, each iteration sets the values of i and j , that is, the size of the second matrix, thus setting two more of the original $2 \cdot (N_M - 1)$ variables. In the same way as in the previous step, the expression inside the summations considers all the feasible configurations, once the dimensions of the first two matrices are set. The recursion continues in the same way. The base case in (4.5) considers all the feasible combinations, once the size of all matrices but the last one are set. Since all the $2 \cdot (N_M - 1)$ variables are already set, the number of remaining feasible solutions is one.

A quick look at (4.4) allows us to observe that the number of feasible solutions, $N_{\Omega_{N_M}}$, can easily shoot up as the values of N_P , N_{FEC} and N_M increase. We include in Table 4.1 some numbers with typical but rather restrained values of the three parameters to illustrate this issue. The solution space quickly becomes far too large to allow the calculation of the distortion associated with each configuration and to compare the obtained values in real time.

4.3 Proposed optimization procedure

The pursued goal is, as mentioned before, to select the most suitable protection configuration to minimize the deterioration of the transmitted video caused by packet losses, given a limited extra bitrate budget for this purpose and a maximum time of accomplishment. To that end, an adapted metaheuristic based on the hybridization of two optimization techniques, simulated annealing (SA) and tabu search (TS), is proposed: hybrid simulated annealing (HSA).

The first step of the proposed procedure is to reduce the number of solutions to each of the subproblems. To that end, two extra restrictions are included to the optimization problem. Now, a configuration is only considered feasible if the new conditions are also met. These new conditions are consistent with the objective of protecting data packets considering their relevance, providing more important packets with more solid protection: lower code rates (more redundancy) and stronger error decorrelation (increasing the probability to recover lost packets).

The second step incorporates the optimization process strictly speaking. It is based on a modified SA, which is enhanced by the inclusion of memory structures via TS, and adapted to the specifics of the scenario.

N_P	N_{FEC}	N_M	$N_{\Omega_{N_M}}$
100	10	2	262
100	10	3	28,029
100	10	4	1,639,291
100	20	2	297
100	20	3	39,698
100	20	4	3,200,804
200	20	2	653
200	20	3	188,246
200	20	4	32,336,043
200	40	2	704
200	40	3	233,423
200	40	4	48,473,869
300	30	2	1,088
300	30	3	539,300
300	30	4	163,310,575
300	60	2	1,158
300	60	3	636,525
300	60	4	222,834,297

Table 4.1: Examples of cardinality of the solution space

In particular, SA has been tuned to make core elements and internal schedules dependent on the imposed application-dependent time constraints, and so force the algorithm to invariably comply with external temporal requirements. Moreover, the search method has been adjusted considering the specific scenario and the characteristics of the solutions, so as to enable performing different types of searches in regard to the needs of the different stages of the procedure. These adjustments result in a smart sampling of the solution space, and hence in a more efficient global search of near-optimal solutions.

For its part, TS contributes by storing and analyzing past moves of the system to more effectively guide the search of new solutions throughout the method.

In the next subsections, both steps are discussed in depth.

4.3.1 Solution space reduction

The solution space is delimited by adding the two following restrictions:

- More resources are devoted to more important packets. Using more columns boosts interleaving, and so, error decorrelation, in channels with memory, since data packets in the same column are more distant in the packet stream. Thus, for each feasible solution of N_M matrices, the number of columns of the matrices will be monotonically decreasing as the index of the matrix increases:

$$C_{m_1} \geq C_{m_2}, 1 \leq m_1 \leq m_2 \leq N_M \quad (4.6)$$

N_P	N_{FEC}	N_M	$\tilde{N}_{\Omega_{N_M}}$
100	10	2	50
100	10	3	999
100	10	4	11,593
100	20	2	50
100	20	3	1,100
100	20	4	14,557
200	20	2	100
200	20	3	4,598
200	20	4	123,026
200	40	2	100
200	40	3	4,719
200	40	4	138,281
300	30	2	150
300	30	3	10,890
300	30	4	467,534
300	60	2	150
300	60	3	10,901
300	60	4	501,975

Table 4.2: Examples of cardinality when applying the proposed conditions for solution space reduction

- Lower code rates are devoted to more important packets. Using fewer rows reinforces the capability of the repair packets to rebuild data packets, as the likelihood of losing more than one packet per column decreases. In this case, for each feasible solution of N_M matrices, the number of rows of the matrices will be monotonically increasing as the index of the matrix increases:

$$R_{m_1} \leq R_{m_2}, 1 \leq m_1 \leq m_2 \leq N_M \quad (4.7)$$

If these extra restrictions are considered, the new number of feasible solutions of a subproblem working with configurations of N_M matrices, $\tilde{N}_{\Omega_{N_M}}$, can be computed as follows:

$$\tilde{N}_{\Omega_{N_M}}(N_P, N_{\text{FEC}}, C_{\text{max}}, R_{\text{min}}) = \sum_{i=\left\lceil \frac{N_{\text{FEC}}}{N_M} \right\rceil}^{\min(C_{\text{max}}, N_{\text{FEC}} - N_M + 1)} \sum_{j=R_{\text{min}}}^{\left\lfloor \frac{N_P - N_{\text{FEC}} + i}{i} \right\rfloor} \tilde{N}_{\Omega_{N_M-1}}(N_P - i \cdot j, N_{\text{FEC}} - i, i, j) \quad (4.8)$$

where the base case of the recursive equation:

$$\tilde{N}_{\Omega_1}(N_P, N_{\text{FEC}}, C_{\text{max}}, R_{\text{min}}) = \begin{cases} 1 & \text{if } \left\lfloor \frac{N_P - N_{\text{FEC}}}{N_{\text{FEC}}} \right\rfloor \geq R_{\text{min}} \\ 0 & \text{otherwise} \end{cases} \quad (4.9)$$

This equation works in the same way as the one in the previous section, only incorporating the new restrictions. In this regard, C_{max} and R_{min} , which respectively are the maximum number of columns and the minimum number of rows that the current matrix can have, according to (4.6) and (4.7), restrict the range of feasible values of the iterators. So, if the maximum number of rows that the current matrix could have had, regarding the remaining number of data and repair packets, is not higher than the imposed minimum number of rows, that is, if $\lfloor (N_P - N_{\text{FEC}}) / N_{\text{FEC}} \rfloor \geq R_{\text{min}}$, then this summation becomes an empty sum, and no further recursion is performed. This condition is also explicitly considered in the base case, i.e., the one that considers all the feasible combinations, once the size of all matrices but the last one are set. Lastly, regarding the first recursive step, i.e., the dimensions of the first matrix, C_{max} equals $N_{\text{FEC}} - N_M + 1$ and R_{min} is 1.

Those conditions make the number of feasible solutions go down, as can be observe in Table 4.2. In Section 4.4, the impact of the reduction of the space solution in the optimization procedure is carefully analyzed.

4.3.2 Proposed hybrid simulated annealing metaheuristic

4.3.2.1 Description of the base methods

- **Simulated annealing**

SA is a metaheuristic for providing sufficiently good solutions to optimization problems. In particular, it is broadly utilized to approximate the optimal solution to nonlinear combinatorial optimization problems, where there exists a global minimum among several local minima. It is especially useful for problems with large solution space, limited computation capacity or hard time restrictions. Moreover, it is highly prone to problem-specific adaptations.

It was introduced in 1983 by Kirkpatrick et al. [96] as an emulation of the physical process through which a molten metal is slowly cooled so when the minimum temperature is reached, this happens at a minimum energy configuration.

The basic iteration of the SA procedure involves randomly selecting a feasible solution, s_n , out of the set of neighbors of the current solution, s_c , and moving to it with some probability. The move to this new solution always takes place (the probability of moving equals one) if its associated cost, $D(s_n)$, is lower than that of the current solution, $D(s_c)$, (downhill move) and with a probability lower than one if it is higher (uphill move). Thanks to allowing these uphill moves, the heuristic can escape from local minima and progress toward more suitable solutions. The probability of acceptance of a more-costly solution, P_a , depends on the difference of cost between the two solutions and on a global control parameter called temperature, T , originally and typically expressed as follows:

$$P_a = \exp\left(\frac{D(s_c) - D(s_n)}{T}\right) \quad (4.10)$$

So, the lower the difference of cost is and the higher the temperature is, the higher it becomes the probability of accepting the new solution.

The algorithm consists of an outer loop and an inner loop. The outer loop is basically employed to update the temperature. The temperature is initialized to some value, T^{init} , greater than zero, and is decreased at each step conforming to a given annealing schedule.

Every inner loop carries out a series of basic iterations as the one described before to explore the solution space. Within each iteration, solution s_n is selected from the ones that make up the neighborhood of solution s_c . The neighbors of a solution are the solutions in the solution space fulfilling some conditions. SA is originally a local search method, and therefore, the neighborhood of a solution is made up of the solutions that are located closest to it in the solution space, regarding a given definition of distance. However, when facing large solution spaces and strict time constraints, this approach encounters difficulties in finding a good-enough solution, as the solution space cannot be properly inspected. In order to overcome this problem, some authors have proposed to use different, more flexible neighborhood structures [97, 98]. In this regard, our proposal uses an approach based on varying scale neighborhood structures. In this way, the number of neighbors changes throughout the performance of the procedure, benefiting it regarding the pursued objective at each stage.

• Tabu search

TS is a metaheuristic search method proposed by F. Glover in 1986 [99] for solving combinatorial optimization problems. It employs flexible memory to avoid being trapped at local minima, either by forbidding or penalizing moves that would return to already visited solution.

This procedure generates neighborhoods of the current solution in accordance to a set of rules and banned solutions included in the so-called tabu lists. A neighborhood is used to search for new solutions to move to. If all the solutions in a neighborhood are tested, a new one is utilized. This iterative method continues until a stop condition is met.

The memory structures are divided into three categories: short-, intermediate- and long-term, considering the scope of the rules and banned moves included in the lists:

- Short-term memory structures are used to prevent the algorithm from cycling, that is, from returning to recently visited regions of the solution space. To that end, recently visited solutions are registered in a tabu list.
- Intermediate-term memory structures are employed to implement so-called aspiration criteria. Those criteria are used to intensify the search of solutions in promising regions of the solution space. For that purpose, short-term prohibitions are relaxed by allowing certain moves previously included in the tabu list.

- Long-term memory structures are used for diversification, that is, to promote the search of solutions in new or scarcely explored regions of the solution space. The objective of these structures is then basically the opposite of that of the intermediate-term ones.

4.3.2.2 Optimization procedure

The proposed HSA combines both base metaheuristics as summarized in the pseudocode presented in Fig. 4.1 and Fig. 4.2. All variables and functions are described in the following paragraphs.

- **Temperature and annealing schedule**

The temperature follows a schedule that relies on the time restriction imposed on the optimization problem. The more time the metaheuristic has for solving the problem, the slower the temperature will decrease and the more inner loops there will happen. In the proposed procedure, the temperature decreases linearly with the outer loop iteration i in the following fashion:

$$T_i = \frac{(I^{\text{outer}} - i)}{(I^{\text{outer}} - 1)} \cdot T^{\text{init}} \quad (4.11)$$

where T_i is the temperature at iteration i , and I^{outer} is the total number of iterations of the outer loop. The latter variable is described in detail in the following point.

Deriving a good initial value for the temperature, T^{init} , is a crucial but usually hard task, as it requires an approximate knowledge of the cost of the feasible solutions. If the temperature value is too high compared to the cost of the solutions, the probability of acceptance of new moves to more-costly solutions, P_a , will remain very close to one. Therefore, pretty much every move would be accepted and the procedure will result in a fairly erratic system. On the contrary, if the temperature takes a very low value, compared to the cost of the solutions, P_a will be very close to zero. In this situation, almost all moves to more-costly solutions will be rejected, thus becoming an nearly pure greedy algorithm. One common method to acquire information on the cost of the feasible solutions is through a random sampling of the solution space. However, this process introduces extra latency. In our scheme, T^{init} is directly set to equal the distortion that is obtained when employing the standard Pro-MPEG COP3 for protecting the data packets in the block (a configuration made up of only one solution, s_1^{opt}). So:

$$T^{\text{init}} = D\left(s_1^{\text{opt}}\right) \quad (4.12)$$

This initial value for the temperature is a good starting point to the procedure, since this value is in the same order of magnitude as the distortion associated with the configurations in the solution space. In addition, in this way, T^{init} is always set automatically at the beginning of the process.

- **Number of iterations of the outer loop**

```

// Subproblem with  $N_M$  equal to 1
 $N_M \leftarrow 1$ 
 $T^{\text{init}} \leftarrow s_1^{\text{opt}}$ 
terminate_procedure  $\leftarrow$  False
while not terminate_procedure
    // New subproblem
    // Subproblem global parameters
     $N_M \leftarrow N_M + 1$ 
     $F_{\Omega_{N_M}} \leftarrow []$ 
    update  $d_{N_M}^{\text{init}}$ 
    // First outer iteration
     $T \leftarrow T^{\text{init}}$ 
     $d_{N_M} \leftarrow d_{N_M}^{\text{init}}$ 
     $s \leftarrow \text{random}(\Omega_{N_M})$ 
    update  $N(s, d_{N_M})$ 
     $s^{\text{best}} \leftarrow s$ 
    include  $s$  in  $F_{\Omega_{N_M}}$ 
    compute  $I^{\text{inner}}$ 
    // Inner loop
    do inner_loop
    // Following outer iterations
    compute  $I^{\text{outer}}$ 
    if  $I^{\text{outer}} < 2$ 
        terminate_procedure  $\leftarrow$  True // Maximum time reached
         $N_{M_{\max}} \leftarrow N_M$ 
    end if
    // Outer loop
     $i \leftarrow 2$ 
    while  $i \leq I^{\text{outer}}$ 
        update  $T$ 
        update  $d_{N_M}$ 
         $s \leftarrow s^{\text{best}}$ 
        update  $N(s, d_{N_M})$ 
        compute  $I^{\text{inner}}$ 
        // Inner loop
        do inner_loop
        update  $I^{\text{outer}}$ 
         $i \leftarrow i + 1$ 
    end for
    // Store subproblem optimal configuration
     $s_{N_{M_{\max}}}^{\text{opt}} \leftarrow s^{\text{best}}$ 
    // Check whether following subproblem possible
    estimate  $t'_{\text{it}_{N_M+1}}$ 
    if  $t'_{\text{it}_{N_M+1}} < (t_T - t_c)$ 
        terminate_procedure  $\leftarrow$  True // Maximum time reached
    end if
end while
    // Obtain overall optimal configuration
     $s^{\text{opt}} \leftarrow \min_{s_1^{\text{opt}}, s_1^{\text{opt}}, \dots, s_{N_{M_{\max}}}^{\text{opt}}} D(s)$ 
    
```

Figure 4.1: Pseudocode of the proposed HSA procedure. The lines belonging to the inner loop are presented in Fig. 4.2.

```

for  $j$  from 1 to  $I^{\text{inner}}$ 
    if  $C(s)$  equal  $\emptyset$ 
        break
    end if
     $s' \leftarrow C(s)$ 
    if  $\left[ D(s') < D(s) \right]$  or  $\left[ \text{random}([0, 1]) \leq \exp\left(\frac{D(s) - D(s')}{T}\right) \right]$ 
         $s \leftarrow s'$ 
        update  $N(s, d_{N_M})$ 
        if  $D(s) < D(s^{\text{best}})$ 
             $s^{\text{best}} \leftarrow s$ 
        end if
    end if
    include  $s$  in  $F_{\Omega_{N_M}}$ 
end for
    
```

Figure 4.2: Pseudocode of the inner loop of the proposed HSA procedure.

In the first place, every time a subproblem is solved, an estimation of the time that will potentially take an outer iteration of the following subproblem, $t'_{\text{it}_{N_M+1}}$, is carried out to check whether there is still time to complete at least one outer iteration of that subproblem. In the case that $t'_{\text{it}_{N_M+1}}$ is greater than the remaining time, the following subproblem will be posed and started. $t'_{\text{it}_{N_M+1}}$ is computed as follows:

$$t'_{\text{it}_{N_M+1}} = t_{\text{it}_{N_M\max}} \cdot \frac{\tilde{N}_{\Omega_{N_M+1}}}{\tilde{N}_{\Omega_{N_M}}} \quad (4.13)$$

where $t_{\text{it}_{N_M\max}}$ is the maximum time taken by any of the outer iterations of the subproblem just solved, $\tilde{N}_{\Omega_{N_M+1}}$ is the number of feasible solutions to the following subproblem, and $\tilde{N}_{\Omega_{N_M}}$ is the number of feasible solutions to the subproblem that has just been solved.

Later on, if the following subproblem can be posed, we compute the number of iterations of the outer loop, I^{outer} , as:

$$I^{\text{outer}} = \min \left(\left\lfloor \frac{(t_T - t_c)}{t_{\text{it}_{N_M+1}}} \right\rfloor, I_{\max} \right) \quad (4.14)$$

where t_T relies on the imposed time restriction, which will in turn depend on the maximum allowed extra latency regarding the service, t_c is the time that has been consumed so far, which basically includes the time spent solving previous optimization subproblems, and $t_{\text{it}_{N_M+1}}$ is the time consumed by a the more lasting outer loop iteration. This value initially equals the time estimated before, $t'_{\text{it}_{N_M+1}}$, but is updated after each complete outer loop with the time used by the longest outer iteration of the current problem so far. On the other hand, $I_{\max}$ is set to a constant number, so as to establish a maximum duration to the process. In the case that I^{outer} is found to be 0 or 1, the process will

terminate immediately. Otherwise, I^{outer} is updated after each complete outer loop.

• Neighbors of a solution

Before discussing the neighborhood of the current solution, let us define the distance between two configurations made up of N_M matrices $s_1 = \{C_{s_1,1}, R_{s_1,1}, \dots, C_{s_1,N_M}, R_{s_1,N_M}\}$ and $s_2 = \{C_{s_2,1}, R_{s_2,1}, \dots, C_{s_2,N_M}, R_{s_2,N_M}\}$ as:

$$d_{N_M}(s_1, s_2) = \left\{ \sum_{m=1}^{N_M-1} \left[(C_{s_1,m} - C_{s_2,m})^2 + (R_{s_1,m} - R_{s_2,m})^2 \right] \right\}^{\frac{1}{2}} \quad (4.15)$$

In the proposed procedure, the neighbors of a solution s , $N(s, d_{N_M})$, are all feasible solutions in the solution space Ω_{N_M} within a distance d_{N_M} , i.e.:

$$N(s, d_{N_M}) = \left\{ s' \in \Omega_{N_M} \mid d_{N_M}(s, s') \leq d_{N_M} \right\} \quad (4.16)$$

where distance d_{N_M} is updated at the beginning of each outer loop iteration to smoothly switch from the pseudorandom solution sampling strategy of the first iterations to the greedy strategy of the last ones. It takes new values according to the same schedule as the temperature (see (4.11)), in the following fashion:

$$d_{N_M} = \frac{(I^{\text{outer}} - i)}{(I^{\text{outer}} - 1)} \cdot d_{N_M}^{\text{init}} \quad (4.17)$$

where $d_{N_M}^{\text{init}}$ is the value that is used to initialize the distance. This value is set to the potential greatest distance between any two solutions in Ω_{N_M} , which is computed as follows:

$$d_{N_M}^{\text{init}} = \left\{ \sum_{m=1}^{N_M-1} \left[(C_{m,\max} - C_{m,\min})^2 + (R_{m,\max} - R_{m,\min})^2 \right] \right\}^{\frac{1}{2}} \quad (4.18)$$

where:

$$\begin{aligned} C_{m,\max} &= \max \left(C_{s',m} \mid s' \in \Omega_{N_M} \right) & C_{m,\min} &= \min \left(C_{s',m} \mid s' \in \Omega_{N_M} \right) \\ R_{m,\max} &= \max \left(R_{s',m} \mid s' \in \Omega_{N_M} \right) & R_{m,\min} &= \min \left(R_{s',m} \mid s' \in \Omega_{N_M} \right) \end{aligned}$$

In this way, at least in the first iteration, all solutions in Ω_{N_M} can be reached from any other solution.

• Number of iterations of the inner loop

The number of iterations of the inner loop, I^{inner} , represents the number of neighbor solutions visited within an iteration of the outer loop. It is set to the maximum value of two options. The first one is the percentage of neighbors of the starting solution s in the current outer loop iteration i , τ . The second one equals the number of neighbors of s within a distance of $\sqrt{N_M}$, i.e., all those that surround s . Therefore:

$$I^{\text{inner}} = \max \left(\tau \cdot N(s, d_{N_M}), N(s, \sqrt{N_M}) \right) \quad (4.19)$$

In this way, during the first iterations of the outer loop, a significant but not excessive number of configurations in the solution space can be sampled. When the first number becomes too low, the second option prevails, guaranteeing a minimum number of candidates to be checked per outer iteration with which to carry out a greedy algorithm.

• Memory

Two types of memory structures are used in this procedure, as a legacy of TS: a short-term one, and an intermediate-term one.

The short-term memory structure basically stores all the solutions visited during the course of the process, which are included in a forbidden move list, $F_{\Omega_{N_M}}$, so that they are not tested again.

On the other hand, the intermediate-term memory structure stores just one value, the best solution found during the course of an internal loop, $s_{N_M}^{\text{opt}}$. This solution is used as the starting solution for the following inner loop, as a means to intensify the search of solutions in promising regions of the solution space.

Long-term memory structures are not necessary, as a type of diversification is actually performed during the first outer iterations of the procedure, thanks to the management of neighborhood structures.

• Candidate solutions

If s is the current solution, candidate solutions $C(s)$ are all the solutions in the solution space the system can move to from s , that is, those ones that are neighbors of s and have not been visited yet. This is expressed next:

$$C(s) = \left\{ s' \in N(s, d_{N_M}) \wedge s' \notin F_{\Omega_{N_M}} \right\} \quad (4.20)$$

4.3.3 Summary

With the proposed parameter values and temperature schedule, three main useful results are sought and achieved:

1. To fulfill time requirements, thanks, first, to only posing those subproblems that can be dealt with in the considered time, and second, to adapting the number of outer iterations in regard of this time.
2. To perform a pseudorandom sampling during the first iterations of the outer loop (so that the whole solution space is inspected to some extent), thanks to a wise selection of the initial values for the temperature and distance, and of the number of inner loop iterations, and to adjusting the search area (neighborhood) throughout the procedure regarding different the needs at each stage.

R (Mbps)	T_L (s)	T_R (s)	r_{FEC}	N_P	N_{FEC}
4	1	0.5	10/11	185	19
4	1	0.5	5/6	37	37
4	0.2	0.1	10/11	185	4
4	0.2	0.1	5/6	37	7
8	1	0.5	10/11	185	19
8	1	0.5	5/6	37	37
8	0.2	0.1	10/11	185	4
8	0.2	0.1	5/6	37	7
12	1	0.5	10/11	185	19
12	1	0.5	5/6	37	37
12	0.2	0.1	10/11	185	4
12	0.2	0.1	5/6	37	7

Table 4.3: System characterizing variables: the procedure parameters N_P and N_{FEC} are estimated from the other four parameters.

3. To favor an efficient greedy strategy for the last iterations, thanks to a favorable annealing schedule, to, again, an appropriate selection of the number of inner loop iterations and of the neighborhood structures, and finally to the inclusion of memory structures.

4.4 Experiments and results

Aiming at evaluating the effects of both the solution space reduction and the proposed HSA procedure, an exhaustive set of experiments were carried out. In these tests, three video sequences of dissimilar bitrate, R , were considered: $R_1 = 4$ Mbps (Seq1), $R_2 = 8$ Mbps (Seq2) and $R_3 = 12$ Mbps (Seq3). Moreover, two values were used to limit FEC latency, T_L : 200 ms and 1 s, which correspond approximately to typical constraints imposed on live event and non-live event broadcasts, respectively. As XOR operations are considered negligible in terms of time, roughly the whole delay introduced by standard Pro-MPEG COP3 codes is due to the time that the receiver needs to wait for all the source and repair packets that belong to the same matrix, so that decoding operations can be carried out [60, 57]. However, in the case of the UEP version, the procedure to obtain the optimal configuration also introduces some extra delay. So, with the aim of always complying with the imposed time restriction, it becomes necessary to split the time that is devoted to FEC-related operations between the receiver, T_R , and the transmitter, T_T , so that there is time to wait for the packets, as in the standard case, and the optimization procedure can be carried out. In these experiments, half of the FEC time is used by the receiver and the other half by the transmitter. So, approximately only half of the data packets are handled at each step of the UEP version, when compared to the standard one. This fact is detrimental to the obtained results, as, generally, the

N_P	N_{FEC}	N_M	$N_{\Omega_{N_M}}$	$\tilde{N}_{\Omega_{N_M}}$
185	19	2	590	85
37	37	2	638	90
185	4	2	63	18
37	7	2	79	15
185	19	3	154921	3887
37	37	3	191941	3999
185	4	3	1207	81
37	7	3	2384	121
185	19	4	24045652	93752
37	37	4	35985286	106826
185	4	4	7140	378
37	7	4	36227	427

Table 4.4: Cardinality of the solution space, both original and restricted.

efficiency of FEC codes (and Pro-MPEG COP3 codes are no exception) increases with the number of data packets that are handled together. However, the fact of exploiting the unequal importance of video packet not only counteracts this drawback in most scenarios, but delivers better results in terms of overall distortion, as will be shown. Finally, two different channel code rates, r_{FEC} , were employed: $r_{\text{FEC}_1} = 10/11$ (10 % overhead) and $r_{\text{FEC}_2} = 5/6$ (20 % overhead). Table 4.3 shows these system characterizing variables, as well as an approximation of the values of parameters N_P and N_{FEC} , which entirely rely on them.

Two sets of experiments were carried out to assess three different aspects of the proposal: the solution space reduction, the performance of the proposed HSA procedure, and the adaptive capacity of the algorithm to the imposed time restrictions, jointly with its effectiveness under those conditions. The first two aspects were assessed in terms of processing time (with a 2-core CPU clocked at 3 GHz with 12 GiB RAM), and resulting distortion in terms of peak-to-noise rate (PSNR). The latter aspects were evaluated in terms of processing time (to validate adaptability; with the same CPU) and overall distortion, again in terms of PSNR. For all the experiments, a partial-context-aware packet-level distortion model was employed to obtain the importance of source packets. In this model, the distortion associated with the transmission of a given data packet equals the approximate percentage of pixels in the GOP that are potentially affected by its loss. This value is calculated as the estimated number of packets in the GOP (in the frame it belongs to and in the ones that use that frame as reference) whose information depend on that of the packet in question for decoding. Despite constituting a rather limited and simple distortion model, it suitably reflects two main aspects of how video streams behaves when impacted by packet losses: inter-frame error propagation and

N_P	N_{FEC}	N_M	Exhaustive search (not restricted)		Exhaustive search (restricted)		HSA (restricted)	
			Time (s)	PSNR (dB)	Time (s)	PSNR (dB)	Time (s)	PSNR (dB)
185	19	2	1.03E-02	28.79	7.82E-04	28.79	6.69E-04	28.79
37	37	2	6.94E-03	33.0	4.74E-04	33.0	4.18E-04	33.0
185	4	2	2.84E-04	28.46	7.74E-05	28.46	1.13E-04	28.46
37	7	2	2.06E-04	33.21	7.05E-05	33.21	8.84E-05	33.2
185	19	3	1.64	28.87	5.07E-02	28.87	6.92E-03	28.86
37	37	3	1.62	33.14	3.76E-02	33.14	4.92E-03	33.11
185	4	3	6.5E-03	28.53	4.28E-04	28.53	4.22E-04	28.53
37	7	3	9.31E-03	33.32	5.56E-04	33.32	4.9E-04	33.32
185	19	4	311.91	28.9	1.33	28.9	6.22E-02	28.89
37	37	4	380.24	33.17	1.06	33.17	6.23E-02	33.14
185	4	4	3.22E-02	28.53	2.15E-03	28.53	2.19E-03	28.53
37	7	4	2.15E-02	33.34	2.49E-03	33.34	2.59E-03	33.34

Table 4.5: Results of the first set of experiments.

intra-frame desynchronization, and provides a relative hierarchization of data packets that suffices for properly illustrating the key factors that impact on the performance of the strategy. This model is utilized again in Chapter 5, where it is fully formalized.

In the first set of experiments, sequence Seq1 was used, the number of iterations of the outer loop, I^{outer} , was set to 10 (no time restrictions were imposed), and the subproblems dealing with 2 to 4 matrices were considered. Simulations were run 100 times and the obtained results averaged afterwards.

As can be observed in Table 4.4, the conditions introduced to restrict the solution space absolutely make the number of configurations go down, whereas barely affecting the result. Thus, the optimization procedure starts from a more advantageous situation. Moreover, in Table 4.5, it can also be seen that the proposed procedure is able to find sufficiently good solutions in a significantly shorter period of time, at the expense of a very small error, almost negligible in a logarithmic scale, which slightly increases with the size of the solution space. In particular, the larger the solution space is, the greater it is the gain in terms of time.

The second set of experiments were aimed at validating the proposed strategy when strict time restrictions apply. In these experiments, Seq2 and Seq3 were employed, and I^{outer} was set to vary according to (4.14). Moreover, a simplified Gilbert-Elliot model was used to model the behavior of the channel, which is described through two parameters: the packet loss rate (PLR) and the average burst length (ABL). In these simulations, four different PLR and three ABL values were considered. The ABL values are given in terms of time, which translate into different numbers of consecutive loss packets, regarding the bitrate of the video stream. The PLR values have been chosen in the region where

Figure 4.3: Results of the second set of experiments in terms of distortion when the whole imposed FEC latency equals 200 ms. The results of the standard codes using the whole imposed FEC latency (200 ms) are presented using solid lines. The results of the standard codes using half of the imposed FEC latency (100 ms) are presented using dotted lines in different colors. The results of the UEP version of the codes optimized through the proposed HSA procedure are presented using dashed lines in different colors.

standard Pro-MPEG COP3 codes start presenting a weaker performance, which thus correspond to our points of interest. Again, simulations were run 100 times and the obtained results averaged afterwards.

The results of these experiments regarding the introduced overall distortion are shown in Fig. 4.3 and Fig. 4.4. These plots present the distortion introduced when using the standard codes using the whole imposed maximum FEC time (which is used as reference), and the distortion introduced when using the UEP version of the codes optimized through the proposed metaheuristic, where half of the imposed maximum FEC latency is devoted to carry out the algorithm and the other half at the receiver. Results presenting the distortion value obtained when using the standard codes employing the whole available FEC time and that obtained when using only half of it are also included for comparison. In particular, this comparison allows on the one hand to capture the detrimental effect mentioned before, and, on the other hand, to directly measure the benefits of employing the UEP version of the codes with respect to the standard when employing the same protection block size. As already mentioned, distortion values are provided in terms of PSNR. The results corresponding to $T_L = 200$ ms are depicted in Fig. 4.3, whereas the ones corresponding to $T_L = 1$ s are presented in Fig. 4.4.

As can be seen, the effectiveness of the proposed procedure increases with the capacity

Figure 4.4: Results of the second set of experiments in terms of distortion when the whole imposed FEC latency equals 1 s. The results of the standard codes using the whole imposed FEC latency (1 s) are presented using solid lines. The results of the standard codes using half of the imposed FEC latency (500 ms) are presented using dotted lines in different colors. The results of the UEP version of the codes optimized through the proposed HSA procedure are presented using dashed lines in different colors.

to generate configurations that are capable of decorrelating channel error, that is, with the number of data and repair packets per protection block, and inversely with the length of error bursts. Specifically, the greater the ABL of the communication channel is, the more columns are required to provide suitable interleaving and so decorrelate error. This means that in the scenarios where it is not possible to generate configurations where matrices have a sufficient number of columns to cope with channel's ABL, that is, where T_L (and therefore the number of data and repair packets per block) is low in comparison with the existing ABL value, the results obtained when applying the adopted UEP scheme cannot outperform those obtained with the standard codes, as none of the feasible configurations can properly deal with burst errors (see Fig. 4.3). On the other hand, if T_L is high enough compared to channel's ABL, there can be found configurations that can correctly cope with error bursts. In this case, results are less dependent with the particular ABL value, as can be extracted from the results of the experiments where $T_L = 1$ s (see Fig. 4.4), where figures are flatter. T_L is in this case sufficiently high in comparison with the considered error burst lengths: 1, 3 and 5 ms. Furthermore, if the channel's PLR value increases, the likelihood of two or more error bursts impacting the same matrix and moreover affecting packets in the same column increases as well. In this situation, the proposed UEP strategy is capable of reducing the number of rows of the matrices where

the most important data packets are to be arranged (and so decrease the code rate and therefore increase the protection capability of the code), and so make that likelihood lower for the more important packets, thus obtaining a better overall distortion results than those of the standard strategy, even for low T_L values. This circumstance can be seen in both Fig. 4.3 and Fig. 4.4, where the difference in dBs between the UEP and the standard versions remains fairly constant, despite the PLR increase, which indicates that the relative difference increases. Obviously, in the scenarios where the PLR is particularly high, those extra dBs might not make a real difference to the user (e.g. if $PLR = 2.5E-2$). However, there exist a considerably wide interval of PLR values where this disparity will make the difference between an acceptable and an unacceptable viewing experience.

So, with good-enough conditions, the UEP codes optimized using HSA can easily obtain a PSNR increase of up to several dBs with respect to the standard approach, depending on the scenario. Moreover, it is worth noticing that results also show that the gap between applying the standard codes using the whole FEC period and using only half of it decreases with the capacity of being able to decorrelate error.

As can be seen, the effectiveness of the proposed procedure increases with the capacity to generate configurations where the number of columns per matrix is great enough to sufficiently decorrelate channel error, that is, with the number of data and repair packets per protection block, and inversely with the length of error bursts. So, with good-enough conditions, the UEP codes optimized using HSA can easily obtain up to 10/20+ % distortion decrease with respect to the standard scenario, depending on the video and FEC rate. Moreover, based on the trends observed in [7], the use of distortion models of the packet stream that more accurately capture the unequal importance of data packets will lead with high probability to deliver even better results. Furthermore, the performance of the HSA procedure improves with respect to the standard case with the PLR, especially when time conditions are stricter. This conclusion is of special interest to us, as the main motivation of using the UEP version of the Pro-MPEG COP3 codes is to improve the performance at high PLRs, where the standard ones are weaker. Finally, results also show that the gap between applying the standard codes using the whole FEC period and using only half of it decreases with the capacity of being able to decorrelate error. However, the latter never outperform the former, unless the UEP version is applied.

To verify the adaptability to these constraints and the performance of the scheme, both the average time spent on carrying out the optimization (all mean, maximum and variance values are included) and the average number of subproblems that it was possible to pose, given those time constraints. Finally, in the same way as before, simulations have been run 100 times and the results averaged afterwards.

As can be seen in Table 4.6, the algorithm is able to adapt on the fly to strict time conditions and comply with them. In most cases, the procedure uses the most part of the available FEC time. Nevertheless, in some circumstances, like that in which Seq3 is protected allowing a FEC latency of 1 s, the time that is finally spent in applying the metaheuristic is not close to the limit. This means that at the end of one of the subproblems (in this example, the third one), the time that one outer iteration of the following

R (Mbps)	r_{FEC}	T_{L} (s)	T_{T} (s)	Time (s)			$N_{M_{\text{max}}}$
				Mean	Maximum	Variance	
8	10/11	0.2	0.1	8.84E-02	9.83E-02	1.48E-04	5.96
8	5/6	0.2	0.1	9.12E-02	9.97E-02	2.42E-04	5.67
8	10/11	1	0.5	3.74E-01	4.85E-01	1.95E-02	4.95
8	5/6	1	0.5	3.93E-01	4.86E-01	1.42E-02	4.5
12	10/11	0.2	0.1	9.30E-02	9.91E-02	1.52E-04	3.85
12	5/6	0.2	0.1	6.79E-02	9.95E-02	9.75E-04	3.91
12	10/11	1	0.5	4.24E-02	6.29E-02	5.09E-05	3
12	5/6	1	0.5	3.58E-02	5.66E-02	4.58E-05	3

Table 4.6: Results of the second set of experiments in terms of processing time.

subproblem is estimated to last exceeds the time that is left (see (4.13)). Therefore, the procedure ends, even if there is some FEC time left. Finally, as expected, the average number of subproblems that can be posed and started per block along the sequence increases inversely with the number of feasible solutions in the solution space, that is, with the number of data and repair packets per FEC block.

Finally, we assess the adaptability of the scheme to the specifics of the scenario. The strategy proposed in this paper potentially provides a different solution for each group of consecutive data packets along the stream, mostly depending on the relative importance of the data packets included in each protection block. This is the reason why unique overall solutions are not provided as a result of the transmission of an encoded sequence under certain conditions. However, an illustrative example is included to fully clarify and characterize the performance of the protection strategy. This example considers the protection of the first data packets in the corresponding stream. The scenario taken into account is as follows: $T_{\text{L}} = 200$ ms, Seq2, r_{FEC_2} , $\text{PLR} = 1.0\text{E-}2$, $\text{ABL} = 1$ ms. So, the first $N_{\text{P}} = 74$ data packets in the stream are arranged for protection in the first block and $N_{\text{FEC}} = 15$ are to be generated. First, the standard strategy is considered. After this, subproblems are posed sequentially until the available time is over. Table 4.6 shows some important results of each subproblem: the final configuration and the relative distortion with respect to the standard configuration. Additionally, Table 4.8 shows the resulting error rate per matrix. In this case, the presented configuration with five matrices is the best solution that could be found, and so it is the one used to arrange data packets and generated packets. The most important 21 data packets are arranged in the first matrix and seven parity packets are generated. The following 16 most important packets are arranged in the second one and four repair packets are generated. And so on. The resulting error rate per matrix considers the probability of not receiving a packet and also not being able to recover it. The presented value is the average of all the data packets in each matrix.

Finally, with the aim of showing the adaptive nature of our proposal, we present, for this same scenario and a second one with quite different characteristics ($T_{\text{L}} = 1$ s, Seq3, r_{FEC_2} , $\text{PLR} = 7.5\text{E-}5$, $\text{ABL} = 3$ ms), a histogram that shows the percentage of times a

N_M	Final solution	Distortion (normalized)
1	$[15 \times 5]_1$	1
2	$[13 \times 4]_1, [2 \times 11]_2$	0.84
3	$[9 \times 3]_1, [5 \times 6]_2, [1 \times 17]_3$	0.78
4	$[7 \times 3]_1, [5 \times 4]_2, [2 \times 7]_3, [1 \times 19]_4$	0.77
5	$[7 \times 3]_1, [4 \times 4]_2, [2 \times 6]_3, [1 \times 9]_4, [1 \times 16]_5$	0.76
6	$[8 \times 3]_1, [2 \times 4]_2, [2 \times 5]_3, [1 \times 6]_4, [1 \times 9]_5, [1 \times 17]_6$	0.78
7	$[4 \times 3]_1, [3 \times 3]_2, [3 \times 4]_3, [2 \times 5]_4, [1 \times 6]_5, [1 \times 8]_6, [1 \times 17]_7$	0.8

Table 4.7: Example of the performance of the proposed strategy. $[C_k \times R_k]_k$ means that the k^{th} matrix in the configuration has R_k rows and C_k columns.

N_M	Resulting error rate per matrix
1	$[4.9\text{E-}4]_1$
2	$[3.94\text{E-}4]_1, [10.47\text{E-}4]_2$
3	$[2.97\text{E-}4]_1, [5.85\text{E-}4]_2, [15.71\text{E-}4]_3$
4	$[2.97\text{E-}4]_1, [3.94\text{E-}4]_2, [6.79\text{E-}4]_3, [17.38\text{E-}4]_4$
5	$[2.97\text{E-}4]_1, [3.94\text{E-}4]_2, [5.85\text{E-}4]_3, [8.65\text{E-}4]_4, [14.85\text{E-}4]_5$
6	$[2.97\text{E-}4]_1, [3.94\text{E-}4]_2, [4.9\text{E-}4]_3, [5.85\text{E-}4]_4, [8.65\text{E-}4]_5, [15.71\text{E-}4]_6$
7	$[2.97\text{E-}4]_1, [2.97\text{E-}4]_2, [3.94\text{E-}4]_3, [4.9\text{E-}4]_4, [5.8\text{E-}4]_5, [7.73\text{E-}4]_6, [15.7\text{E-}4]_7$

Table 4.8: Example of the performance of the proposed strategy. $[P_k]_k$ is the resulting error rate of the data packets protected through matrix k with the dimensions indicated in Table 4.7

Figure 4.5: Histograms showing how many times (in percentage) a configuration of a given number of matrices is selected.

configuration of each number of matrices is selected as the best option during the transmission of the whole sequence. The first video stream consists of 641580 packets, that is, 8670 blocks of $N_P = 74$ data packets, whereas the second one is made up of 639400 packets, i.e., 1150 blocks of $N_P = 556$ data packets. Such histograms are depicted in Fig. 4.5. It can be observed that, in the first scenario, the protection strategy opts more often for configurations made up of six matrices, followed by configurations of five and seven. In the case of the second scenario, the protection technique almost always selects configurations of two or three matrices, with greater preference for the latter. The selection of a given configuration is, as already seen, the result of a number of elements: available time, channel behavior, characteristics of the transmitted video, etc. Scenarios with very strict time limitations and large block sizes, and/or where packets are rather similar in terms of the distortion that they might cause if they are lost (e.g. encoded sequences made up of short closed GOPs) tend to select configurations of few matrices. On the other hand, if there is enough time to pose and solve the necessary subproblems, and the distortion associated with the different packets is sufficiently unequal (e.g. long GOPs with hierarchical B-frames), the protection mechanism tends to select configuration with more matrices.

4.5 Conclusions

In this chapter, a metaheuristic stemming from the hybridization of two powerful optimization techniques, simulated annealing (SA) and tabu search (TS), has been presented. This optimization method is situated in the context of the UEP mechanisms that aim at smartly protecting video transmission over IP networks when providing very time-sensitive services. In particular, the presented procedure aims at finding near-optimal protection configurations of the UEP version of the Pro-MPEG COP3 codes -which allow the utilization of a potentially very large number of configurations- under strict time restrictions.

The proposed method consists of two steps. In the first one, the solution space is significantly reduced through imposing UEP-aware restrictions. In the second one, SA's core procedure is modified to take into account imposed time restrictions and explore the solution space considering a proposed definition of distance between configurations that allows a periodic update of neighbor structures. Moreover, it incorporates memory structures from TS to more finely channel the search of new solutions.

A wide number of simulations have been carried out and grouped in two different sets of experiments, so as to assess the performance of the proposed method. The first set of experiments validates the reduction of the solution space and the performance of the proposed procedure. The second one verifies that the HSA method is able to invariably fulfill real-time constraints, and simultaneously find sufficiently good protection configurations that allow the UEP Pro-MPEG COP3 codes to clearly outperform the standard codes in a wide range of scenarios. Additionally, some extra results have been included to better show the adaptive nature of our proposal.

The work presented in this chapter has been included in two publications. A first

version of the proposed metaheuristic was presented in [9]. This initial version focused on the proposed conditions for reducing the number of feasible configurations in the solution space and included the varying scale neighborhood structures. A complete version of the optimization approach was presented in [10]. This final version includes all the modifications performed on the core procedures of simulated annealing and tabu search to consider imposed time restrictions and explore the solution space as desired.

Chapter 5

Distortion model

5.1 Introduction

This chapter deals with the differentiation of video data units regarding the unequal degradation that these elements can potentially cause in the decoded sequence in case of loss, as a means to refine QoS management mechanisms and thus improve their effectiveness.

There exist a wide range of proposals in the literature addressing data differentiation and prioritization in the scope of video transmission. These strategies fundamentally differ in terms of granularity, that is, the level at which they operate, and in how data elements are classified. With respect to the former, the operational level determines the data structures in the bitstream that are taken as basic working units, i.e., the entities whose importance is considered as a whole, and over which decisions are made later on. Regarding the latter, data prioritization can be carried out either in a rigid fashion, i.e., regarding pre-established categories, or in a more flexible way, as a result of a distortion model that considers potentially changing characteristics of the encoded video stream or other actors of the system, that is, the possible variation in relevance of different elements in the packet stream.

Regarding basic working units, traditionally encoded 2D video sequences (e.g. in AVC [100] or HEVC [101]) are usually managed in terms of the characteristics of their frames or slices, as can be seen in [102] or [103], where decisions are made at frame level, and in [104] or [105], which operate at slice level. Considering layered video sequences (e.g. encoded in H.264/MPEG-4 Scalable Video Coding -SVC- [47] or H.265/MPEG-H Scalable High Efficiency Video Coding -SHVC- [106]), data prioritization is most frequently carried out at layer level. The strategies proposed in [48] and [71] very well exemplify this approach. Moreover, data of 3D video sequences (e.g. encoded using the standard extensions H.264/MPEG-4 Multiview Video Coding -MVC- [107] or Multiview High Efficiency Video Coding -MV-HEVC- [106]) is commonly handled regarding the view it belongs to, as illustrated in the proposal in [108]. Additionally, there exist a number of strategies that perform data prioritization at lower levels. For instance, the

proposals in [109] and [110] analyze and classify the MBs in a frame regarding their motion activity. Or the strategies considered in [111], [73] and [51], which very clearly exemplify how video data can be assessed and prioritized regarding the characteristics of the different data packets along the stream.

A general drawback of many proposals is that they work at levels that are either too coarse (e.g. layers, views...), what leads to making uniform decisions over sets of data that may present strong interdependence and thus of highly unequal importance, or too fine (e.g. MBs), making the procedure computationally considerably costly, intolerable if intended to perform in real time. In this regard, working at frame/slice level and at packet level is generally a more suitable choice, as these levels represent a better tradeoff between accuracy and practicality. Indeed, data packets are the working units of the communication channel and most upper-level QoS control mechanisms, and frames/slices are upper-level working elements of most video coding standards and extensions, as well as the upper-level units of information that are presented to users.

Regarding how data structures are classified, existing proposals move between immediacy and accuracy [112]. In the former end of the spectrum, data prioritization is carried out in regard to pre-established categories, without actually modeling the behavior of the packet stream [71, 113, 114]. This procedure introduces no extra delay. Nevertheless, as pointed out, it may be inaccurate, since it neither considers the specifics of the piece of data that is prioritized (e.g. content characteristics) nor its actual context (e.g. coding structure, packetization results, channel behavior). Strategies that opt for a greater degree of accuracy use distortion models to assess the importance of the different parts of the video. These models adapt to some degree to the specifics of the content and/or the actual context of the system to better reflect the real relevance of video data elements, hence enabling a smarter management. However, they highly differ in the degree of adaptation to these video characteristics. In some schemes, an analysis of the content of the sequence in question is carried out before performing data prioritization, whereas in others it is assumed that there is no access to the uncoded source and so their performance is based on elements that can be obtained from the already encoded (and likely encapsulated) stream. Usually, the first group of strategies are extremely accurate but are generally not suitable for time-sensitive applications or even possible in most transmission systems, where the module in charge of performing channel coding has no feedback from previous modules. Finally, the accuracy and good performance of the second group heavily depends on how good they manage to model the behavior of the stream. They are typically backed by sets of training examples, but they do not carry out any online analysis.

5.2 Distortion model

5.2.1 General description

A distortion model establishes a direct link between key features that characterize the data elements that make up an encoded video stream at a particular level and the distortions that affect them.

tion that will be potentially introduced in the decoded sequence presented to final users if these elements are not available at the decoder. Due to intra-frame coding and inter frame prediction, there exists a strong interdependence between elements. Therefore, their context is also necessarily considered, as the effect of losing a piece of data varies depending on whether other elements in the stream are or not available.

The connections that are established are the outcome of a process that consists of the following steps:

1. Study and analysis of the unequal effects that the loss of data elements with dissimilar features has on the decoded video sequence in terms of video quality degradation.
2. Extrapolation and generalization of the previously analyzed results.
3. Formalization of the link between features and estimated distortion.

In this process, three different data structure levels are considered. The first one is the basic working unit. As mentioned in the introduction, this is the entity whose importance is assessed, relative to others, and over which uniform decisions are made afterwards. The second one is the window of observation, which is the amount of information processed at the same step, i.e., the number of basic working units considered simultaneously. And the third one is the level at which the effect of losing working units on the decoded video is analyzed and modeled. This structure must be suitable to describe to a high extent how errors affect the quality of the video sequence presented to users. The analysis of the effect of losing basic working units is used to infer their importance in the window of observation. Moreover, the way in which the distortion introduced in the video sequence is measured is also of great importance.

5.2.2 Previous work

In the literature, there exists a variety of models that use frames/slices or network packets as working units. Those models, however, do not fully adapt to the requirements of our scenario or do not properly take into account some important characteristics of today's common encoded video streams. For instance, the majority of the proposed strategies do not even consider non-reference frames in their models, but only intra-coded and predicted pictures [115, 116]. Furthermore, a number of models assume that one single network packet can transport all the information relative to a whole frame/slice [117, 118]. This feature, which is key to those approaches, is definitively out of date, as nowadays most video sequences are of high quality and resolution and the bitstreams that result from encoding them have bitrates that in the majority of the cases lead to requiring more than one data packet per frame (specially for reference frames). Thus, as an extension to this approach, some strategies consider that a number of packets are used per frame. However, they assume that video sequences are encoded using a number of slices per frame that allows transporting all the information relative to one or several slices in one network packet [73, 105]. Again, this assumption is not realistic and definitively

not widely put in practice, among other reasons because it would require a noticeable increase of the bitstream overhead, which most providers are not willing to accept.

Additionally, most approaches consider non-interdependence-aware models, that is, models in which the relevance of a piece of information is estimated without considering the rest of the data in the window of observance, regardless of the possibility that these elements are interdependent [119, 115, 51, 73]. Thus, the distortion associated to the loss of a particular network packet is calculated considering that the possible loss of other packets does not affect that value, i.e., as if it was singly lost. This approach is more simple than those that consider interdependence to some extent, but is less accurate. On the other hand, interdependence-aware models take into account that there exists a correlation factor that considers the effect on the overall distortion of potentially simultaneously losing interdependent data packets. This factor can be positive, negative or zero, depending on the position that lost packets hold in the packet stream and the relation of dependency between the frames/slices they belong to. A positive correlation value means that there exists a multiplier effect derived from losing these packets, as described in [120] or [116]. This effect usually takes place when the lost packets belong to different interdependent frames/slices and if a content-dependent measurement of distortion is employed. If, on the contrary, the correlation factor is negative, the effect of losing one or more packets is masking the effect of losing other packets. This masking effect occurs, for instance, when the lost packets belong to the same frame, as described in Subsection 5.3. Finally, if the correlation factor is zero, the frames/slices these packets belong to are not interdependent. That is the case, for instance, of packets belonging to different non-hierarchical B-frames. The interdependence-aware models that have been proposed in the literature differ in the extent to which they consider this interdependence, since if entirely considered, the model can lead to unapproachable computations, particularly regarding time-sensitive scenarios.

5.2.3 Selection of the measurement of distortion

Due to time-related limitations and other restrictions, it is assumed that uncoded sequences cannot be accessed during the actual transmission process, and that the bitstream cannot be analyzed in depth to study coding-related outcomes throughout the sequence, like the resulting motion vectors and residuals, which would allow to infer to some extent some local and global features of the sequence content. Hence, it is not possible to have a rather acceptable approximation of the nature of that sequence. Therefore, we need to select a measurement that is simultaneously sufficiently independent from the content that is transported (and from methods or techniques whose result are influenced to some extent by that content) and accurate enough to properly model the impact of errors on the decoded sequence presented to users. In that sense, measurements like the Peak Signal-to-Noise Ratio (PSNR) or the Structural Similarity (SSIM) index are not sufficiently adequate for that purpose, for several reasons:

- Despite constituting valuable means to infer operating points of the transmission system, in terms of the average quality that can be achieved given a series of

encoded video stream, QoS/QoE-control mechanism, and communication channel characteristics, these measurements are not sufficiently well suited for evaluating the unequal importance of single data elements in the bitstream. The reason is that they depend strongly on the content of the sequence, regarding both the complexity of the scene and the amount of motion in it, to which, as mentioned, we do not have access. The PSNR/SSIM values that respectively correspond to losing two units with the same or very similar characteristics from the point of view of the result of coding (e.g. two video data packets transporting information belonging to approximately the same pixels of frames situated in the same position of different GOPs of the same length encoded using the same prediction structure) might differ considerably, but this cannot be known in advance without more deeply inspecting the video data. Therefore, considering the scenario, they are not trustworthy enough to establish a direct, sufficiently-reliable link between general content-independent key features of the data units and the potential distortion that might be introduced if they are lost and so to properly estimate the relative importance of these units.

- They rely greatly on the error concealment technique that is used to replace the missing data at the decoder. Both when the information that is employed comes from the same frame (e.g. intra-frame interpolation and extrapolation techniques) or from previously decoded ones (e.g. frame copy or inter-frame interpolation techniques), the result depends on the actual content in the pictures and therefore this is reflected in the measurements.
- The values of the first pixels that come out of the decoder after a packet loss are often quite unpredictable, basically random. The reason for that outcome is that decoders do not detect information gaps in the bitstream unless, at some point, due to the loss of synchronization, they encounter incongruous data, that is, information that does not comply with the standard recommendations in use. Until this happens, they keep doing their job, even if the result is a series of incoherent values. This effect is particularly damaging when combined with the error concealment technique.

Therefore, we propose to characterize the distortion as the number of pixels whose value differs from that of the original pixels, but not the difference between those values, is better suited to create a more-proper link between key characteristics of the video data elements and their relative relevance in the considered scenario. Obviously, there is always an influence from the content of the sequence, or, more precisely, on the amount of motion in the scene, but this is far lower than in the cases presented above. This measurement based on considering affected picture areas does not depend on the error concealment technique that is employed or the potential misbehavior of the decoder when information gaps cause losses of synchronization.

The way in which the distortion is measured has consequences regarding how the effect of different simultaneous packet losses correlates and the way in which the contribution of each packet loss to the overall introduced distortion is estimated. This is finally reflected in the resulting expected distortion values associated to each data unit, which

utterly determine their actual importance. In particular, the sign of the correlation factor highly depends on the way distortion is characterized. For instance, if content-aware measurements are used, the factor can either be positive, negative or zero, as mentioned in the previous subsection. However, with the proposed measurement, this factor can only be negative or zero, as will be illustrated throughout this chapter. This situation greatly influences the relative relevance of the packets in the window.

5.3 Proposed distortion model

5.3.1 Motivation

In the considered scenario, distortion models are desired to fulfill several conditions, regarding accuracy, processing time and whether they require access to the actual content of the sequence. Most of the models present in the literature lack to some extent any of these features, either because: (i) they do not properly assess the relevance of different packets (for instance because of not suitably considering the correlation effect of simultaneous losses), (ii) they need to access the source content or are content dependent, which assuming a structure of rather independent modules in the transmitter, is not always possible or desirable, or (iii) the time that it takes to process them exceeds real-time limitations (a circumstance that is many times related to the previous factor). Thus, we propose a distortion model that is accurate enough to make the most of QoS mechanisms, particularly of the unequal protection approach proposed in this thesis, that uses as input information in the stream that can be very readily fetched, and that can be processed in real time.

5.3.2 Expected distortion introduced in a video sequence

The overall distortion that is introduced in a video sequence during its transmission is estimated in our model considering the contribution of all the packets in the window of observation simultaneously, as a consequence of the fact that the information carried by packets is interrelated as a result of intra- and inter-frame prediction, as pointed out before. We here assume that one source packet transports information associated with only one frame/slice¹.

To compute the overall expected distortion, we need to consider the distortion that results from every possible combination of received and lost packets within a window of observance and the likelihood of each of these combinations happening. In this sense, let $\mathbf{t} = \{t_1, t_2, \dots, t_{N_{P,W}}\}$ be a vector of stochastic variables associated with the transmission of the $N_{P,W}$ packets in the window, where t_p is a binary stochastic variable that takes

¹If M2TS is used for the carriage of encoded video, typically up to 7 consecutive M2TS packets are grouped into each RTP packet, regardless of the type of information they transport. In order to guarantee that no RTP unit carries information associated with two or more frames/slices, an explicit aware distribution is required. This can be done through a rewrapping tool like the one presented in [121]

If native RTP is employed, this can be guaranteed by not allowing the utilization of RTP aggregation packets [34].

one of the two possible values that can result from the event of transmitting packet p : to successfully receive it, which is indicated with a 0, or to lose it during the transmission process (this includes not receiving the packet on time), indicated with a 1. Therefore, there exist $2^{N_P, w}$ combinations, regarding the two possible realizations of each variable. If cardinal number c is used to enumerate all the possible combinations of realizations, $\mathbf{t}^c$ is the c^{th} combination. Then, the overall expected distortion of the group of packets is expressed as:

$$\overline{D}_T = \sum_{c=1}^{2^{N_P, w}} D(\mathbf{t}^c) \cdot P(\mathbf{t}^c) \quad (5.1)$$

where $D(\mathbf{t}^c)$ is the distortion in the video that results from combination $\mathbf{t}^c$, and $P(\mathbf{t}^c)$ is the likelihood of occurrence of this combination.

As can be seen, the number of combinations increases exponentially with N_P, w , quickly becoming computationally too costly to fulfill even relatively lax time restrictions. Therefore, it becomes necessary to tackle this problem differently.

As mentioned, in the proposed model, the distortion is characterized as the amount of affected area in the different pictures that make up the video sequence, that is, the number of pixels that differ between the sequence encoded and subsequently decoded and the one that results from decoding the potentially incomplete information that arrives at the decoder, out of the total. This measurement does not consider source distortion, i.e., the quality drop introduced in the encoding process. The overall distortion introduced by each combination of received and lost packets in the window can be expressed as the sum of the distortion introduced by every frame in this window when this combination takes place, as expressed next:

$$\overline{D}_T = \sum_{c=1}^{2^{N_P, w}} \left(\sum_{f=1}^{N_F, w} D_f(\mathbf{t}^c) \right) \cdot P(\mathbf{t}^c) = \sum_{f=1}^{N_F, w} \sum_{c=1}^{2^{N_P, w}} D_f(\mathbf{t}^c) \cdot P(\mathbf{t}^c) = \sum_{f=1}^{N_F, w} \overline{D}_f \quad (5.2)$$

where $D_f(\mathbf{t}^c)$ is the quality degradation caused by frame f when combination $\mathbf{t}^c$ happens. $\overline{D}_f$ is then the expected distortion introduced by frame f , considering all the possible combinations of received and lost packets in the window and its likelihood of occurrence.

5.3.3 Expected distortion introduced by a frame ($\overline{D}_f$)

The expected distortion introduced by a frame f , $\overline{D}_f$, depends on both what can happen to this very frame and on what can occur in its direct and indirect reference frames. The direct reference frames of frame f are the set of frames in the sequence whose information is used to encode that of frame f . The indirect references of frame f are the set of frames in the sequence whose information is used to encode that of the direct reference frames of frame f , plus those whose information is used to encode that of the latter ones, and so on. Basically, all the frames that have a relation of dependence with frame f but do not belong to the set of direct reference frames, are indirect reference frames. So, $\overline{D}_f$

relies on all the possible combinations of received and lost packets belonging to each of the frames that affects frame f , including itself.

So, let $R_f = \{f_{\text{ref},1}, f_{\text{ref},2}, \dots, f_{\text{ref},N_{F,R_f}}\}$ be the set of reference frames of frame f , where N_{F,R_f} is its cardinality. Moreover, let $N_{P,f}$ be the number of packets carrying information of frame f , and $N_{P,f_{\text{ref},1}}, N_{P,f_{\text{ref},2}}, \dots, N_{P,f_{\text{ref},N_{F,R_f}}}$ be the number of packets in each of its references, respectively. Let N_{P,R_f} then be the total number of packets in the stream that belong to any of the frames in R_f , that is:

$$N_{P,R_f} = \sum_{k=1}^{N_{F,R_f}} N_{P,f_{\text{ref},k}}$$

Then, in a similar way to the one used in the previous subsection, let $\mathbf{t}_f = \{t_{f,1}, t_{f,2}, \dots, t_{f,N_{P,R_f}}, \dots, t_{f,N_{P,R_f}+N_{P,f}}\}$ be a vector of stochastic variables associated with the transmission of the $N_{P,R_f} + N_{P,f}$ packets in frame f and its reference frames, where $t_{f,p}$ expresses the two possible realizations regarding the result of transmitting the p^{th} packet in this sequence. Again, if cardinal number $c_{R_f \cup f}$ is used to enumerate all the possible combinations of the $N_{P,R_f} + N_{P,f}$ packets in frame f and its reference frames, $\mathbf{t}_{R_f \cup f}^{c_{R_f \cup f}}$ is employed to express the $c_{R_f \cup f}^{\text{th}}$ combination. $\overline{D}_f$ can then be expressed in the following fashion:

$$\overline{D}_f = \sum_{c_{R_f \cup f}=1}^{2^{N_{P,R_f}+N_{P,f}}} \overline{D}_f(\mathbf{t}_{R_f \cup f}^{c_{R_f \cup f}}) \quad (5.3)$$

where $\overline{D}_f(\mathbf{t}_{R_f \cup f}^{c_{R_f \cup f}})$ is the distortion introduced by frame f when combination $\mathbf{t}_{R_f \cup f}^{c_{R_f \cup f}}$ occurs, considering the likelihood of occurrence of this combination.

Nevertheless, the number of combinations to be considered can be highly reduced taking into account the loss of synchronization that takes place when a packet is lost: when losing a portion of a bitstream, the decoder discards all the succeeding bits until the following synchronization marker arrives, which happens by the arrival of the next NALU header, i.e., by the beginning of the subsequent frame/slice.

The consequence of this factor is that once a packet is lost, the realizations associated with the subsequent packets in the stream carrying information of the same frame need not be considered, as, regardless of the result of the transmission of these packets, the distorting power that is introduced is the same. This masking effect, then, leads to the negative correlation factor mentioned at the beginning of the section, which, as shown, can be used to our advantage in the following fashion: all the combinations within each frame sharing the location of the first lost packet can be handled together.

Consequently, in a given frame f' with $N_{P,f'}$ packets, there only exist $N_{P,f'} + 1$ different groups of accumulated combinations: one per packet belonging to the frame plus one group for the combination where no packets are lost. Now, if cardinal number $g_{f'}$ is used to enumerate all the possible groups of combinations within frame f' , $\mathbf{t}_{f'}^{p_{f',g_{f'}}}$ is used to express the group of combinations where packet $p_{f',g_{f'}}$ is the first lost packet (the first $g_{f'} - 1$ packets are successfully received). For convenience, $g_{f'}$ equals $N_{P,f'} + 1$

Figure 5.1: Propagation of the error caused by single packet losses throughout the sequence for different sequences

for the combination where no packets are lost. However, frame f introduces no distortion in this case, and so is not considered in the general expression.

So, (5.3) can be rewritten as follows:

$$\begin{aligned}
 \overline{D}_f &= \sum_{g_{f_{\text{ref}},1}=1}^{N_{P,f_{\text{ref}},1}} \sum_{g_{f_{\text{ref}},2}=1}^{N_{P,f_{\text{ref}},2}} \dots \sum_{g_{f_{\text{ref}},N_F,R_f}=1}^{N_{P,f_{\text{ref}},N_F,R_f}} \sum_{g_f=1}^{N_{P,f}} \\
 &\quad \overline{D}_f(t_{f_{\text{ref}},1}^{p_{f_{\text{ref}},1},g_{f_{\text{ref}},1}}, t_{f_{\text{ref}},2}^{p_{f_{\text{ref}},2},g_{f_{\text{ref}},2}}, \dots, t_{f_{\text{ref}},N_F,R_f}^{p_{f_{\text{ref}},N_F,R_f},g_{f_{\text{ref}},N_F,R_f}}, t_f^{p_f,g_f}) \quad (5.4)
 \end{aligned}$$

where $\overline{D}_f(t_{f_{\text{ref}},1}^{p_{f_{\text{ref}},1},g_{f_{\text{ref}},1}}, t_{f_{\text{ref}},2}^{p_{f_{\text{ref}},2},g_{f_{\text{ref}},2}}, \dots, t_{f_{\text{ref}},N_F,R_f}^{p_{f_{\text{ref}},N_F,R_f},g_{f_{\text{ref}},N_F,R_f}}, t_f^{p_f,g_f})$ is the distortion introduced by frame f when packets $p_{f_{\text{ref}},1}, g_{f_{\text{ref}},1}, p_{f_{\text{ref}},2}, g_{f_{\text{ref}},2}, \dots, p_{f_{\text{ref}},N_F,R_f}, g_{f_{\text{ref}},N_F,R_f}$ are the first lost ones in each reference frame, respectively, and p_{f,g_f} is the first lost packet in frame f , considering the likelihood of occurrence of these groups of combinations.

In order to estimate the expected distortion introduced by a given combination of received and lost packets of frames f and its direct and indirect references, we will first consider that this frame has no reference frames at all, hence assuming that the introduced distortion exclusively depends on the errors originating in it. Later on, we will incorporate the impact of the reference frames on that frame.

Figure 5.2: Graphic representation of the distortion introduced as a result of the propagation of the error caused by the loss of a single packet. Arrows point at the direct reference of each frame. The blue cross indicates the initial pixel in the frame the lost packet carries information about.

5.3.3.1 Distortion introduced without considering the references ($D_{\text{indep},f}$)

The model presented in this subsection directly derives from a study that analyzes, for a number of packetized encoded sequences, how the distortion caused by the loss of a packet in a frame propagates to dependent frames, that is, the percentage of affected pixels in those frames. Figure 5.1 depicts the results for four well-known sequences: “coastguard”, “foreman”, “football” and “container”. These sequences were encoded using a prediction structure of 100 frames following the pattern IPP... and setting a target bitrate of 2Mbps. The included figures present the proportion of affected pixels in all the frames that make up the GOP when only one of the packets in the first frame is lost. Each line in each graph reflects the result of losing only one of the packets transporting information of the first frame.

As can be seen, the percentage of pixels whose value differs from the original when an error in a reference frame occurs remains fairly constant with the distance to the frame where the error takes place, or goes up or down very slowly over a great number of frames. These tendencies remain steady throughout the sequence unless a significant disturbance in the scene (e.g. considerable increment of motion, scene change...) gets in the way. Hence, we will assume in our model that all interdependent pictures (i.e. the frames in a GOP) belong to the same scene, where errors can be assumed to propagate affecting a rather constant number of pixels in the frames. Under this premise, with high probability, no big irregularities will break the steady tendency. Of course, the probability is even higher if the scene change detection mode was enabled when encoding the sequence.

So, in light of the presented data, the percentage of affected pixels in dependent frames is modeled assuming that it remains constant with the distance. Thus, the distortion introduced by an independent frame f when packet p_{f,g_f} is the first lost one is modeled as follows:

$$\overline{D}_{\text{indep},f}(\mathbf{t}_f^{p_{f,g_f}}) = \xi_{\text{indep},p_{f,g_f}} \cdot P(\mathbf{t}_f^{p_{f,g_f}}) \cdot n_{F,f} \quad (5.5)$$

where $\xi_{\text{indep},p_{f,g_f}}$ is the percentage of affected pixels in frame f when packet p_{f,g_f} is the first lost one, $P(\mathbf{t}_f^{p_{f,g_f}})$ is the likelihood of this occurrence, and $n_{F,f}$ is the number of frames in the window of observance that directly or indirectly depend on frame f for

decoding.

$\xi_{\text{indep}, p_{f, g_f}}$ indicates how potentially distorting packet p_{f, g_f} is. Figure 5.2 presents a visual example of the distortion introduced as a result of the propagation of the error caused by the loss of a packet. In that figure, it can be seen that the isolated loss of packet p_{f_1, p_1} in frame f_1 results in $\xi_{\text{indep}, p_{f_1, p_1}}$ percent of the pixels in this very frame and the ones depending on it taking wrong values.

The index of this first lost packet, jointly with the size of the packets belonging to frame f , determines the percentage of affected picture. It is assumed that the size of a packet is proportional to the number of pixels whose information is transported by it. Let $s_{f,1}, s_{f,2}, \dots, s_{f, N_{P,f}}$ be the size in number of octets of the $N_{P,f}$ packets carrying information of frame f , as they are located along the packet stream. Therefore, considering the effect derived from the loss of synchronization, $\xi_{\text{indep}, p_{f_1, p_1}}$ can be estimated in the following way:

$$\xi_{\text{indep}, p_{f, g_f}} = \frac{\sum_{i=g_f}^{N_{P,f}} s_{f,i}}{\sum_{i=1}^{N_{P,f}} s_{f,i}} \quad (5.6)$$

where the case where none of the $N_{P,f}$ packets is lost is also contemplated ($g_f = N_{P,f} + 1$). In this case, the area of the frame affected on account of intra-frame propagation is non-existent.

The likelihood of packet p_{f, g_f} being the first lost one in the frame is addressed later on.

5.3.3.2 Impact of a simultaneous errors in a reference frame

Before actually formalizing the influence of previous packets in the distortion introduced by a group of combinations, we discuss in this subsection the effect of simultaneously losing two or more packets belonging to different interrelated frames, regarding the correlation between them and the distribution of the resulting damage in the sequence among the lost packets.

We first consider the loss of two packets. To that end, let packet p_{f_1, p_1} belonging to frame f_1 be the first of the two lost packets in the transmitted packet stream and packet p_{f_2, p_2} in frame f_2 be the second one. The isolated loss of the first of them would affect $\xi_{\text{indep}, p_{f_1, p_1}}$ percent of the pixels of frame f_1 and the frames depending on it, whereas the isolated loss of the second one would change the value of $\xi_{\text{indep}, p_{f_2, p_2}}$ percent of the pixels of frame f_2 and of the frames relying on it, according to the previous subsection.

Two different cases can then be identified. In the first one, $\xi_{\text{indep}, p_{f_1, p_1}} < \xi_{\text{indep}, p_{f_2, p_2}}$. In the second one, $\xi_{\text{indep}, p_{f_1, p_1}} \geq \xi_{\text{indep}, p_{f_2, p_2}}$. The two examples are respectively illustrated in Figure 5.3. In these examples, the effect of packet p_{f_1, p_1} is presented in orange and the effect of packet p_{f_2, p_2} in blue. The striped areas reflect, for each packet, the pixels in each frame that would be affected by its loss, if it were an isolated loss. As can be clearly seen, there exist an overlap of the two areas, which means that there exists a correlation between the effect of both losses. The first conclusion of this fact is the assertion that non-interdependence-aware models cannot accurately capture the effect that

Figure 5.3: Graphic representation of the distortion introduced as a result of the propagation of the error caused by the loss of two packets. Arrows point at the direct reference of each frame. The orange cross indicates the initial pixel the first lost packet carries information about and the blue one indicates the initial pixel the first lost packet carries information about.

results from losing more than one interrelated data packets, as they do not consider this correlation factor whatsoever. This correlation factor is, for the distortion measurement used in this approach, negative. This contrasts with what happens when using PSNR or other measurements, as stated by [120] and other works.

If an area of a frame has been affected by the loss of two or more packets in different frames (or, more generally, combinations of received and lost packets where, in each frame, these packets are the first lost ones), this damage needs to be attributed to one of the packets, so as to determine the actual contribution of each packet to the overall distortion in the sequence. According to the sequence encoding order, which determines the dependence relations in the stream, the damage is attributed to the packet that is located first in the packet stream. In addition, this is consistent with the assignment of responsibility within frames when a loss of synchronization takes place.

Hence, in the example presented in Figure 5.3, the affected area that is assigned to each lost packet is surrounded by a rectangle of the corresponding color. In the first of them (Fig. 5.3a), the contribution of the loss of the packet colored in blue to the overall distortion introduced in the sequence, initially estimated assuming that the frame it belongs to is independent, is reduced by the simultaneous loss of a packet in a reference frame (colored in orange). In the second one (Fig. 5.3b), the contribution of the loss of the packet colored in blue to the overall distortion is reduced from the initially estimated one to zero if the packet colored in orange is also lost. This makes perfect sense, since the overall distortion that is introduced as a results of all the losses remains the same regardless of this packet being correctly received or lost.

Finally, we present another example where three packets are lost. This example is

Figure 5.4: Graphic representation of the distortion introduced as a result of the propagation of the error caused by the loss of three packets. Arrows point at the direct reference of each frame. The black cross indicates the initial pixel the first lost packet carries information about, the orange cross indicates the initial pixel the second lost packet carries information about and the blue one indicates the initial pixel the third lost packet carries information about.

shown in Figure 5.4. Let packet p_{f_1,p_1} belonging to frame f_1 be the first of the three lost packets in the transmitted packet stream, packet p_{f_2,p_2} in frame f_2 be the second one, and packet p_{f_3,p_3} in frame f_3 be the third. The isolated loss of the first of them would affect $\xi_{\text{indep},p_{f_1,p_1}}$ percent of the pixels of frame f_1 and the frames depending on it, the isolated loss of the second one would change the value of $\xi_{\text{indep},p_{f_2,p_2}}$ percent of the pixels of frame f_2 and of the frames relying on it, and the isolated loss of the third one would change the value of $\xi_{\text{indep},p_{f_3,p_3}}$ percent of the pixels of frame f_3 and of the frames depending on it. As can be seen in the illustration, $\xi_{\text{indep},p_{f_1,p_1}} < \xi_{\text{indep},p_{f_2,p_2}} < \xi_{\text{indep},p_{f_3,p_3}}$ and $\xi_{\text{indep},p_{f_2,p_2}} > \xi_{\text{indep},p_{f_3,p_3}}$. According to the above description on how damage is distributed among the lost packets, we have introduced rectangles surrounding the areas in the frames whose damaged is attributed to the packet with the corresponding color (black, orange or blue). In this example, in the same way as before, the contribution of the third packet is zero.

5.3.3.3 Formalization of the impact of the reference frames

As already mentioned, the loss of synchronization caused by not receiving a packet provokes a masking effect within frames that we have used to notably decrease the number of combinations to consider within each frame. In the same way, the masking effect identified in the previous analysis can also be used to our advantage as follows: all the combinations of received and lost packets belonging to reference frames that lead to causing the same impact can be handled together. To properly formalize this, first let $p_{\text{ref},1}, p_{\text{ref},2}, \dots, p_{\text{ref},N_P,R_f}$ be the packets transporting information of the reference frames of frames f , sorted regarding their potential distortion, that is, $\xi_{\text{indep},p_{\text{ref},1}} \geq \xi_{\text{indep},p_{\text{ref},2}} \geq \dots \geq \xi_{\text{indep},p_{\text{ref},N_P,R_f}}$. So, if packets $p_{\text{ref},1}$ to $p_{\text{ref},i-1}$ are successfully received and packet $p_{\text{ref},i}$ is lost, whatever happens to packets $p_{\text{ref},i+1}$ to p_{ref,N_P,R_f} , the effect on frame f is the same. Thus, in the same fashion as before, if cardinal number g_{R_f} is used to enumerate all the possible groups of combinations of the packets in the reference frames of frame f , $t_{R_f}^{p_{\text{ref}},g_{R_f}}$ is used to express the group of combinations

where packet $p_{\text{ref},g_{R_f} \cup f}$ is the first lost packet (the first $g_{R_f} \cup f - 1$ packets are successfully received).

Therefore, expression (5.4) can be rewritten as follows:

$$\bar{D}_f = \sum_{g_{R_f} \cup f=1}^{N_{P,R_f}} \sum_{g_f=1}^{N_{P,f}} \bar{D}_f(\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}}, \mathbf{t}_f^{p_{f,g_f}}) \quad (5.7)$$

where $\bar{D}_f(\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}}, \mathbf{t}_f^{p_{f,g_f}})$ is the distortion introduced by frame f when combinations $\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}}$ and $\mathbf{t}_f^{p_{f,g_f}}$ occur, considering the likelihood of occurrence of these combinations.

Following the previous analysis, $\bar{D}_f(\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}}, \mathbf{t}_f^{p_{f,g_f}})$ can be computed thorough subtracting the effect of every group of combinations in the reference frames (their masking effect) from the percentage of pixels in frame f that would be affected if there were no influence from other frames. That is, by subtracting portions that should actually be attributed to other data units in the stream from the potential distortion power initially attributed to the different packets in frame f . This is stated as follows:

$$\bar{D}_f(\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}}, \mathbf{t}_f^{p_{f,g_f}}) = \bar{\xi}_{p_{f,g_f}} \left(\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}}, \mathbf{t}_f^{p_{f,g_f}} \right) \cdot n_{F,f} \quad (5.8)$$

where $\bar{\xi}_{p_{f,g_f}}$ represents the actual distorting power of packet p_{f,g_f} . $n_{F,f}$ is, as before, the number of frames in the window that directly or indirectly depend on frame f for decoding. The value of $\bar{\xi}_{p_{f,g_f}}$ is calculated as:

$$\bar{\xi}_{p_{f,g_f}} \left(\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}}, \mathbf{t}_f^{p_{f,g_f}} \right) = \left(\xi_{\text{indep},p_{f,g_f}} - \sum_{g_{R_f}=1}^{N_{P,R_f}} \bar{\eta}_{p_{\text{ref},g_{R_f}},p_{f,g_f}} \left(\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}} \right) \right) \cdot P \left(\mathbf{t}_f^{p_{f,g_f}} \right) \quad (5.9)$$

where $\bar{\eta}_{p_{\text{ref},g_{R_f}},p_{f,g_f}}$ is the loss of distorting power of packet p_{f,g_f} if the $g_{R_f} - 1$ most potentially distorting packets carrying information of the references of frame f are correctly received and packet $p_{\text{ref},g_{R_f}}$ is lost, considering the likelihood of this event. The value of $\bar{\eta}_{p_{\text{ref},g_{R_f}},p_{f,g_f}}$ is computed as:

$$\bar{\eta}_{p_{\text{ref},g_{R_f}},p_{f,g_f}} \left(\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}} \right) = \min \left(\xi_{\text{indep},p_{f,g_f}}, \xi_{\text{indep},p_{\text{ref},g_{R_f}}} \right) \cdot P \left(\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}} \right) \quad (5.10)$$

where, as aforementioned, $\xi_{\text{indep},p_{\text{ref},g_{R_f}}}$ represents the potential distortion of $p_{\text{ref},g_{R_f}}$, and $P(\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}})$ is the likelihood of losing packet $p_{\text{ref},g_{R_f}}$ and correctly receiving the first $g_{R_f} - 1$ more potentially distorting packets of the N_{P,R_f} .

The previous expression states that the effect of packet $p_{\text{ref},g_{R_f}}$ on packet p_{f,g_f} equals the percentage of pixels affected by the loss of packet p_{f,g_f} that should actually be

attributed to packet $p_{\text{ref},g_{R_f}}$ if the latter is also lost, but none of the $g_{R_f} - 1$ more potentially distorting packets is. If the potential distorting power of packet $p_{\text{ref},g_{R_f}}$ is lower than that of packet p_{f,g_f} , the loss of packet $p_{\text{ref},g_{R_f}}$ will mean a reduction of the actual distorting power of packet p_{f,g_f} (initially estimated as equal to $\xi_{\text{indep},p_{f,g_f}}$) by $\xi_{\text{indep},p_{\text{ref},g_{R_f}}}$. If the potential distorting power of packet $p_{\text{ref},g_{R_f}}$ is equal or greater than that of packet p_{f,g_f} , the actual distorting power of packet p_{f,g_f} will be reduced to zero, if the former is lost.

Finally, both the likelihood of occurrence of a group of combinations in a frame, $P(\mathbf{t}_f^{p_{f,g_f}})$, and the likelihood of occurrence of a group of combinations in the reference frames, $P(\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}})$, are addressed in the following subsections.

5.3.3.4 Likelihood of occurrence of a group of combinations in a frame ($P(\mathbf{t}_f^{p_{f,g_f}})$)

The likelihood of occurrence of the group of combinations of received and lost packets belonging to frame f where packet g_f is the first lost one, $P(\mathbf{t}_f^{p_{f,g_f}})$, is expressed as:

$$P(\mathbf{t}_f^{p_{f,g_f}}) = P(0^{g_f-1}1) \quad (5.11)$$

where 0 indicates that the data packet is correctly received and 1 that it is lost. So, $P(0^{g_f-1}1)$ represents the likelihood of successfully receiving the first $g_f - 1$ packets of the frame, and losing the following one.

In addition, we define $P(1 - \mathbf{t}_f^{p_{f,g_f}})$ as the likelihood of none of the combinations including the loss of packets $p_{f,1}$ to p_{f,g_f} happening, that is:

$$P(1 - \mathbf{t}_f^{p_{f,g_f}}) = P(0^{g_f}) \quad (5.12)$$

Therefore, $P(1 - \mathbf{t}_f^{N_{P,f}})$ is the likelihood of successfully receiving all the packets belonging to frame f .

These expressions have to be particularized regarding two main elements: (i) the channel model that is employed to model that behavior, and (ii) the QoS control mechanism employed to increase the quality of the sequence presented to the user, e.g., the FEC scheme used to protect the video stream.

5.3.3.5 Likelihood of occurrence of a group of combinations in the reference frames ($P(\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}})$)

As already mentioned, when applying the proposed distortion model, the set of packets in the stream that belong to any of the reference frames of frame f are sorted regarding the potential distortion they may introduce in the frame they belong to, regardless what frame this is. To compute the likelihood of receiving the $g_{R_f} - 1$ packets with more distorting power of them and losing the following one, it is necessary to consider the frame they belong to. The likelihood of packet $p_{\text{ref},g_{R_f}}$ belonging to frame $f_{\text{ref},i}$ being the

Figure 5.5: Example of the application of the proposed distortion model to estimate the distortion that the packet colored in blue is expected to introduce in the sequence. Arrows point at the direct reference of each frame. The numbered packets are the ones belonging to reference frames of the frame the packet under study belongs to. The numbers indicate how they are sorted regarding how potentially distorting they are.

one impacting the packet under study is computed simultaneously considering that this is the first lost packet in frame $f_{\text{ref},i}$ and that there exist no combinations of received and lost packets in other reference frames leading to affecting a greater number of pixels in the frame under study.

Considering, as before, that frames $f_{\text{ref},1}, f_{\text{ref},2}, \dots, f_{\text{ref},N_{\text{F},R_f}}$ are the references of frame f , let packet $p_{f_{\text{ref},j}}/j \in \{1, \dots, N_{\text{F},R_f}\} \cup \{i \neq j\}$ be the less potentially distorting one in reference frame $f_{\text{ref},j}$ that is still more or equally potentially distorting than packet $p_{\text{ref},g_{R_f}}$, i.e., $\xi_{\text{indep},p_{f_{\text{ref},j}}} \geq \xi_{\text{indep},p_{\text{ref},g_{R_f}}}$. That is, if there exists at least one packet in reference frame $f_{\text{ref},j}$ that has more distorting power than packet $p_{\text{ref},g_{R_f}}$, then packet $p_{f_{\text{ref},j}}$ is one of the $g_{R_f} - 1$ more potentially distorting packets in the reference frames. Then, the likelihood of packet $p_{\text{ref},g_{R_f}}$ belonging to frame $f_{\text{ref},i}$ being the one impacting the packet under study is formalized as follows:

$$P(\mathbf{t}_{R_f}^{p_{\text{ref},g_{R_f}}}) = P(\mathbf{t}_{f_{\text{ref},i}}^{p_{\text{ref},g_{R_f}}}) \cdot \prod_{\substack{j=1 \\ j \neq i}}^{N_{\text{F},R_f}} P(1 - \mathbf{t}_{f_{\text{ref},j}}^{p_{f_{\text{ref},j}}}) \quad (5.13)$$

5.3.3.6 Example of use of the proposed distortion model

In this example, the following setup is employed: the window of observance consists of one GOP with pattern IPP... that is made up of five frames (f_1, f_2, f_3, f_4 and f_5), and the number of packets transporting information related to each frame is $N_{\text{P},1} = 4$, $N_{\text{P},2} = 3$, $N_{\text{P},3} = 2$, $N_{\text{P},4} = 2$, and $N_{\text{P},5} = 1$, respectively. This is reflected in Figure 5.5. To properly illustrate how the proposed distortion model operates, the real contribution to the expected overall distortion of the second packet in the third frame is computed, that is, packet $p_{2,3}$. This packet transports information related to a number of pixels in frame f_3 . This area is colored in blue in the figure. Moreover, the loss of this packets leads to changing the original value of a number of pixels in frames f_4 and f_5 . The area that would be affected by the loss of packet $p_{2,3}$ if frame f_3 had no references and the first packet in this very frame, packet $p_{1,3}$, were received correctly, consists of the already

mentioned pixels in frame f_3 and the pixels in frames f_4 and f_5 that are colored with blue stripes in the figure. Therefore, the different parameters in the distortion model related to the single loss of this packet take the following values: $\xi_{\text{indep},f_3,p_{2,3}} = 0.5$, $n_{F,f_3} = 3$, and $P(\mathbf{t}_{f_3}^{p_{2,3}}) = P(01)$. Assuming that the behavior of the channel is modeled through the broadly used simplified Gilbert model, $P(01) = p_G \cdot p_{GB}$. Thus, if frame f_3 had no references, the distortion associated with packet $p_{2,3}$ would be:

$$\overline{D}_{\text{indep},f_3,p_{2,3}} = D_{\text{indep},f_3}(\mathbf{t}_{f_3}^{p_{2,3}}) \cdot P(\mathbf{t}_{f_3}^{p_{2,3}}) = \xi_{\text{indep},p_{2,3}} \cdot n_{F,f_3} \cdot P(\mathbf{t}_{f_3}^{p_{2,3}}) = 0.5 \cdot 3 \cdot P(01) = 0.5 \cdot 3 \cdot p_G \cdot p_{GB}$$

We will now consider the seven packets in the stream that carry information belonging to the reference frames of frame f_3 , i.e., frames f_1 and f_2 . In Figure , they are sorted regarding how potentially distorting they are, that is, considering the the value that parameter $\xi_{\text{indep},f,p}$ takes for each of them. The expected distortion that packet $p_{2,3}$ is expected to introduce equals:

$$\overline{D}_{f_3,p_{2,3}} = D_{f_3}(\mathbf{t}_{f_3}^{p_{2,3}}) \cdot P(\mathbf{t}_{f_3}^{p_{2,3}}) = \left(\xi_{\text{indep},p_{2,3}} - \sum_{i=1}^7 \bar{\eta}_{p_{\text{ref},i},p_{f,gf}} \right) \cdot n_{F,f_3} \cdot P(\mathbf{t}_{f_3}^{p_{2,3}})$$

where the expected loss of distorting power on account of each of them is computed as presented next:

$$\bar{\eta}_{p_{\text{ref},1},p_{2,3}} = \min(\xi_{\text{indep},p_{2,3}}, \xi_{\text{indep},p_{1,1}}) \cdot P(\mathbf{t}_{f_{\text{ref}}}^{p_{1,1}}) = \min(0.5, 1.0) \cdot P(\mathbf{t}_{f_1}^{p_{1,1}}) = 0.5 \cdot P(1) = 0.5 \cdot p_B$$

$$\bar{\eta}_{p_{\text{ref},2},p_{2,3}} = \min(\xi_{\text{indep},p_{2,3}}, \xi_{\text{indep},p_{2,1}}) \cdot P(\mathbf{t}_{f_{\text{ref}}}^{p_{2,1}}) = \min(0.5, 1.0) \cdot P(1 - \mathbf{t}_{f_1}^{p_{1,1}}) \cdot P(\mathbf{t}_{f_1}^{p_{2,1}}) = 0.5 \cdot P(0) \cdot P(1) = 0.5 \cdot (p_G) \cdot (p_B)$$

$$\begin{aligned} \bar{\eta}_{p_{\text{ref},3},p_{2,3}} &= \min(\xi_{\text{indep},p_{2,3}}, \xi_{\text{indep},f_1,p_{1,2}}) \cdot P(\mathbf{t}_{f_{\text{ref}}}^{p_{1,2}}) = \\ &\min(0.5, 0.75) \cdot P(\mathbf{t}_{f_1}^{p_{1,2}}) \cdot P(1 - \mathbf{t}_{f_1}^{p_{2,1}}) = 0.5 \cdot P(01) \cdot P(0) = 0.5 \cdot (p_G \cdot p_{GB}) \cdot (p_G) \end{aligned}$$

$$\bar{\eta}_{p_{\text{ref},4},p_{2,3}} = \min(\xi_{\text{indep},p_{2,3}}, \xi_{\text{indep},p_{2,2}}) \cdot P(\mathbf{t}_{f_{\text{ref}}}^{p_{2,2}}) = \min(0.5, \frac{2}{3}) \cdot P(1 - \mathbf{t}_{f_1}^{p_{1,2}}) \cdot P(\mathbf{t}_{f_1}^{p_{2,2}}) = 0.5 \cdot P(00) \cdot P(01) = 0.5 \cdot (p_G \cdot p_{GG}) \cdot (p_G \cdot p_{GB})$$

$$\bar{\eta}_{p_{\text{ref},5},p_{2,3}} = \min(\xi_{\text{indep},p_{2,3}}, \xi_{\text{indep},p_{1,3}}) \cdot P(\mathbf{t}_{f_{\text{ref}}}^{p_{1,3}}) = \min(0.5, 0.5) \cdot P(\mathbf{t}_{f_1}^{p_{1,3}}) \cdot P(1 - \mathbf{t}_{f_1}^{p_{2,2}}) = 0.5 \cdot P(001) \cdot P(00) = 0.5 \cdot (p_G \cdot p_{GG} \cdot p_{GB}) \cdot (p_G \cdot p_{GG})$$

$$\bar{\eta}_{p_{\text{ref},6},p_{2,3}} = \min(\xi_{\text{indep},p_{2,3}}, \xi_{\text{indep},p_{2,3}}) \cdot P(\mathbf{t}_{f_{\text{ref}}}^{p_{2,3}}) = \min(0.5, \frac{1}{3}) \cdot P(1 - \mathbf{t}_{f_1}^{p_{1,3}}) \cdot P(\mathbf{t}_{f_1}^{p_{2,3}}) = \frac{1}{3} \cdot P(000) \cdot P(001) = \frac{1}{3} \cdot (p_G \cdot p_{GG} \cdot p_{GG}) \cdot (p_G \cdot p_{GG} \cdot p_{GB})$$

$$\bar{\eta}_{p_{\text{ref},7},p_{2,3}} = \min(\xi_{\text{indep},p_{2,3}}, \xi_{\text{indep},p_{1,4}}) \cdot P(\mathbf{t}_{f_{\text{ref}}}^{p_{1,4}}) = \min(0.5, 0.25) \cdot P(\mathbf{t}_{f_1}^{p_{1,4}}) \cdot P(1 - \mathbf{t}_{f_1}^{p_{2,3}}) = 0.25 \cdot P(0001) \cdot P(000) = 0.25 \cdot (p_G \cdot p_{GG} \cdot p_{GG} \cdot p_{GB}) \cdot (p_G \cdot p_{GG} \cdot p_{GG})$$

The actual values of the different probabilities depend on the packet loss rate (PLR) and the average burst length (ABL) of the channel.

Figure 5.6: Evolution of the average percentage of affected pixels with the recovery capacity equivalent to indicated recovery rate for different channel's PLR values. Scenario with GOP pattern IPBBBP... and channel's ABL = 3 ms.

5.4 Experiments

In this section, the proposed distortion model is first validated in terms of accuracy and time. To begin with, we analyze the accuracy of our proposal through a comparison with other distortion models. To do so, we apply an ideal protection scheme that enables the protection of each individual packet in proportion to the result of the selected model. Later on, we test through a battery of processing time measurements if our approach performs fast enough so that the subsequent modules in the transmitter have enough time to operate and the whole system can perform in real time.

Finally, we use our distortion model to estimate the relevance of the data packets that are subsequently protected with the proposed UEP extension to the Pro-MPEG COP3 codes. So, it can be tested in the context for which it was primarily intended.

5.4.1 Performance in terms of accuracy

To assess the accuracy of our approach, we have compared the result of applying the proposed approach in a number of scenarios to those obtained after applying the following distortion models:

- **Equal importance model:** in this model, all the packets in the video stream

Parameter	Considered values
GOP pattern	IPP... and IPBBBP...
PLR	5%, 1%, 0.5%, 0.1%
ABL	1 ms, 3 ms, 5 ms, 8 ms
Channel code nominal recovery rate	5% to 95% in steps of 5%

Table 5.1: Parameters and values used in the experiments

are considered equally important and so are assigned the same distortion value. Therefore:

$$\overline{D}_{p_f, g_f} = 1 \quad (5.14)$$

- **Frame-level model:** in this model, all the packets carrying information of the same picture are considered equally important and so are assigned the same value. This value basically only depends on the number of frames in the window of observance that have this one as reference, as stated next:

$$\overline{D}_{p_f, g_f} = \frac{n_{F,f}}{N_{P,f}} \quad (5.15)$$

- **Packet-level non-interdependency-aware model:** in this model, the distortion that a packet is expected to introduce only depends on its potential distorting power and the number of frames in the window of observance that directly or indirectly depend on frame f for decoding, as follows:

$$\overline{D}_{p_f, g_f} = \xi_{\text{indep}, p_f, g_f} \cdot n_{F,f} \quad (5.16)$$

We have performed the comparison in a number of scenarios. Each of these scenarios consists in the transmission through a simulated IP channel of a packetized encoded video sequence that has been protected with a malleable, ideal channel code. The presented results have been measured in terms of the average percentage of pixels per frame in the received sequence whose value differs from the original one. We have employed two video sequences encoded using GOP prediction structures of different patterns and a target bitrate of 10 Mbps. Moreover, in all the scenarios, a simplified Gilbert-Elliot model was used to model the behavior of the channel. The values of parameters PLR and ABL, which characterize the model, differ in each scenario. The ABL values are given in terms of time, which translate into different numbers of consecutive loss packets, regarding the bitrate of the video stream. Finally, the channel code that has been used to protect the transmitted video stream is characterized by a given nominal recovery rate. Any individual network packet that is lost during the transmission process is recovered with some probability. This probability depends on both its relative relevance (i.e. the distortion that it is expected to introduced with respect to that of the other packets in

Figure 5.7: Evolution of the average percentage of affected pixels with the recovery capacity equivalent to indicated recovery rate for different channel's ABL values. Scenario with GOP pattern IPBBBP... and channel's PLR = 1%.

the window) and the already mentioned nominal recovery rate in the following fashion: the ideal channel code provides a recovery capability to each packet in the window that directly depends on its relative relevance, in such a way that the average recovery rate per packet equals the nominal one. In this way, more important packets have a higher probability of being recovered (above the nominal recovery rate) than less important packets (below the nominal recovery rate). However, the overall recovery capability remains the same regardless of the distortion model used to determine the relevance of the packets in the window. Table 5.1 indicates the values of the parameters used in the set of experiments.

The presentation of the results has been structured so as to make easy the comparison both between distortion models in a given scenario and between performances of the same model in different scenarios. First, in Figure 5.6, we include the results in four scenarios where the only variable is the channel's PLR (both the GOP pattern used to encode the sequence and the ABL remain constant). Later on, in Figure 5.7, only the channel's ABL varies from scenario to scenario (the GOP pattern and the PLR are constant). Last, in Figure 5.8, we assess the influence of the GOP pattern. To that end, two different channels are utilized.

As can be seen, in all the considered scenarios, our interdependence-aware packet-level distortion model outperforms the other ones. That means that taking into account both

Figure 5.8: Evolution of the average percentage of affected pixels with the recovery capacity equivalent to indicated recovery rate for two different GOP patterns. Comparison in two scenarios: PLR = 1%, ABL = 3 ms, and PLR = 0.1%, ABL = 8 ms.

intra- and inter-frame data interdependence makes the distortion model more robust and accurate, as it allows us to better infer the real importance of each data packet in the stream.

Specifically, in the graphs presented in Figure 5.6, we can nevertheless see that the relative gap between the results of the considered distortion models decreases for lower PLR values. The reason is that the effect that the packets in references frames have on the packet under study is reduced as the PLR gets lower. In this situation, the expected distortion values that are obtained with our approach are closer to the ones obtained with other models, specially with the other packet-level model. Furthermore, two expected tendencies can be readily noticed. First, one can observe, graph to graph, that the average distortion that is obtained using any of the considered models goes down as the channel's PLR is reduced. That makes sense, since, regardless of the recovery rate associated to each packet, the number of lost packets drops, which leads to a lower percentage of affected pixels per frames on average. Additionally, in every subfigure, the obtained distortion goes down as the recovery capability of the channel code increases: the more packets can be recovered, the less distortion there will be, regardless of the distortion model that is used. However, the relative distance between the results obtained with our approach and the ones obtained after applying other models remains fairly constant with the capacity of the code.

Bitrate (Mbps)	GOP size	No. B-frames	Average time (ms)	Variance (ms)
5	25	0	1.37	4.0E-03
		3	0.73	2.1E-04
	50	0	6.66	4.9E-03
		3	3.57	1.2E-03
10	25	0	5.15	4.5E-03
		3	3.43	1.8E-03
	50	0	11.27	1.0E-02
		3	5.86	4.0E-03
15	25	0	6.92	5.1E-03
		3	4.45	1.4E-03
	50	0	24.8	3.8E-02
		3	12.86	1.7E-02

Table 5.2: Time that it takes to apply the presented approach to determine the distortion that every packet in the window is expected to introduce.

In Figure 5.7, it can be noticed that the average length of the error burst significantly impacts the resulting introduced average distortion. Indeed, the average percentage of affected pixels decreases with the ABL. The cause for this behavior is linked to the loss of synchronization that takes place within a frame once one of its packet is lost. As the value of the ABL increases, packet losses tend to cluster and the impact of several losses is in practice the same as that caused by one. In this way, if the value of the PLR remains constant, the number of actually damaging losses is reduced. Finally, regarding the comparison of performance between the different distortion models, we can see that the value of the ABL has little effect on it.

Finally, in Figure 5.8, we see the impact of the type of GOP prediction structure used for encoding the video sequence. The distortion that is introduced when no B-frames are used is greater than in the case with B-frames. The reason for that is that GOPs with pattern IPP... are more vulnerable to errors, as regardless of the frame a lost packet belongs to, the error will propagate until the end of the GOP. In the case of GOPs with B-frames, a portion of the lost packets will belong to those non-reference frames. In these situations, the error will be contained and will not propagate to other frames in the GOP. In this final comparison, the proposed scheme gets slightly better comparative results (that is, the relative gap between the results of our approach and those of obtained after applying the other models) in the case where GOPs with B-frames are used. This improvement is caused by better modeling the dependency relations where there exist frames that use more than one direct reference frames.

5.4.2 Performance in terms of processing time

To validate the performance of the proposed model, we use 12 12-second-long sequences encoded using different GOP prediction structures and unequal bitrates. In this way, the

Parameter	Considered values
GOP pattern	IPP... and IPBBBP...
PLR	2.5%, 1%, 0.75%, 0.5%
ABL	1 ms, 3 ms, 5 ms
FEC overhead	10%, 20%
FEC latency	1 s

Table 5.3: Parameters and values used in the experiments

number of packets in the window of observance, which is selected to equal one GOP, and the number of packets in the reference frames to be considered throughout the window varies notably from sequence to sequence. The experiment has been run 10 times per sequence. In Table 5.2, we include the average processing time and the variance per GOP, measured with a 2-core CPU clocked at 3 GHz with 12 GiB RAM.

First, we compare the measurements that result from the different scenarios. It can be observed that the time that it is required to apply the distortion model increases with the bitrate and the GOP size (as the average number of packets per window increases in the same proportion). Furthermore, the use of B-frames makes the consumed time go down. The reason is that the number of indirect reference frames decreases, and with it, the number of packets to consider for computing the loss of distorting power of a certain packet.

On the other hand, it can be observed that the average time that it takes to apply our distortion model and so determine the distortion that every packet in the window is expected to introduce varies notably depending on the characteristics of the GOP prediction structure and bitrate. However, it only consumes a fraction of the constraint times that are typically imposed in live and non-live event video transmissions for protection-related purposes, which commonly range from one to a few hundreds of milliseconds in the first case and from one to several seconds in the second one. Nevertheless, in the case that, due to the characteristics of the scenario (including type of service, GOP prediction structure or bitrate), the time that can be devoted to apply the distortion model needs to be limited, a shorter window of observance can be chosen, at the cost of reducing the accuracy of the results.

5.4.3 Performance when used jointly with UEP extension to the Pro-MPEG COP3 codes

The performance of our model is validated through a set of experiments characterized by process described next. First, a protection module receives sets of video data packets and a distortion model is applied to estimate their relevance. Those packets, jointly with the values provided by the distortion model, are input into the proposed UEP strategy, which generates the corresponding repair packets. Then, both the data and repair packets are streamed (this transmission is simulated) and a process of data recovery is carried out to try to rebuild the lost data packets. Last, the resulting stream is decoded and its

Figure 5.9: Analysis of the performance of the system in a number of scenarios when employing both proposed elements: the UEP strategy and the full-context-aware distortion model. We use as reference the performance that results from using the non-interdependent-aware distortion model with our UEP framework, as in Chapter 4. Results for GOP prediction structure pattern IPP...

distortion measured.

In this process, we use two distortion models: our approach and the packet-level non-interdependency-aware model used before, which is the same one considered in the experiments presented in Chapter 4. Moreover, two video sequences have been employed. They have been encoded using GOP prediction structures of different patterns and a target bitrate of 10 Mbps. Furthermore, in the same way as before, a simplified Gilbert-Elliot model was used to model the behavior of the channel and the values of parameters PLR and ABL differ in the different simulations. The FEC latency has been set to 1 s, which allows 0.5 s in the transmitter to carry out both the distortion model, perform the optimization procedure to obtain a suitable configuration and generate the redundancy (see Subsection Chapter 4). Finally, the overall distortion has been measured in terms of both the average percentage of affected pixels per frame. The values of the parameters employed in the simulations are listed in Table 5.3.

Figure 5.9 and Figure 5.10 depict the performance of both distortion models in a number of transmission scenarios. The first of them shows the simulations in which the GOP pattern IPP... was used to encode the video sequence and the second one the scenarios with GOP pattern IPBBP... As can be seen, the average distortion that is

Figure 5.10: Analysis of the performance of the system in a number of scenarios when employing both proposed elements: the UEP strategy and the full-context-aware distortion model. We use as reference the performance that results from using the non-interdependent-aware distortion model with our UEP framework, as in Chapter 4. Results for GOP prediction structure pattern IPBBBP...

introduced per frame when employing our model is invariably lower than that obtained when employing the non-interdependent-aware one in all the considered scenarios. Moreover, as expected, the resulting distortion values that are obtained with both approaches decrease as channel conditions improve and when increasing the FEC overhead. Furthermore, the gap between resulting values grows with the value of the channel's PLR. This evolution validates the introduction in our model of the masking effect caused by the potential loss of reference data. The reason is that taking the dependence into account makes the relative importance of packets in reference frames rise with respect to that of the packets, in the dependent frames, that is, it increases the relative difference of importance between packets, and this effect is more pronounced with high PLR values.

Additionally, one can observe that the results barely vary with the average length of the burst. This means that there exist enough available protection resources to ensure sufficient error decorrelation (for further explanation, refer to Subsection 3.3.2).

Finally, the influence of the type of prediction structure can be checked through comparing the graphs in the same position in both figures. As can be seen, the distortion values that are obtained when using pattern IPBBBP... are slightly better than those that result from using pattern IPP... As mentioned in a previous subsection, the reason

is that GOPs with pattern IPP... are more vulnerable to errors, as, contrary to what happens in GOPs with non-reference frames, any error will *always* propagate to all the subsequent frames in the GOP.

5.5 Conclusions

In this chapter, we have presented a lightweight packet-level distortion model for inferring the degradation that the transmission of a given packet in the packetized encoded video data stream is expected to introduced. This approach is intended for time-sensitive video transmission systems, as it requires no access to the original source and all the information it needs can be easily fetched from RTP and Network Abstraction Layer Unit (NALU) headers of the packets. The proposed distortion model is based on the inherent characteristics of the encoded video streams, particularly regarding data interdependence, and on the behavior of the communication channel. To determine the contribution of a given packet belonging to a certain frame to the overall distortion introduced in a given window of observance, and thus, its actual relevance, it not only considers its distorting power, that is, the measured distortion that the loss of this packet potentially introduces and the likelihood of this loss occurring. It also integrates what happens with the other packets in the stream it is in any way interrelated to.

The expected distortion associated with the transmission of a packet is computed in two steps. In the first one, we calculate the impact that its loss would have on the frame it belongs to and the ones in the window of observance that use it as reference, if this were an isolated occurrence. This impact is measured in terms of the percentage of pixels in those frames that are liable of changing their value from the original one. This first step also includes the likelihood of that occurrence happening. In the second step, the effect of inter-frame interdependence is taken into account. This step considers the likelihood of also losing packets that belong to reference frames and the effect that these losses have on the real distorting power of the packet under study.

The proposed approach has been validated in terms of both required processing time for its application and accuracy. Through the first set of experiments, we have proven that our model is suitable to be used in real-time video transmission applications. The results of the second set of experiments show that our approach can infer the real importance of each video data packet in the stream significantly better than widely-used non-interdependent-aware packet- and frame-level distortion models, hence enabling the possibility of refining QoS management mechanisms and improve their effectiveness.

Early versions of this work were included to support the unequal error protection strategy initially outlined in [11], and finally described in [12]. Later on, a more advanced version was published in [13]. The model presented in this paper is aware of the frame prediction structure but considers only to a limited extent the interdependence relations between data units.

Chapter 6

Conclusions and future work

6.1 Conclusions

The way in which video data is transported is changing considerably these days. Already now, IP networks are more frequently utilized to stream multimedia than the long-established RF-based structures, and video traffic accounts for the largest portion of the whole consumer IP traffic. Nevertheless, although the overall performance of packet-switched networks improves day by day, they present a major drawback when it comes to distributing video: transmission unreliability. To overcome this problem, QoS management mechanisms are typically used. In these scenarios, the Pro-MPEG COP3 codes are habitually used, due to their capability to cope with packet loss bursts and to their simplicity. However, their performance decreases as the packet loss rate is increased. On the other hand, unequal error protection strategies usually perform better than these codes, as they are capable of a better allocation and use of the available protection resources. However, their complexity usually exceeds that of the Pro-MPEG COP3 codes.

In this thesis, we have explored frameworks to generalize and optimize the Pro-MPEG COP3 codes, so as to extend the range of operating points, and improve their efficiency and effectiveness. This has been done, among other ways, by means of enabling that they can be used to protect unequally the data packet stream, but without compromising their simplicity and preserving their structural encoding and decoding methods.

To that end, we have first presented two extensions to the standard codes. The first of them preserves the uniform-protection nature of the standard version. It allows the use of one, two or three interleaving depths for boosting error decorrelation. As a result, the number of lost packets per repair packet is reduced and so is then the number of unrecoverable lost packets. Thus, the resulting overall performance is improved. The second one is an unequal error protection framework that allows the use of not only one but a number of matrices per protection block. The dimensions of each matrix utterly conditions the recovery rate of the packets arranged in it for protection. So, data packets can be grouped depending on their relevance and arranged in matrices of dissimilar dimensions. In this way, the recovery rate of each packet depends on the matrix used to protect it, which in turn depends on the relevance of the packet, among other things

that include the amount of available resources and the behavior of the communication channel. Using more matrices does not necessarily imply increasing the amount of used resources, but a different distribution of the available ones. In this sense, whereas more important packets are better protected than in the standard case, less important ones are worse protected than when applying the standard. The overall outcome, however, is a better performance in terms of introduced image degradation, as results both from the proof of concept experiments and the real ones.

Secondly, we have introduced a metaheuristic to guide the search of the most suitable configuration for the protection of data when using the proposed UEP framework. This optimization procedure is based on the hybridization of two broadly-used methods: simulated annealing (SA) and tabu search (TS). The core procedures of these metaheuristics have been modified to consider specific characteristics of the scenario, particularly in terms of imposed processing time, and so find near-optimal solutions under those conditions. First, to make the method able to perform in real time, a scenario-aware solution space reduction is carried out in the first place. In this way, the number of considered solution is drastically decreased, whereas barely affecting the result. Thus, the optimization procedure starts from a more advantageous situation. This procedure includes specific structures of both well-known metaheuristics: SA constitutes the core of the proposed strategy and TS provides memory structure to more effectively guide the search. The resulting hybridization considers the importance of the video data packets, the available resources and the behavior of the channel to periodically find sufficiently-good protection configurations in terms of number of matrices and their dimensions. The proposed metaheuristic invariably fulfill real-time constraints at the expense of frequently not selecting the optimal but a near-optimal solution, which, nevertheless habitually allows a better handling of resources that when applying the standard case.

Finally, we have introduced a packet-independence-aware distortion model. This model, although fairly independent on the scenario, is intended to be used with UEP schemes like the one proposed in this thesis. The presented approach estimates the contribution of any given video packet to the resulting overall expected distortion of the sequence presented to the user, and thus, how important it actually is. To do that, it considers upper-level features of the input encoded video stream and the behavior of the communication channel. The actual distortion that a packet is expected to introduce is estimated considering: (i) the image degradation that the loss of the packet will potentially introduce in the frame it belongs to and the ones that use this frame as reference, (ii) the likelihood of this packet being the first lost packet in the frame it belongs to (and so this packet being responsible for all the distortion steaming from this frame considered in the previous point), and (iii) the masking effect that the potential loss of the different packets in the reference frames/slices has on the actual importance of the packet under consideration. This quite simple model allows a more accurate assessment of the relative importance of the packets in the block, thus enabling the possibility of further refining the proposed protection mechanism and so improve its effectiveness.

6.2 Future work

As a general observation, given that a large proportion of the work reflected in this thesis has been built around a certain type of channel codes, the Pro-MPEG COP3 codes, a key task for the future would necessarily consist in the generalization of some of the ideas that so far have only been applied to these particular codes. In this way, a wide range of standard channel codes currently carrying out a uniform protection (e.g. Reed-Solomon codes) or imposing some kind of uncontrolled internal UEP that is not aware of the actual relevance of each packet (e.g. Low-density parity-check codes -LDPC-, LT codes, Raptor codes) could benefit from them and improve their performance.

Moving from the general to the particular, that is, considering separately the different elements presented in this thesis, we have identified several possible outstanding enhancements:

- **Protection scheme:** regarding the proposed EEP extension to the Pro-MPEG COP3, it could be easily extended to include not only up to three, but an arbitrary number of dimensions. In scenarios where the number of data and repair packets considered per block is sufficiently (considerably) great, the use of more dimensions could result in a better performance of the codes, as the likelihood of encountering two or more lost packets per protection packet would be reduced and with that the overall likelihood of not being able to recover them.

Regarding the UEP approach, current imposed conditions state that all data packets in the window are necessarily protected, regardless of their importance. Furthermore, each data packet can only be arranged in one matrix. Therefore, the first enhancement would consist in a further generalization of the strategy that would include the possibility of both allowing not protecting the set of packets with the lowest expected distortion scores and enabling that data packets (most likely, the set of most important ones) could be arranged in more than one matrix.

- **Protection scheme optimization:** if the latter enhancements to the UEP version were applied, the number of feasible configurations would notably increase. For that reason, it would be first necessary to impose new conditions to further reduce the cardinality of the solution space. Moreover, a new way to measure the distance between solutions would be required, as two configurations with the same number of matrices and dimensions could introduce different levels of distortion, since they could internally include different sets of data packets. Finally, the procedure would likely require some sort of parallelization, either within or outside the inner loop, so as to speed up the process and therefore be able to comply with strict time conditions.
- **Distortion model:** the current version of the model works with independent windows of observance. If the imposed time restrictions allow to include whole GOPs within each window, it is possible to properly consider all the frame interdependence relations (assuming close GOPs) and so obtain an accurate result for each

data packet. This is the situation that we have considered in this thesis. However, if the imposed time restrictions are harder or GOPs are too large, packets belonging to a certain GOP will be considered in different windows. Moreover, a single window will include data belonging to more than one GOP. This new factors will result in inaccurate values regarding the distortion that a packet is expected to introduce. Thus, to accurately assess the relevance of a packet in any situation, regardless of whether all the references of the frame the packet belongs to are in the same or different window, some additional information should be used. This extra information to be passed from window to window, until the considered GOP is over, would consist of data such as the type of the frames in the GOP or the number of packets in each frame. Moreover a further generalization of the model that properly formalizes the case where frames are composed of two or more slices would be of great interest.

Additionally, the proposed approach could also be extended to consider how different errors are perceived by users. For instance, two errors that result in changing the value of an equivalent number of pixels throughout the sequence might not be perceived equally distorting if in one case all the wrong pixels are contained in just one frame and in the second one those pixels are spread along several frames.

Bibliography

- [1] J. Postel, “Internet Protocol,” Sep. 1981, RFC 791.
- [2] Cisco, “Visual Networking Index: Forecast and Methodology, 2015-2020,” 2016. [Online]. Available: <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf>
- [3] —, “Visual Networking Index: Forecast and Methodology, 2009-2014,” 2010. [Online]. Available: http://large.stanford.edu/courses/2010/ph240/abdul-kafi1/docs/white_paper_c11-481360.pdf
- [4] ETSI TS 102 034, “Digital Video Broadcasting (DVB): Transport of MPEG-2 TS-based DVB services over IP-based networks,” Aug. 2009.
- [5] ETSI TS 102 796, “Hybrid Broadcast Broadband TV,” Oct. 2015.
- [6] C. Díaz, C. Hellge, J. Cabrera, F. Jaureguizar, and T. Schierl, “Enhancement of Pro-MPEG COP3 codes and application to layer-aware FEC protection of two-layered video transmission,” in *20th IEEE International Conference on Image Processing (ICIP)*, Sep. 2013, pp. 1598–1602.
- [7] C. Díaz, J. Cabrera, F. Jaureguizar, and N. García, “An extension to the PRO-MPEG COP3 codes for unequal error protection of real-time video transmission,” in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, Sep. 2015, pp. 927–931.
- [8] —, “Packet signaling for systematic coding schemes with arbitrary data-to-repair-packet connection setups,” in *2015 IEEE International Symposium on Consumer Electronics (ISCE)*, Jun. 2015, pp. 1–2.
- [9] —, “Optimization of unequal error protection AL-FEC scheme via adapted simulated annealing,” *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 345–346, Jan. 2016.
- [10] —, “Application-Layer FEC Scheme Configuration Optimization via Hybrid Simulated Annealing,” *IEEE Transactions on Broadcasting*, vol. PP, no. 99, pp. 1–15, 2017.

-
- [11] —, “A video-aware FEC-based unequal loss protection scheme for RTP video streaming,” *2011 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 3–4, Jan. 2011.
 - [12] —, “A video-aware FEC-based unequal loss protection system for video streaming over RTP,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 523–531, May 2011.
 - [13] —, “Adaptive protection scheme for MVC-encoded stereoscopic video streaming in IP-based networks,” in *2012 IEEE Visual Communications and Image Processing (VCIP)*, Nov. 2012, pp. 1–6.
 - [14] D. B. Levey and S. McLaughlin, “The statistical nature of impulse noise interarrival times in digital subscriber loop systems,” *Signal Processing*, vol. 82, no. 3, pp. 329–351, 2002.
 - [15] Broadband Forum, “DQS: DSL Quality Management Techniques and Nomenclature,” pp. 1–61, Aug. 2012, TR-197.
 - [16] V. Cerf, Y. Dalal, and C. Sunshine, “Specification of Internet Transmission Control Program,” Dec. 1974, RFC 675.
 - [17] J. Postel, “Transmission Control Protocol,” Sep. 1981, RFC 793.
 - [18] M. Allman, V. Paxson, and W. Stevens, “TCP Congestion Control,” Apr. 1999, RFC 2581.
 - [19] J. Postel, “User Datagram Protocol,” vol. Aug., 1980, RFC 768.
 - [20] E. Kohler, M. Handley, and S. Floyd, “Datagram Congestion Control Protocol (DCCP),” Mar. 2006, RFC 4340.
 - [21] R. Stewart, “Stream Control Transmission Protocol,” Sep. 2007, RFC 4960.
 - [22] C. Jin, S. H. Low, D. X. Wei, W. Bartek, A. Tang, and H. Choe, “Method and apparatus for network congestion control using queue control and one-way delay measurements,” Apr. 2010, US 7693052 B2.
 - [23] C. Jin, G. S. Lee, S. Low, D. Ng, and R. Witt, “Dynamic adjustment of receive window utilized by a transmitting device,” May 2014, US 8730799 B2.
 - [24] R. Hamilton, J. Iyengar, I. Swett, and A. Wilk, “QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2,” Jul. 2016, Internet Draft.
 - [25] E. R. Fielding and E. J. Reschke, “Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing,” no. RFC 7230, Mar. 2014.
 - [26] A. C. Begen, T. Akgul, and M. Baugher, “Watching Video over the Web: Part 1: Streaming Protocols,” *IEEE Internet Computing*, vol. 15, no. 2, pp. 54–63, Mar. 2011.

- [27] ISO/IEC 23009-1:2014, “Dynamic adaptive streaming over HTTP (DASH) Part 1 Media presentation description and segment formats,” Tech. Rep., 2014.
- [28] R. Pantos and W. May, “HTTP Live Streaming,” Apr. 2017, IETF Draft.
- [29] M. Corporation, “Live and On-Demand Video with Silverlight and IIS Smooth Streaming,” Feb. 2010, Whitepaper. [Online]. Available: <https://www.iis.net/learn/media/smooth-streaming/smooth-streaming-transport-protocol>
- [30] R. Doverspike, G. Li, K. N. Oikonomou, K. K. Ramakrishnan, R. K. Sinha, D. Wang, and C. Chase, “Designing a Reliable IPTV Network,” *IEEE Internet Computing*, vol. 13, no. 3, pp. 15–22, May 2009.
- [31] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications Status,” Jul. 2003, RFC 3550.
- [32] H. Schulzrinne and S. Casner, “RTP Profile for Audio and Video Conferences with Minimal Control,” Jul. 2003, RFC 3551.
- [33] D. Hoffman, G. Fernando, V. Goyal, and M. R. Civanlar, “RTP Payload Format for MPEG1/MPEG2 Video,” Jan. 1998, RFC 2250.
- [34] Y.-K. Wang, R. Even, T. Kristensen, and R. Jesup, “RTP Payload Format for H.264 Video,” May 2011, RFC 6184.
- [35] Y.-K. Wang, Y. Sanchez, T. Schierl, S. Wenger, and M. Hannuksela, “RTP Payload Format for High Efficiency Video Coding (HEVC),” no. RFC 7798, Mar. 2016.
- [36] H. Schulzrinne, A. Rao, and R. Lanphier, “Real Time Streaming Protocol (RTSP),” Apr. 1998, RFC 2326.
- [37] M. Volk, J. Guna, A. Kos, and J. Bester, “IPTV Systems, Standards and Architectures: Part II - Quality-Assured Provisioning of IPTV Services within the NGN Environment,” *IEEE Communications Magazine*, vol. 46, no. 5, pp. 118–126, May 2008.
- [38] J. Asghar, F. Le Faucheur, and I. Hood, “Preserving Video Quality in IPTV Networks,” *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 386–395, Jun. 2009.
- [39] Cisco, “DiffServ - The Scalable End-to-end Quality of Service Model,” Aug. 2005. [Online]. Available: http://www.cisco.com/en/US/technologies/tk543/tk766/technologies_white_paper09186a00800a3e2f.pdf
- [40] —, “Comparing Traffic Policing and Traffic Shaping for Bandwidth Limiting,” May 2014. [Online]. Available: <http://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-policing/19645-policevsshape.pdf>
- [41] S. Blake, D. Black, M. A. Carlson, E. Davies, Z. Wang, and W. Weiss, “An Architecture for Differentiated Services,” Dec. 1998, RFC 3260.

-
- [42] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," Jun. 1994, RFC 1633.
 - [43] Cisco, "Understanding Selective Packet Discard (SPD)," Jul. 2005. [Online]. Available: <http://www.cisco.com/c/en/us/support/docs/routers/12000-series-routers/29920-spd.pdf>
 - [44] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *The Bell System Technical Journal*, pp. 1253–1265, Sep. 1960.
 - [45] E. O. Elliott, "Estimates of Error Rates for Codes on Burst-Noise Channels," *Bell System Technical Journal*, vol. 42, no. 5, pp. 1977–1997, Sep. 1963.
 - [46] V. Pullano, R. Skupin, G. Corazza, C. Hellge, and T. Schierl, "Modeling effects of impulse noise on Application-Layer FEC in DSL channels," in *2013 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. IEEE, Jun. 2013, pp. 1–5.
 - [47] 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*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.
 - [48] C. Hellge, D. Gomez-Barquero, T. Schierl, and T. Wiegand, "Layer-aware forward error correction for mobile broadcast of layered media," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 551–562, Jun. 2011.
 - [49] P. Pérez, J. Macías, J. J. Ruiz, and N. García, "Effect of packet loss in video quality of experience," *Bell Labs Technical Journal*, vol. 16, no. 1, pp. 91–104, Jun. 2011.
 - [50] L. Chen and G. Liu, "A Delivery System for Streaming Video Over DiffServ Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 9, pp. 1255–1259, sep 2010.
 - [51] P. Pérez and N. García, "Lightweight multimedia packet prioritization model for unequal error protection," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 132–138, Feb. 2011.
 - [52] V. Miguel, J. Cabrera, F. Jaureguizar, and N. García, "Distribution of high-definition video in 802.11 wireless home networks," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 53–61, Feb. 2011.
 - [53] Y. Huo, C. Hellge, T. Wiegand, and L. Hanzo, "A tutorial and review on inter-layer fec coded layered video streaming," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–1, 2015.
 - [54] Pro-MPEG Forum Code of Practice 3 release 2, "Transmission of professional MPEG-2 transport streams over IP networks," Jul. 2004.

- [55] SMPTE ST 2022-1, “Forward error correction for real-time video/audio transport over IP networks,” May 2007.
- [56] ETSI TR 102 993, “Digital Video Broadcasting (DVB); Upper Layer FEC for DVB Systems,” Feb. 2011.
- [57] ETSI TS 102 542-3-2, “Digital Video Broadcasting (DVB): Guidelines for the implementation of DVB-IP Phase 1 specifications. Part 3: Error Recovery. Subpart 2: Application Layer FEC,” May 2011.
- [58] A. Begen, “RTP Payload Format for 1-D Interleaved Parity Forward Error Correction (FEC),” Oct. 2010, RFC 6015.
- [59] A. Shokrollahi and M. Luby, “Raptor codes,” *Foundations and Trends in Communications and Information Theory*, vol. 6: No 3-4, pp. 213–322, 2011.
- [60] DVB Bluebook A115, “DVB Application Layer FEC Evaluations,” May 2007.
- [61] F. Battisti, M. Carli, E. Mammi, and A. Neri, “A study on the impact of AL-FEC techniques on TV over IP Quality of Experience,” *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, p. 86, 2011.
- [62] M. Watson, A. Begen, and V. Roca, “Forward Error Correction (FEC) Framework,” Oct. 2011, RFC 6363.
- [63] M. Watson, T. Stockhammer, and M. Luby, “Raptor Forward Error Correction (FEC) Schemes for FECFRAME,” Aug. 2012, RFC 6681.
- [64] M. Watson, T. Stockhammer, and M. G. Luby, “RTP Payload Format for Raptor Forward Error Correction (FEC),” Aug. 2012, RFC 6682.
- [65] J. Lacan, V. Roca, J. Peltotalo, and S. Peltotalo, “Reed-Solomon Forward Error Correction (FEC) Schemes,” Apr. 2009, RFC 5510.
- [66] V. Roca, M. Cunche, J. Lacan, A. Bouabdallah, and K. Matsuzono, “Simple Reed-Solomon Forward Error Correction (FEC) Scheme for FECFRAME,” Feb. 2013, RFC 6865.
- [67] A. Li, “RTP Payload Format for Generic Forward Error Correction,” Dec. 2007, RFC 5109.
- [68] T. Cheng, K. Peng, F. Yang, J. Song, and Y. Zhixing, “A Near-Capacity MIMO Coded Modulation Scheme for Digital Terrestrial Television Broadcasting,” *IEEE Transactions on Broadcasting*, vol. 61, no. 3, pp. 367–375, 2015.
- [69] C. Ye, G. Ozcan, M. C. Gursoy, and S. Velipasalar, “Multimedia Transmission Over Cognitive Radio Channels Under Sensing Uncertainty,” *IEEE Transactions on Signal Processing*, vol. 64, no. 3, pp. 726–741, Feb. 2016.

-
- [70] S. K. Srinivasan, J. Vahabzadeh-Hagh, and M. Reisslein, "The Effects of Priority Levels and Buffering on the Statistical Multiplexing of Single-Layer H.264/AVC and SVC Encoded Video Streams," *IEEE Transactions on Broadcasting*, vol. 56, no. 3, pp. 281–287, Sep. 2010.
 - [71] C. Zhou, C.-W. Lin, X. Zhang, and Z. Guo, "A Novel JSCC Scheme for UEP-Based Scalable Video Transmission Over MIMO Systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 6, pp. 1002–1015, Jun. 2015.
 - [72] S. Sandberg and N. von Deetzen, "Design of bandwidth-efficient unequal error protection LDPC codes," *IEEE Transactions on Communications*, vol. 58, no. 3, pp. 802–811, Mar. 2010.
 - [73] A. Vosoughi, P. C. Cosman, and L. B. Milstein, "Joint Source-Channel Coding and Unequal Error Protection for Video Plus Depth," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 31–34, 2015.
 - [74] Y. C. Chang, S. W. Lee, and R. Komiya, "A low complexity hierarchical QAM symbol bits allocation algorithm for unequal error protection of wireless video transmission," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 3, pp. 1089–1097, Aug. 2009.
 - [75] H. Ha and C. Yim, "Layer-weighted unequal error protection for scalable video coding extension of H.264/AVC," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 2, pp. 736–744, May 2008.
 - [76] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, Sep. 2003.
 - [77] Y. Wu, S. Kumar, F. Hu, Y. Zhu, and J. D. Matyjas, "Cross-Layer Forward Error Correction Scheme Using Raptor and RCPC Codes for Prioritized Video Transmission Over Wireless Channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 6, pp. 1047–1060, Jun. 2014.
 - [78] T. Fang and L.-P. Chau, "GOP-based channel rate allocation using genetic algorithm for scalable video streaming over error-prone networks," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1323–1330, Jun. 2006.
 - [79] V. Stanković, R. Hamzaoui, and Y. Charfi, "Real-time unequal error protection algorithms for progressive image transmission," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 10, pp. 1526–1535, Dec. 2003.
 - [80] L. Cao, "On the Unequal Error Protection for Progressive Image Transmission," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2384–2388, Sep. 2007.
 - [81] M. Gendreau and J.-Y. Potvin, Eds., *Handbook of metaheuristics*. Springer, 2010.

- [82] C. R. Reeves, “Genetic algorithms,” in *Handbook of metaheuristics*. Springer, 2010.
- [83] M. Dorigo and T. Stützle, “Ant colony optimization,” in *Handbook of metaheuristics*. Springer, 2010.
- [84] R. Poli, J. Kennedy, and T. Blackwell, “Particle swarm optimization. An overview,” *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, Jun. 2007.
- [85] A. G. Nikolaev and S. H. Jacobson, “Simulated annealing,” in *Handbook of metaheuristics*. Springer, 2010.
- [86] M. Gendreau and J.-Y. Potvin, “Tabu search,” in *Handbook of metaheuristics*. Springer, 2010.
- [87] C. Voudouris, E. P.K. Tsang, A. Alsheddy, “Guided local search,” in *Handbook of metaheuristics*. Springer, 2010.
- [88] R. S. Sexton, R. E. Dorsey, and J. D. Johnson, “Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing,” *European Journal of Operational Research*, vol. 114, pp. 589–601, 1999.
- [89] B. D. Mori, H. F. de Castro, and K. L. Cavalca, “Development of hybrid algorithm based on simulated annealing and genetic algorithm to reliability redundancy optimization,” *International Journal of Quality & Reliability Management*, vol. 24, no. 9, pp. 972–987, 2007.
- [90] S. Kannan, S. M. R. Slochanal, and N. P. Padhy, “Application and Comparison of Metaheuristic Techniques to Generation Expansion Planning Problem,” *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 466–475, Jan. 2005.
- [91] M. Garcia-Lozano, M. A. Lema, S. Ruiz, and F. Minerva, “Metaheuristic Procedure to Optimize Transmission Delays in DVB-T Single Frequency Networks,” *IEEE Transactions on Broadcasting*, vol. 57, no. 4, pp. 876–887, Dec. 2011.
- [92] D. Cheng, J. Rao, C. Jiang, and X. Zhou, “Elastic Power-Aware Resource Provisioning of Heterogeneous Workloads in Self-Sustainable Datacenters,” *IEEE Transactions on Computers*, vol. 65, no. 2, pp. 508–521, Feb. 2016.
- [93] M. F. Uddin, H. M. K. AlAzemi, and C. Assi, “Optimal Flexible Spectrum Access in Wireless Networks with Software Defined Radios,” *IEEE Transactions on Wireless Communications*, vol. 10, no. 1, pp. 314–324, Jan. 2011.
- [94] Y. A. Katsigiannis, P. S. Georgilakis, and E. S. Karapidakis, “Hybrid Simulated Annealing-Tabu Search Method for Optimal Sizing of Autonomous Power Systems With Renewables,” *IEEE Transactions on Sustainable Energy*, vol. 3, no. 3, pp. 330–338, Jul. 2012.

-
- [95] K. Lenin, B. R. Reddy, and M. Suryakalavathi, "Hybrid Tabu search-simulated annealing method to solve optimal reactive power problem," *International Journal of Electrical Power & Energy Systems*, vol. 82, pp. 87–91, Nov. 2016.
- [96] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Optimization by simulated annealing*. American Association for the Advancement of Science, May 1983, vol. 220, no. 4598.
- [97] D. Zhang, Y. Liu, R. M'Hallah, and S. C. Leung, "A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems," *European Journal of Operational Research*, vol. 203, no. 3, pp. 550–558, Jun. 2010.
- [98] X. Yao, "Dynamic Neighbourhood Size in Simulated Annealing," in *Proc. of Int'l Joint Conf. on Neural Networks (IJCNN'92)*, 1992, pp. 1–7.
- [99] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549, Jan. 1986.
- [100] T. Wiegand, G. 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, Jul. 2003.
- [101] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
- [102] Z. He and X. Hongkai, "Transmission Distortion Analysis for Real-Time Video Encoding and Streaming Over Wireless Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 9, pp. 1051–1062, Sep. 2006.
- [103] C.-I. Kuo, C.-H. Shih, C.-K. Shieh, W.-S. Hwang, and C.-H. Ke, "Modeling and Analysis of Frame-Level Forward Error Correction for MPEG Video over Burst-Loss Channels," *Applied Mathematics & Information Sciences*, vol. 8, no. 4, pp. 1845–1853, Jul. 2014.
- [104] E. Baccaglini, G. Marchetto, T. Tillo, and G. Olmo, "Efficient slice-aware H.264/AVC video transmission over time-driven priority networks," *International Journal of Communication Systems*, vol. 27, no. 12, pp. 3822–3836, Dec. 2014.
- [105] Y. Hou, J. Xu, W. Xiang, M. Ma, and J. Lei, "Near-Optimal Cross-Layer Forward Error Correction Using Raptor and RCPC Codes for Prioritized Video Transmission Over Wireless Channels," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2016.
- [106] G. J. Sullivan, J. M. Boyce, Y. Chen, J. R. Ohm, C. A. Segall, and A. Vetro, "Standardized extensions of high efficiency video coding (hevc)," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1001–1016, Dec. 2013.

- [107] A. Vetro, T. Wiegand, and G. Sullivan, "Overview of the Stereo and Multiview Video Coding Extensions of the H.264/MPEG-4 AVC Standard," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626–642, Apr. 2011.
- [108] Y. Zhou, H. C., W. Xiang, and W. Feng, "Channel Distortion Modeling for Multi-View Video Transmission Over Packet-Switched Networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 11, pp. 1679–1692, Nov. 2011.
- [109] H. D. Pham and S. Vafi, "Motion-energy-based unequal error protection for H.264/AVC video bitstreams," *Signal, Image and Video Processing*, vol. 9, no. 8, pp. 1–8, Nov. 2015.
- [110] J. Shih and W. Tsai, "A new unequal error protection scheme based on FMO," *Multimedia Tools and Applications*, vol. 47, no. 3, pp. 461–476, Aug. 2009.
- [111] Z. Luo, L. Song, S. Zheng, and N. Ling, "Raptor Codes Based Unequal Protection for Compressed Video According to Packet Priority," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2208–2213, Dec. 2013.
- [112] M. Schier and M. Welzl, "Optimizing Selective ARQ for H.264 Live Streaming: A Novel Method for Predicting Loss-Impact in Real Time," *IEEE Transactions on Multimedia*, vol. 14, no. 2, pp. 415–430, Apr. 2012.
- [113] K. Go, M. Kim, S. Kang, and Y. Yoon, "A systematic reallocation and prioritization scheme for error-resilient transmission of video packets," *Multimedia Tools and Applications*, Feb. 2016.
- [114] X.-W. Yao, W.-L. Wang, S.-H. Yang, Y.-F. Cen, X.-M. Yao, and T.-Q. Pan, "IPB-frame adaptive mapping mechanism for video transmission over IEEE 802.11e WLANs," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 2, pp. 5–12, Apr. 2014.
- [115] S. Tao, J. G. Apostolopoulos, and R. Guerin, "Real-Time Monitoring of Video Quality in IP Networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 5, pp. 1052–1065, Oct. 2008.
- [116] Z. Li, J. Chakareski, X. Niu, Y. Zhan, and W. Gu, "Modeling and Analysis of Distortion Caused by Markov-Model Burst Packet Losses in Video Transmission," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 917–931, 2009.
- [117] Y. J. Liang, J. G. Apostolopoulos, and B. Girod, "Analysis of Packet Loss for Compressed Video: Effect of Burst Losses and Correlation Between Error Frames," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 7, pp. 861–874, Jul. 2008.

- [118] Z. Li, J. Chakareski, L. Shen, and L. Wang, “Video Quality in Transmission over Burst-Loss Channels: A Forward Error Correction Perspective,” *IEEE Communications Letters*, vol. 15, no. 2, pp. 238–240, 2011.
- [119] K. Stuhlmüller, N. Färber, M. Link, and B. Girod, “Analysis of video transmission over lossy channels,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012–1032, Jun. 2000.
- [120] Y. J. Liang, J. G. Apostolopoulos, and B. Girod, “Analysis of packet loss for compressed video: does burst-length matter?” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*., vol. 5, 2003, pp. V–684–7.
- [121] D. C. Robinson and A. Villegas, “Intelligent wrapping of video content to lighten downstream processing of video streams,” Jun. 2009, eP 2071850 A1.