UNIVERSITÀ DEGLI STUDI DI TRIESTE

XXVII CICLO DEL DOTTORATO DI RICERCA IN
INGEGNERIA DELL'INFORMAZIONE

Audio/Video Transmission over IEEE 802.11e Networks: Retry Limit Adaptation and Distortion Estimation

SETTORE SCIENTIFICO DISCIPLINARE: ING-INF/03

Dottorando:
Riccardo CORRADO

Coordinatore:
Chiar.mo Prof. Roberto VESCOVO

Supervisore di Tesi:
Chiar.mo Prof. Fulvio BABICH

Co-Supervisore di Tesi:
Chiar.mo Prof. Massimiliano COMISSO

ANNO ACCADEMICO 2014/2015

“We are what we think. All that we are, arises with our thoughts. With our thoughts, we make the world.”

Buddha

Abstract

This thesis concerns the audio and video transmission over wireless networks adopting the family of the IEEE 802.11x standards. In particular, the original contribution of this study involves the discussion of four issues: the adaptive retransmission of video packets with distortion and delay requirements, the reliability of different video quality assessments for retry limit adaptation purposes, the fast distortion estimation, and the joint adaptation of the retry limits of both voice and video packets. The material presented in this dissertation is the result of four-year studies performed within the Telecommunication Group of the Department of Engineering and Architecture at the University of Trieste during the course of Doctorate in Information Engineering.

Multimedia applications have registered a tremendous increase in the last years. The request of video data represents a big portion of the total data traffic on the Internet, and hence improving the Quality of Service (QoS) at the end-user is a very important topic in nowadays telecommunication research. To introduce QoS control at the MAC layer of an 802.11 network, task group e (TGe) developed the 802.11e amendment, which extends the functionalities of the legacy Distributed Coordination Function (DCF) by adopting the Enhanced Distributed Channel Access (EDCA). The EDCA enables a prioritization of the traffic during the contention period by defining four Access Categories (ACs): Voice (VO), Video (VI), Best Effort (BE), and BacK ground (BK), whose differentiation is based on parameters such as transmission opportunity, arbitration inter

frame space (AIFS), minimum and maximum contention windows. Another parameter associated to each AC is the maximum retry limit associated to each packet, which represents the maximum number of times that a packet can be retransmitted. Even if the EDCA settings are specified to provide a higher priority to the voice and video ACs, collisions involving audio/video packets may still occur, thus making necessary proper prioritization policies of the traffic. Accordingly, several studies focused on this issue, aiming to prioritize audio and video signals according to an importance metric. More precisely, in scientific literature are present solutions for audio transmission, based on pre-computers descriptors of the audio signals, multi-resolution techniques for 3D audio rendering, prioritization mechanisms relying on the perceptual quality of audio signals exposed to packet loss. Furthermore, for the transmission of video flows, solutions based on lightweight prioritization schemes or on more sophisticated pixel-based techniques have been presented. An interesting possibility, is represented by the manipulation of the retry limit associated to each packet. Several studies have investigated this issue. The main objective of these methods is the adaptation of the number of retransmissions to the acceptable delay and the perceived distortion. More precisely, for the packets containing information whose loss would imply a high distortion on the video sequence, it is necessary to set a higher retry limit value. Instead, for the packets with information less important for the decoding process, the corresponding retry limit can be lower. This leads to the derivation of elaborate optimization strategies, able to provide significant performance improvements with respect to those achievable using the 802.11e default settings.

In the context of wireless networks adopting the standard 802.11e, the core of this thesis is the development of fast algorithms capable to calculate the best retry limit for each packet in the audio and video queue, aiming to choose the best retry limit in accordance with the distortion associated to that packet, for providing a better multimedia quality

at end user. The novel aspects of this study are represented by the theoretical and numerical modeling which account for the presence of the other AC's in the evaluation of the best retry limit, but maintaining always a very low computational cost for the evaluation of the network behavior. Furthermore, a study on the best quality assessment to use with retransmission purposes is portrayed in this thesis, trying to find not only the most suitable quality assessment to adopt, but also finding an approximation of the values provided by it, and requiring low computational cost, thus making the adoption of this index suitable for scenarios characterized by low delay.
More precisely, the first proposed algorithm rely on a distortion estimation method that is capable to reliably evaluate the Mean Square Error (MSE)-based distortion of a video sequence. The proposed algorithm, which requires low computational cost, evaluates the retry limits accounting for the presence of the other access categories and using the available distortion values. In order to find the best video quality assessment to employ in these retry limit adaptation scenarios, a second study on the most suitable video quality assessment has been portrayed. The study has shown that the Structural SIMilarity (SSIM)-based distortion can outperform the MSE-based one, when it is used with adaptive retransmission purposes. On the other hand, if the SSIM results to be more suitable than the MSE, it requires a high computational cost to be evaluated. To overcome the drawback introduced by the SSIM, in terms of computational cost, an algorithm capable to evaluate the SSIM-based distortion in low CPU times has been created, making this quality assessment adoptable also in transmissions characterized by low acceptable delays. Finally the last part of the thesis has focused on extending the first proposed algorithm, to a scenario involving the transmission of not only video contents, but also audio contents, which are usually both present in multimedia flows.

This thesis is organized in two parts. The first part provides the background material, while the second part is dedicated to the original results. With reference to the first part, the fundamentals of multimedia transmission over Wi-Fi Networks are briefly summarized in the first chapter. An overview of the most common audio and video coding standards are presented in the second chapter, focusing mainly on the two standards adopted in the second part of this dissertation, the G.729 speech coding standard, and the H.264 Video coding standard. The third chapter introduces the most significant aspects of distributed wireless networks, both considering the Physical (PHY) and the Medium Access Control (MAC) layers.

The second part describes the original results obtained in the field of 802.11e retry limit adaptation and low-cost SSIM-based estimation. In particular, the fourth chapter presents an algorithm for the fast evaluation of the best retry limit associated to each video packet in an 802.11e contention-based scenario. This algorithm, in accordance with the estimated distortion and the maximum cumulated delay for each packet, selects the best retry limit with a low computational burden. Given that the MSE often fails in terms of the visual perception of the quality of the scene, chapter five places its focus on comparing, in a retry limit adaptation scenario, the adoption of the MSE with the adoption of another video quality assessment: the SSIM. The aim is to exploit the possibility of adopting a video quality assessment capable to better measure the fidelity of the video distortion. Chapter six focuses on overcoming the drawback introduced by the SSIM, represented by the high computation burden required for evaluating the SSIM-based distortion. In this chapter, a fast distortion estimation based on the structural similarity for videos encoded with the H.264 standard is presented. Finally the last chapter of this dissertations, extends the model presented in chapter four, aiming to jointly evaluate the best retry limits for audio and video flows, both present in multimedia transmissions.

The intent of the work presented hereafter, is to develop and test computationally cheap solutions for improving the quality of audio/video delivery in 802.11-based wireless networks, focusing on the careful selection of the retransmission strategy and the reliable estimation of the content distortion.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Fulvio Babich for the continuous support on my Ph.D study and for incenting me to widen my research from various perspectives. Besides my advisor, I would like to thank my co-supervisor Prof. Massimiliano Comisso, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research.

A special thanks go to my family, Roberto, Annamaria, Enrico and Pamela which supported me through all these years never failing in providing me all the support and help I needed, letting me focus only on studies and research. They are the reason why I can be here today. A big thanks goes to Patchanee whose endless help gave me the chance to face all the problems and brightened my days. She made me understand that if a problem is fixable then there is no need to worry and if it's not fixable, then there is no help in worrying. Thanks also to Giovanni, Matteo, Asma and Christian for the fun times which made me able to recover from the though periods. Furthermore a big thanks to Tihitena for the beautiful conversations and comforting words always spent for me.

Last but not the least, my sincere thanks go to my fellow labmates for the stimulating discussions, for the help and for all the time shared in laboratory in these years.

Contents

Abbreviations

AAC	**A**dvanced **A**udio **C**oding
ACK	**A**cknowledgment
ACR LQ	**A**bsolute **C**ategory **R**ating **L**istening **Q**quality
AIFS	**A**rbitration **I**nter-**F**rame **S**pacing
AIFSN	**A**rbitration **I**nter-**F**rame **S**pacing **N**umber
AP	**A**ccess **P**oint
ASPEC	**A**daptive **S**pectral **P**erceptual **E**ntropy **C**oding
AVC	**A**dvanced **V**ideo **C**oding
BSS	**B**asic **S**ervice **S**et
CCIR	**C**onsultative **C**ommittee **I**nternational **R**adio
CCK	**C**omplementary **C**ode **K**eying
CCR	**C**omparison **C**ategory **R**ating
CELP	**C**ode **E**xcited **L**inear **P**rediction
CF	**C**ontention **F**ree
CF-Poll	**C**ontention **F**ree-**P**oll
CFP	**C**ontention **F**ree **B**ursting
CFP	**C**ontention **F**ree **P**eriod
CGS	**C**oarse **G**rain **S**calability
CS-ACELP	**C**onjugate-**S**tructure **A**lgebraic-**C**ode-**E**xcited **L**inear **P**rediction
CSMA	**C**arrier **S**ense **M**ultiple **A**ccess
CSMA/CD	**C**arrier **S**ense **M**ultiple **A**ccess with **C**ollision **D**etection
CSMA/CA	**C**arrier **S**ense **M**ultiple **A**ccess with **C**ollision **A**voidance
CTS	**C**lear **T**o **S**end
CW	**C**ontention **W**indow

DAM	**D**iagnostic **A**cceptability **M**easure
DBPSK	**D**ifferential **I**inary **P**hase **S**hift **K**eying
DCF	**D**istributed **C**oordination **F**unction
DCR	**D**egradation **C**ategory **R**ating
DCT	**D**iscrete **C**osine **T**ransform
DIFS	DCF **I**nter-**F**rame **S**pace
DQPSK	**D**ifferential **Q**uadrature **P**hase **S**hift **K**eying
DS	**D**istribution **S**ystem
DSCQS	**D**ouble **S**timulus **C**ontinuous **Q**uality **S**cale
DSM	**D**istribution **S**ystem **M**edium
DSSS	**D**irect **S**equence **S**pread **S**pectrum
DVMRP	**D**istance **V**ector **M**ulticast **R**outing **P**rotocol
DWN	**D**istributed **W**ireless **N**etwork
EDA	**E**xponential **D**istortion **A**lgorithm
EDCA	**E**nhanced **D**istributed-**C**hannel **A**ccess
EDCAF	**E**nhanced **D**istributed-**C**hannel **A**ccess **F**unction
ESS	**E**xtended **S**ervice **S**et
FHSS	**F**requency **H**opping **S**pread **S**pectrum
FGS	**F**ine **G**rain **S**calability
FR	**F**ull **R**eference
FTP	**F**ile **T**ransfer **P**rotocol
GFSK	**G**aussian **F**requency **S**hift **K**eying
GOP	**G**roup **O**f **P**ictures
HCCA	**HCF** **C**ontrolled **C**hannel **A**ccess
HDTV	**H**igh **D**efinition **T**ele **V**ision
HCF	**H**ybrid **C**oordination **F**unction
HEVC	**H**igh **E**fficiency **V**ideo **C**oding
HVS	**H**uman **V**isual **S**ystem
IBSS	**I**ndependent **B**asic **S**ervice **S**et
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
IFS	**I**nter-**F**rame **S**pace

IGMP	**I**nternet **G**roup **M**anagement **P**rotocol
IP	**I**nternet **P**rotocol
ISO	**I**nternational **O**rganization of **S**andardization
ITU-T	**I**nternational **T**elecommunication **U**nion-**T**elecommunication Standardization Bureau
JCT-VC	**J**oint **C**ollaborative **T**eam on **V**ideo **C**oding
JVT	**J**oint **V**ideo **T**eam
LAN	**L**ocal **A**rea **N**etwork
LSP	**L**inel **S**pectrum **P**airs
LAN	**L**ocal **A**rea **N**etwork
LLC	**L**ogical **L**ink **C**ontrol
LP	**L**inear **P**rediction
MAC	**M**edium **A**ccess **C**ontrol
MANET	**M**obile **A**d-hoc **NET**work
MGS	**M**edium **G**rain **S**calability
MCDT	**M**odified **D**iscrete **C**osine **T**ransform
MOS	**M**ean **O**pinion **S**core
MSE	**M**ean **S**quared **E**rror
MNB	**M**easuring **N**ormalizing **B**locks
MOSFP	**M**ulticast **O**pen **S**hortest **P**ath **F**irst
MSDU	**M**AC **P**rotocol **D**ata **U**nit
NALU	**N**etwork **A**daptation **L**ayer **U**nit
NDP	**N**ull **D**ata **P**acket
NR	**N**o **R**eference
NTSC	**N**ational **T**elevision **S**ystem **C**ommittee
OCF	**O**ptimal **C**oding in the **F**requency Domain
OFDM	**O**rthogonal **F**requency **D**ivision **M**ultiplexing
OSI	**O**pen **S**ystems **I**nterconnection
PAL	**P**hase **A**lternating **L**ine
PC	**P**oint **C**oordinator
PCF	**P**oint **C**oordination **F**unction

PCM	**P**ulse **C**ode **M**odulation
PESQ	**P**erceptual **E**valuation of **S**peech **Q**uality
PIFS	PCF **I**nter-**F**rame **S**pace
PHY	**PHY**isical
PIM	**P**rotocol **I**ndependent **M**ulticast
PESQ	**P**erceptual **E**valuation of **S**peech **Q**uality
PPM	**P**rediction **P**artial **M**atching
PSNR	**P**eak **S**ignal-to-**N**oise **R**atio
PS-Pool	**P**ower **S**ave - Pool
PSQM	**P**erceptual **S**peech **Q**uality **M**easure
QPSK	**Q**uadrature **P**hase **S**hift **K**eying
QoS	**Q**uality of **E**xperience
QoS	**Q**uality of **S**ervice
PHY	**PHY**sical
RGB	**R**ed **G**reen **B**lue
ROPE	Recursive per-pixel end-to-end distortion estimate
RR	**R**educed **R**eference
RTS	**R**equest **T**o **S**end
SDG	**S**ubjective **D**ifference **G**rade
SIFS	**S**hort **I**nter-**F**rame **S**pace
SNR	**S**ignal to **N**oise **R**atio
SSCQE	**S**ingle **S**timulus **C**ontinuous **Q**uality **E**valuation
SSIM	**S**tructural **SIM**ilarity
SVC	**S**calable **V**ideo **C**oding
THD	**T**otal **H**armonic **D**istortion
TXOP	**T**ransmission **OP**portunity
UHD	**U**ltra **H**igh **D**efinition
USRP	**U**niversal **S**oftware **R**adio **P**eripheral
VANET	**V**ehicular **A**d-hoc **N**etwork
VANET	**V**ector **Q**uantization
WLAN	**W**ireless **L**ocal **A**rea **N**etwork

WMA	**W**indows **M**edia **A**udio
WMN	**W**ireless **M**esh **N**etwork

Symbols

MSD	MSD value
PSNR	PSNR value
SSIM	SSIM value
D_k	distortion estimated by EDA associated to the k-th packet
D_q^k	distortion estimated by PESQ ($q = 1$) or SSIM ($q = 2$) associated to the k-th packet
$D_{k,MSE}$	MSE-based distortion associated to the k-th video packet
$D_{k,SSIM}$	SSIM-based distortion associated to the k-th video packet
$D_{i,PESQ}$	PESQ-based distortion associated to the i-th audio packet
$\tilde{D}_l$	distortion estimated by EDA and associated to the frame l
$\tilde{D}_{l,\mathrm{SSIM}}$	SSIM associated to the loss of the l-th video frame
$\tilde{D}_{l,\mathrm{MSE}}$	MSE associated to the loss of the l-th video frame
$D^{'}_{k,\mathrm{SSIM}}$	SSIM associated to the loss of the k-th video packet
$D^{'}_{k,\mathrm{MSE}}$	MSE associated to the loss of the k-th video packet
$\tilde{D}_{l,l'}$	SSIM-based distortion on the frame l' due to the loss of l
$D_{l,l'}$	complementary SSIM-based distortion on the frame l' due to the loss of l
μ	mean value
σ	standard deviation
bv	similarity of the local area luminance
ac	similarity of the local area contrast
ps	similarity of the local patch structure
C_{bv}	adopted constant for bv
C_{ac}	adopted constant for ac

C_{ps}	adopted constant for ps
C_o	adopted constant in the SSIM calculation
Pix	number of signal samples (pixels) in one frame
w	patch area used for the SSIM calculation
$\mathcal{F}_I$	subset of I frames
$\mathcal{F}_P$	subset of P frames
$\mathcal{F}_B$	subset of B frames
M^P	number of frames affected by the loss of a P frame
M^B	number of frames affected by the loss of a B frame
V	number of encoding layers
α	GOP size
T	delay
T_{a_k}	cumulated delay for the k-th packet
T_{e_k}	estimated expiration time for the k-th packet
T_{e_l}	expiration time for the l-th frame
$\tilde{T}_{e_l}$	expiration time for the l-th frame
$\mathcal{T}$	set of estimation times used ad MAC layer
t_S	total time for an exact SSIM calculation
t_A	total time required by the algorithm proposed in Ch.6
t_{DEC}	decoding time
t_{SSIM}	computing time for the evaluation of the SSIM
t_{ALG}	time for evaluating the complementary SSIM-based distortions in Ch.6
t_{d_S}	time required to perform the decoding process (for an exact calculation)
t_{s_S}	time required to evaluate all the SSIM values (for an exact calculation)
t_{d_A}	time required to perform the decoding process (Ch.6)
t_{s_A}	time required to evaluate all the SSIM values (Ch.6)
T_{γ_l}	temporal layer of the frame l
T_{Γ_l}	maximum temporal layer of the frame l
T_S	success time
T_C	collision time
$\bar{T}$	average transmission time

$\mathcal{V}$	video sequence
$\mathcal{V}'$	video sequence distorted
$\mathcal{S}_q^k$	VO ($q = 1$) or VI ($q = 2$) sequence decoded in absence of the packet s_q^k
Λ	payload size
p	collision probability
p_{drop}	drop probability
τ	transmission probability
t_q	complementary of the transmission probability
π	802.11 packet
f_l	l-th video frame
ς	slot time
R	data rate
R_c	control rate
ACK	length Acknowledgment
H	length header
$AIFS$	AIFS value
W	minimum contention window
m'	maximum backoff stage
$\eta_{i,n}$	generic steady-state probability in the Markov chain
Ens	average number of backoff decreases for a packet
Es	average time required for a decrease of the backoff
$\bar{S}$	mean throughput
S	source
D	destination
N	number of nodes
Q	number of queues
En	energy required for the transmission of a whole video sequence
λ	average number of arrivals in the queue
$\lambda_{1,q}$	probability that at least one new packet arrives on the transmission queue during the process of the previous packet
$\lambda_{2,q}$	probability that at least one arrival occurs between two

	consecutive backoff
$m_{q,k}$	retry limit of the k-th packet in the q queue
m_l	retry limit of the l-th frame
$m_{k,MSE}$	retry limit of the k-th packet adopting the MSE-based distortion
$m_{k,SSIM}$	retry limit of the k-th packet adopting the SSIM-based distortion
$\hat{m}_q^k$	retry limit for VO or VI queue in Ch.7
$\bar{l}$	first frame after which the cumulated delay is considered
ζ	parameter adopted for the manipulation of the distortion
$\mathcal{L}_k$	set of frames fully or partially contained in the k-th packet
a_q, b_q, c_q	second order polynomial parameters used in Ch.4
β, β'	degrees of the equations in Ch.7

Dedicated to my family . . .

Part I

Background

Chapter 1

Multimedia over wireless networks

In the last years the fast growth of the Internet has led to the development of many new multimedia applications in the field of entertainment, communication and electronic commerce and they can rely on Local Area Network (LAN) technologies based on Internet Protocol (IP), capable to connect always more networks all over the world. Multimedia communications are characterized by high data rate, a stream oriented highly bursty traffic pattern, low latency requirement and communication multi-point. In particular, the streaming video represents a very big part of the source traffic. On-demand video streaming applications such as Netflix [1] and Hulu [2] make possible to watch a movie or any video with the use of a laptop or a tablet, adopting adaptive bitrate streaming technology aiming to adjust the video and the audio quality to match a customer's real time network condition and broadband connection speed. Furthermore video sharing applications like Youtube [3] allow one with a connection to the Internet to download or upload a video anytime, and introducing also the possibility of streaming a live event. Streaming video is being employed in media, such as film, television, gaming and even in literature and visual arts. An important form of file transfer is the web browsing. Every web page that a user requests, starts a series of file transfers.

The importance of multimedia spread also to private or public areas, like homes, school, campus and airports, where WiFi networks are employed to allow one to have access to the Internet or to be connected with others users present in the same area. In this way one can watch a video while waiting for his flight, or upload a video obtained by his cell phone camera while walking in a city area covered by a public WiFi network or organize a private LAN for gaming, without being wired connected. Nowadays, WiFi has been adopted in roughly 200 million households worldwide and there are more than 750,000 hot-spot globally and over 800 million WiFi-enabled devices are sold every year [4].

All these presented applications are related with the quality of the connection. The Internet and the wireless networks, in fact, are subject to very unpredictable and variable conditions, which depend on many factors. When these conditions are averaged in time, they could not cause serious issues to file transfer, but could cause real problems to real-time or streaming applications. In fact, multimedia is usually delay sensitive, loss tolerant and bandwidth intense. Variations in the network conditions can lead to deep changes in the quality of service (QoS). To cope with these issues, these applications usually provide different levels of quality. For example Netflix allows one to choose the download rate quality of video due the available bandwidth, from a minimum quality level to a 4K Ultra High Definition (UHD) [1]. Similarly, Youtube can stream low, medium and high-quality videos [3]. To reduce the problem of high delay or interrupted communications, the receiver node adopts a buffer, to store the data that have been already received. Thus, if an interruption occurs, the visualization is not affected until the buffer gets empty.

To efficiently transmit multimedia contents through a wireless network, it is important to keep in consideration the variability of the rate, the quality requirement, the resilience to losses and the delay constraints. Due to the importance of these topics, in the last years, many studies have focused their interest in creating techniques and new algorithms capable to improve the quality of the video at end user, exploiting different solutions, capable to satisfy the stringent QoS requirements of the market.

1.1 Unicasting, multicasting and broadcasting

The multimedia communication can be performed with three different methods: unicast, multicast and broadcast. This classifications rely on the relationship between the number of senders and receivers involved in the process. In unicast transmissions, a single device sends a message to a specific receiver. This means that when a sender needs to send a message to multiple receivers, it must send multiple separate messages to single IP addresses, that it must know, corresponding to the receiver nodes. The advantage of such methodology is that a feedback channel can be established between the sender and the receiver, allowing the latter to send information to the sender about the channel conditions or about requirements of the end user, aiming to improve the different transmission aspects.

In multicast transmission instead, multicast groups are defined, and each of them is identified by a special IP address belonging to a particular defined range of addresses. This method uses the Internet Group Management Protocol (IGMP) which allows a host to inform a router connected to itself that one of its applications is asking to connect to a particular multicast group. Other routing protocols such as the Protocol-Independent Multicast (PIM), the Distance-Vector Multicast Routing Protocol (DVMRP) and the Multicast Open Shortest Path First (MOSFP), are implemented in the router to support the multicasting. Furthermore, the IGMP snooping feature enables the switches to know which device belongs to which multicast group, so when a message arrives at the switch, and it is addressed to a specific group, the switch can route the message to the right multicast group. Multicast is more efficient in terms of resource utilization, but at the same time introduces the drawback of making unable a message to be delivered to only a specific receiver.

Finally, the third method is the broadcasting. In this methodology a message carries a special broadcast address and it is sent to all the devices of a network. All the devices

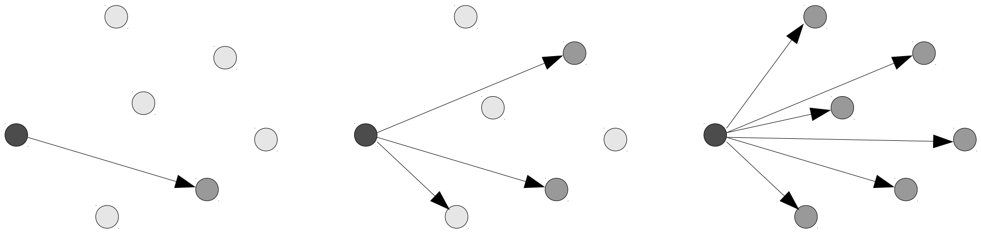

FIGURE 1.1: The multimedia communication methods: (left) unicasting, (center) multicasting, (right) broadcasting.

in that segment of the network will process it. In fig.1.1 it is possible to see an example of the three afore described multicast methods.

1.2 Streaming and downloading

Downloading applications, considered as file transfer such as File Transfer Protocol (FTP) involve the download of a file before it can be consumed by a user. In this scenario, the file is saved to the user's device hard drive, giving the possibility to play it at a later time. After the connection to the source and the download, the file is saved on the hard drive and can be copied on other devices and played any time again. Usually the download process requires to be finished before a user can start to use the file, even if some applications such as iTunes [5] allows one to start to use the file once a part of itself has been downloaded. Usually downloading is a robust way to deliver media, with the drawback of large buffers at the receiver end and possible long time of waiting before being able to use the file requested. An alternative to the downloading is the streaming of media. In this scenario, the streaming applications divide the bit stream into packets, which are sent independently to the destination. Meanwhile, the receiver is capable to use the already received packets for starting the playback. The streaming applications are very sensitive to network quality degradation, like packet loss and jitter, and its requirements of bandwidth depend on the codec and compression rate used [6].

It is understandable that with these applications, the allowed delay is low, because in scenarios like conferences or video on demand applications, the delay at end user must be very low, resulting as a strict constraint. The streaming results more flexible and requires low storage space, but on the other hand results much more complicated and subject to time constraints.

1.3 Real-time, live broadcast and media on demand

Multimedia streaming applications can be organized into three main categories due to their delay tolerance: real-time communications, streaming media on demand and live broadcasting [7,Ch.1]. The first one has very strict delay constraints since involves interactive video and audio applications, where usually more then 200 ms delay at end user would cause an uncomfortable experience. On the opposite side, live broadcasting does not involve interaction in the communications and thus is tolerant to delay up to 20 or 30 seconds. In the middle of the two, there are the streaming on demand applications. Usually they can accept some seconds of delay since they involve only limited interactivity, like changing a channel. Real time communications usually refer to real-time communications while the streaming media on demand and live broadcasting are usually considered as streaming [7,Ch.1]. The differences on delay acceptance lead these three categories to have very different structures and characteristics, most of all on the error recovery aspect. For this reason an important aspect is to resolve this underlying problem, trying to perform a prioritization in the transmission of multimedia contents.

1.4 Distortion calculation for Quality of Service (QoS) purposes

Wireless networks are heterogeneous in reliability, bandwidth and transmission characteristics. A packet can be delayed due to many issues, like transmission and retransmissions, queueing, channel access delays, or even discarder or lost in the transmission. Fortunately, audio and video contents are usually elastic to loss and the end user overall quality may tolerate its effects. Multimedia transmission applications though, including most demanding low delay and streaming applications, impose strict QoS requirements, expected from the new generation of networks. Furthermore, unlike conventional communication applications, the multimedia applications cannot be simply evaluated in terms of throughput or bit rate, since the delay is an important aspect and the bits are not ”created equal” [7,Ch.1]. To achieve a better QoS many studies have been portrayed aiming to create a prioritization of the audio and video contents. The knowledge of which parts of an audio or video stream are more important for the decoding process, due to the distortion introduced by the loss of packets of frames, may lead one to provide better protection or prioritization to these parts, with an improvement of the QoS at the end user. In the next subsections, we provide a fast overview of some of the most important techniques for evaluating the distortion in video and audio transmission, useful for performing a prioritization on the audio and video flows with QoS purposes. The knowledge of the distortion introduced by each packet or frame of the multimedia stream may allow one to protect more some parts in accordance with the distortion introduced. For this reason, in the next subsections, we don’t differentiate the concept of distortion evaluation and prioritization, since the first may lead to the second very easily.

To evaluate the distortion on a sequence, audio or video, in literature one may find several audio or video quality assessments, capable to estimate the distortion on a sequence when compared to the same one, without any distortion. In the next subsections we provide a quick overview of the most diffused video quality assessments, for both audio and video contents.

1.4.1 Audio quality assessments

Assessing the perceptual quality of audio signal represents an important tool in the study and evaluation of multimedia transmission. During the past years many audio quality assessments has been proposed. In the early 1990s the standard way to measure the audio quality was to conduct a subjective test, which provides the mean opinion score (MOS) of the quality of each condition under test. In [8] the authors classify the quality measurements in two big groups. The subjective quality measurements and the intrusive objective ones.

Subjective quality measurements consists in adopting original signals, and processing them under different conditions. The so corrupted signals are then played randomly for listener which give their quality opinion about the quality of the audio signal. In telephony the most used approach is described by the ITU-T P.800 recommendation [9]. Speeches from male and female talkers are processed in different conditions and the score provided by listeners is presented using the five points absolute category rating (ACR) listening quality (LQ) scale, in Tab. 1.1.

Listening quality
1.Bad
2.Poor
3.Fair
4.Good
5.Excellent

TABLE 1.1: Five-point ACR LQ scale.

Other subjective tests can involve the degradation category rating (DCR) or the comparison category rating (CCR) presented in [9]. Another method for categorizing distortion is presented in [10], where the authors propose the diagnostic acceptability measure (DAM). The ITU-R BS.1116 [11] defines how to test the quality of an audio sequence. Subjects listen different degraded signals and provide a vote in accordance with the subjective difference grade (SDG). Furthermore, other quality assessments for the evaluation of the quality of the audio in the presence of interpersonal communications needs to consider the listening quality, the talking quality and the interaction quality [12] of the audio over the phone. Finally in [13] the ITU-T P.800.1 defines the accuracy of the different subjective tests.

The intrusive objective measures, instead, perform a comparison between the original audio process and the one processed through the conditions under test. Initially developed objective methods for the evaluation of the quality of an audio process were based on principles like the total harmonic distortion (THD) and the signal to noise ratio (SNR), without accounting for psychoacoustic features of the human audiotory system [14]. In 1996 the International Telecommunication Union (ITU-T) adopted the perceptual speech quality measure (PSQM) [15] in the reccomandation P.861 [16], which has been extended in [17] by adopting a solution based on measuring normalizing blocks (MNB). MNB, basing on the sensitivity of human's hear to the distribution of the distortion, converts speech signals into an approximated loudness domain using frequency warping and logarithmic scaling techniques, and then generating an approximated loudness vector for each audio frame. In 1999 ITU-T introduced as recommendation the perceptual evaluation of quality measure (PEQM) [18], obtained through the combination of different quality assessments. An evolution of these assessments is represented by the perceptual analysis measurement system (PAMS) [19, 20], that approaches the analysis from an end to end behavior point of view. Finally in [16, 21] the perceptual evaluation of speech quality (PESQ) model has been presented, whose scores are on an arbitrary scale that is not representative of typical subjective tests [8].

1.4.2 Video quality assessments

The goal of objective image and video quality assessments is to predict perceived image and video quality automatically. The video quality assessments can be used for monitoring the quality for control systems, as benchmark in video processing systems and algorithms [22,Ch.41]. We can differentiate the evaluation of video quality in two main categories. Subjective and objective evaluation.

Subjective evaluation results to be complicate due to many aspects of viewing conditions and human psychology. The most important methodologies are the single stimulus continuous quality evaluation (SSCQE) and the double stimulus continuous quality scale (DSCQS). In the first methodology the viewers provide their impression about the quality of the video in a continuate way and track the changes in time with a slider over a quality scale like the one in Tab. 1.1. In the DSCQS methodology instead the original video and its version affected by distortion are shown to the viewers one after the other one, in time intervals of few seconds. The quality of the video is obtained as difference between the scores of the two video samples. These two techniques are the most relevant in the subjective approaches and for this reason are adopted as standard in the ITU-R BT.500-10 [23]. Furthermore the results provided by these two tests can be averaged for obtaining the MOS.

Objective quality image metrics can be classified into three major categories: full reference (FR), reduced reference (RR) and no reference (NR). NR metrics base on the philosophy that all images are perfect unless they have been distorted during acquisition, processing or reproduction stage [22,Ch.41]. In [24] the authors present a statistical model for natural images working in the wavelet domain for evaluating the quality of JPEG2000 images. For the nature itself of this type of methodologies, for being effective the NR metrics need to be specifically created for a specific distortion. To avoide this problem, the RR metrics can be adopted. These metrics don't assume to have a full reference image, but they assume to have a partial reference image sent on an optional channel used with this purpose. In [25] the authors propose a semi-RR model capable to evaluate the distortion that is introduced during the compression and the transmission over the channel for estimating the perceptual degradation after the reception of the video sequence. Furthermore in [26] an RR method is presented. It is based on the extraction of localized temporal and spatial activity and provides three different comparison metrics trained on data obtained from human subjects. The most effective metrics for the evaluation of the quality of an image or a video are however the FR ones. The most widely used RR metric is the Mean-Squared Error (MSE) and the peak signal-to-noise ratio (PSNR) [27]. These two quality metrics are widely used because they have clear physical meaning and are computationally cheap to evaluate, but have been widely criticized for the lack of capacity in evaluating perceived quality [28–30]. To better evaluate the perceived quality of an image, the research community has focused on finding algorithms basing on the human visual system (HVS). In [29,Ch.14] the author presents a technique to evaluate a probability-of-detection map between two signals: a reference and a distorted one. A similar approach is taken in [31,Ch.10], where an estimation of the detection probability of the differences between the original and the distorted video is presented. An approach through the discrete cosine transform (DCT) is presented in [32]. The spectrum of the signal is partitioned into 64 bands and a visibility threshold is calculated for each band and corrected with texture and luminance masking. These thresholds are then used to perform a pooling operations between sub-bands. These metrics have the characteristic of providing scalar values. Another approach is considered in [33] where the focus is placed on targeting certain types of distortions measuring the image quality by filters that model single-channel scenarios. Furthermore a modified version of the PSNR has been presented in [34]. Finally, a completely different approach is introduced in [27, 35], where the structural similarity (SSIM) index is presented. The SSIM differently from the previous methods, does not consider error measures, but structural distortion measures. The image degradation is considered as the perceived structural information loss [22,Ch.41].

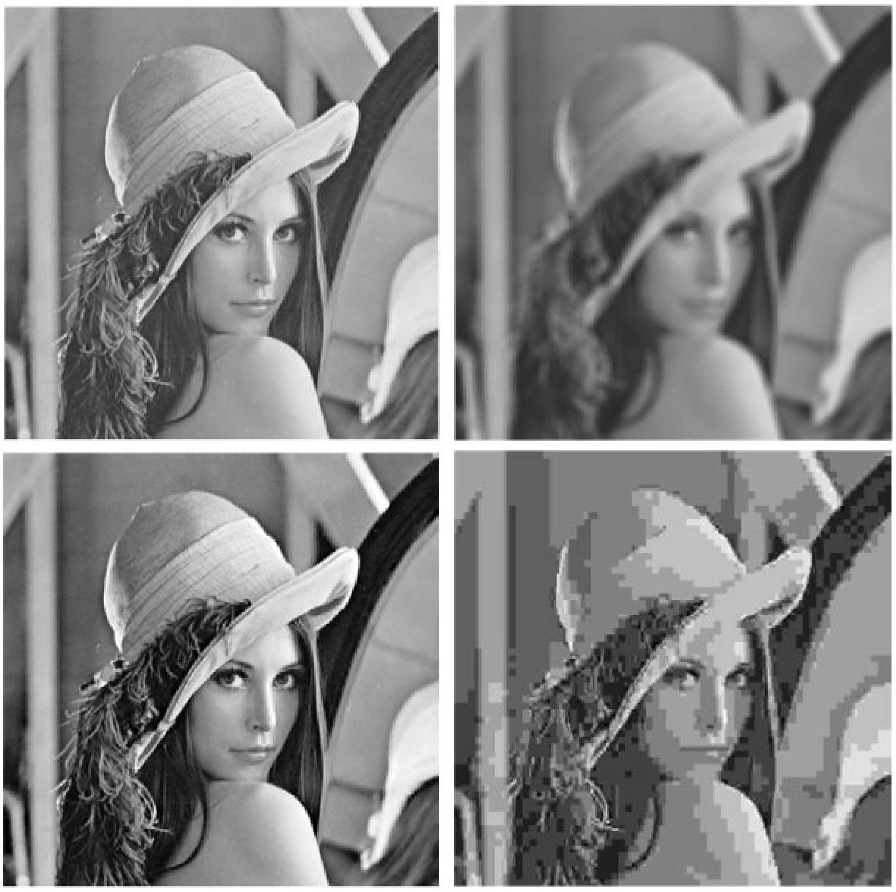

FIGURE 1.2: Images with 4 different type of distortions. Top-left: MSE=225, SSIM=0.99; top-right: MSE=225, SSIM=0.35; bottom-left: MSE=225, SSIM=0.94; bottom-right: MSE=225, SSIM=0.29 [22,Ch.41].

The different capability of evaluating the distortion of an image, can be observed in Fig. 1.2. More precisely, for different types of distortions affecting the same image, the widely used MSE provides the same value of distortion. The SSIM instead, is capable to highlight the different impacts on the images, showing to be drastically more effective in perceiving the distortion as the humans eye sees it.

1.5 Distortion evaluation for Quality of Service (QoS) purposes

Adopting the previous described audio and video quality assessments, one may develop techniques capable to approximate the distortion on a multimedia sequence due to the structure or the properties of the content, or the characteristics of the transmission. In scientific literature several algorithm for the prioritization of the multimedia content are proposed, basing on the same idea that the information more important for a decoding process need to have higher priority in the transmission.

1.5.1 Audio prioritization

Priority techniques to select the sounds of an audio stream due to their importance in the decoding process may be a good way to perform a prioritization of an audio stream that has to be transmitted. In the scientific literature, there are some studies focused on this issue, aiming to prioritize audio signals according to an importance metric. In [36] the authors proposed a model for rendering complex virtual environments in which sound sources are first sorted by an importance metric, more precisely the loudness level of the sound signal. Pre-computers descriptors of the audio signals are used for re-evaluating the importance of each sound source, accounting for the location in relation with the one of the listener. In [37] the authors propose an approach for spatialized audio where the samples are generated in accordance with a perceptual metric aiming to save computing time, while in [38] the authors propose a multi-resolution approach to 3D audio rendering, adopting an importance sampling strategy to select a sub-set of the audio sources to render.

Another way to classify the audio signal is based on the distortion introduced by the loss of some packets, and in [39] the authors evaluate the perceptual quality of audio signals exposed to packet loss, adopting both very simple and very complex error concealment schemes. The results show statistically significant differences between different packet loss rates, error concealment techniques and audio files.

1.5.2 Video prioritization

To achieve better QoS of video contents at the end user, many studies have been performed and cross-layer strategies have been proposed at different layers of the protocol stack [40–42]. Aiming to improve the quality of the multimedia contents at the end user, it is very important to adopt prioritization schemes capable to provide higher priority to the audio or video data that result more important in the decoding process and whose loss would imply a higher distortion with consequent lower QoS. For the video data, [43] defines the lightweight prioritization schemes. These methods differentiate the key-frames from the non-key ones or the intra-coded from the inter-coded ones [44–46], and are characterized by computationally cheap mechanisms. In [43] the authors use interchangeably the terms prioritization methods and distortion estimation, considering the second one as a tool required to perform a proritization of the data. These solutions require low computational cost and are of easy implementation, but with the drawback to be too much simplistic, thus not capable to provide a good prioritization to the traffic. Another further criterion may focus on the layers of the scalable coding as proposed in [46]. The proposal in [43], instead, classifies as sophisticated prioritization schemes the ones based on pixel or block methods that provide a better evaluation of the distortion with the drawback of higher computational cost. These solutions may incorporate algorithms for taking into account propagation errors concealment techniques, channel modeling techniques and distortion estimation models. In [47] the authors propose a block-based method, introducing the idea of a distortion matrix, which allows one to evaluate the distortion caused by dropping a frame in a GOP of a video sequence, and adopting a frame copy technique in order to replace the frame lost. Furthermore, the additional distortion for a specific dropping pattern results to be the sum of the distortion on each frame of the pictures that have been concealed. Each of the dropping scenarios considers the drop of a I, P or B frame, assuming to drop all the frames of the sequence depending from another frame whose absence would cause an interruption of the decoding process due to missing references. The distortion matrix though requires a high over-head to transmit the elements of the matrix and it does not take into account the capacity of the decoders to use the frames previously received to mitigate the distortion caused by a missing frame. Better solutions are represented by pixel-based methods. The decay of error propagation in time is instead considered in [48]. Assuming that the loss pattern probabilities for that channel are computable, the authors analyze the impact of the quality of the video due to Markov-model burst packet losses accounting for the temporal dependencies induced by both the motion-compensated coding scheme and the Markov-model channel losses. Based on the study of the decaying behavior of the error propagation, an algorithm is presented aiming to perform the MSE estimation with low complexity. This solution requires content-aware transportations aiming to avoid interruptions in the visualization of the video sequence by the end user. Another approach to provide QoS consists in analyzing the decoder behavior. In [49, 50] the authors propose another method capable to estimate the channel distortion through the simulation of the decoder behavior when a single packet is lost, requiring though high computational cost like in the previous solutions. The impact of multi packet losses is treated as additive because an exhaustive analysis would be computationally untreatable. Aiming to provide an algorithm requiring computational cost, but at the same time providing a good estimation of the distortion introduced by the channel, for QoS purposes, in [51] the authors propose a model to shape the decay of distortion over the time, due to channel loss. In order to obtain this, the authors use the MSE evaluated between the lost frame and the previous one, thus adopting a frame copy technique at the receiver. This solution assumes the loss of an entire frame but requires low computational cost, thus being suitable for streaming scenarios where low delay is required.

A more sophisticated approach is a pixel-based approach. In [52] the authors propose a technique called recursive per-pixel end-to-end distortion estimate (ROPE), which recursively calculates the expected distortion at pixel level evaluating the coding mode per macroblock and the error propagation on them. Additionally, adopting cross-correlation approximations, this algorithm can perform pixel-filtering/averaging and subpixel motion compensation operations. This technique though, introduces the drawback of a high computational cost, making this solution not suitable for scenarios that require low calculation time. Moreover, it only accounts for isolated packet losses as pointed out in [53], where the authors furthermore propose a ROPE-based first-order Taylor expansion to model the expected distortion, with the assumption that one packet corresponds to one frame and that I-frames are never lost.

In conclusion existing approaches can be roughly classified as computationally lightweight and intuitive, or sophisticated methods. Furthermore, the latter can be categorized in block and pixel based techniques.

1.6 Modeling the Medium Access Control (MAC) layer for QoS purposes

Wireless networks are subject to bit errors caused by the not reliability of the medium, that can introduce degradation on the transmitted signal. The errors introduced and not detected and corrected at the physical (PHY) layer, propagate to the Medium Access Control (MAC) layer, where usually are dropped through the adoption of a checksum. Wireless standards usually adapt the PHY layer and maintain unchanged the MAC layer [54, 55]. An accurate model of the MAC layer can render important insight into the underlying characteristics of an impairment [7,Ch.14]. This insight allows one not only to design and evaluate the performance at this layer, but also to tune the different parameters for QoS purposes. Protocols for congestion controls, reliable routing, rate adaptation and error control can be implemented for improving the Quality of Experience (QoE) in the multimedia transmission. Such MAC protocols can, for instance, tune some parameters in accordance with particular needs, aiming to differentiate the QoS or to improve the overall perceived quality of the multimedia contents. The manipulation of one of these MAC layer parameters: the maximum number of retransmissions, is one of the topic discussed in the second part of this thesis.

Chapter 2

Audio and video coding

The multimedia transmission represents a very big portion of nowadays transmissions. Content delivery networks have prevailed as a dominant method to deliver audio/video. Globally, 72 percent of all Internet video traffic will cross content delivery networks by 2019, up from 57 percent in 2014 [56]. The sum of all forms of video (TV, video on demand [VoD], Internet, and P2P) will be in the range of 80 to 90 percent of global consumer traffic by 2019 [56]. Much of the video streamed or downloaded through the Internet consists of TV episodes, clips or movies. A studio performed by Cisco highlighted how the load of multimedia contents delivery is in rapid growth and a quick overview is depicted in Tab. 2.1.

In the past decade, the rapid growth of this kind of data brought the need to find a way to send huge amount of multimedia. To sustain this high request of data, multimedia contents needs to be compressed. Compression bases on reducing the amount of data, taking into account the redundancy in it, which can be spatial, temporal, spectral or psycho-visual. A first classification in coding can be performed defining lossless and lossy compression. When the original data is perfectly equal to the compressed one, we talk about lossless compression. Lossless compression is essential for example in

Consumer Internet Video 2015–2019			
	2015	2017	2019
By Network per Month			
Fixed	25,452	43,226	74,319
Mobile	2,014	5,842	14,999
By Region per Month			
Asia Pacific	9,366	16,039	28,469
North America	8,207	14,009	23,794
Western Europe	4,422	7,696	13,766
Central and Eastern Europe	1,956	4,398	9,577
Latin America	2,600	4,483	7,808
Middle East and Africa	915	2,443	5,905
Total per Month			
Consumer Internet video	27,466	49,068	89,319

TABLE 2.1: Global consumer Internet video growth in Peta Bytes [PB].

text compression, where the text needs to be exactly the same compared to the original version. One example is the Lempel-Ziv-Welch compressor, where sequences of characters in the original text are replaced by codes that are dynamically determined. Another one is the Prediction by Partial Matching (PPM). This is an statistical adaptive data compression technique which bases on context modeling and prediction. PPM uses previous symbols in the uncompressed symbol stream to make a prediction of the next symbol. The very well known standard Joint Photographic Experts Group (JPEG) [57], used to compress digital images, provides versions that are lossless, to be used when no information needs to be lost in the process. But lossless compression provides still big amount of data load. For example, in one of the standards for High Definition TeleVision (HDTV), each frame is formed by 1920 pixels per line, and 1080 lines vertically, and each second 30 frames needs to be visualized. When 8 bits are used for each of the three main colors, the total data rate would result roughly between 1 and 2 Gb/sec. Very effective compression techniques need to be adopted even at the cost of some information loss. In the video context, very well known compression standards are the H.264 video compression standard and the Moving Picture Experts Group (MPEG) standard, which is used also for audio data compression. In this chapter the focus is placed on the lossy compression standards, that are the ones mainly used in multimedia transmission, since more effective in reducing the size of data. Furthermore, a brief overview of the most common audio and video coding standards is presented, focusing mainly on the audio coding standard G.729 [58] and the video coding standard H.264 [59].

2.1 Speech and audio coding

For obtaining a digital representation of an audio signal, many aspects need to be considered. The main target is to obtain a trade off between size of the data and perceived quality. The aim in fact, consists in compressing as much as possible the total amount of audio data, without implying a decrease in the perceived quality of the audio. Thus the perceived quality of the audio signal, is as closer as possible to the original uncompressed signal.

Some of the most important factors that one needs to take into consideration in the design of an audio coder, are the fidelity, the data rate, the complexity and the delay [60]. Fidelity describes how perceptually equivalent the signal provided by a codec is compared to the original input signal. The quality depends on the application and it could range from telephony quality to CD one or more. Higher quality implies higher

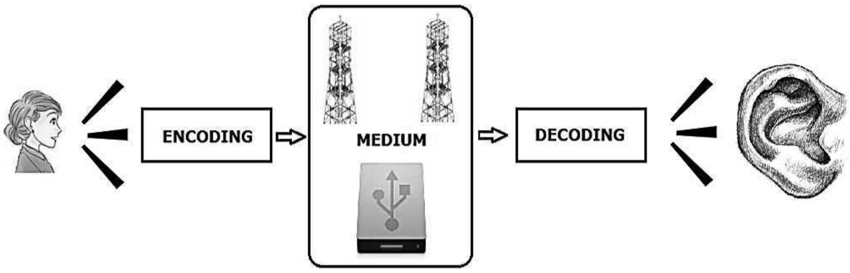

FIGURE 2.1: Digital audio coding chain.

amount of data. The data rate of the audio coding system is connected to the throughput, the storage, and the bandwidth capacity of the available system. Higher data rates typically require higher costs in transmission and storage of the digital audio signals [60]. Another important role is played by the complexity. In fact a system as simple as possible is preferred, aiming to face low cost in the production of consumer devices. Furthermore, low complexity is required most of all on the decoder side, trying to move all the complexity at the encoder side. Thus improvements in the decoding process can be performed without requiring updates or changes in the decoding devices. Another important factor in the design of audio coders include coding delay. For example, in systems working in real time, or streaming, the delay introduced by the decoding process needs to be very low, in accordance with the considered scenario. In general, the target of every audio coding system is to provide high fidelity with low data rates, while maintaining the complexity of the system as low as possible [60]. Another important aspect to keep into consideration during the design, is represented by the errors. In fact, sampling errors, quantizations errors and storage or transmission errors can happen, decreasing the quality of the audio coding process.

The simplest audio coder is based on Pulse Code Modulation (PCM). In PCM the input signal is fed to a sampler with a sampling frequency two times the maximum frequency of the original signal and then to a quantizer which compares the input samples with fixed levels and assigns any one of the samples to a level, for obtaining the minimum error, called quantization error. Finally each level is then binary encoded. In ”code excited linear prediction” (CELP) speech coders, the error signal is mapped onto the best matching of a sequence of pre-defined error signals. More complex and efficient coding standards are used nowadays and a quick overview of the most used is presented through this chapter. Firstly a general overview on the most well known speech coding is presented followed by an overview on the most used audio codecs.

2.1.1 Speech coder: G.711

The speech G.711 is an ITU-T standard for audio speech coding. Mainly used in telephony, the standard was released in 1972. The nominal value recommended for the sampling rate is 8000 samples per second. G.711 defines two main algorithms, the μ-law algorithm [61] and the A-law algorithm [61], which are both logarithmic.
The μ-law algorithm is mainly used in 8-bit PCM digital telecommunication systems in North America and Japan. The μ-law is described as:

$$F_\mu(x) = \text{sign}(x)\ \frac{\log(1+\mu\,|x|)}{\log(1+\mu)}\ , \quad -1 \leqslant |x| \leqslant 1 \tag{2.1}$$

where μ is a parameter equal to 255 (8 bits) in North American and Japanese standards. This algorithm accounts for the characteristic of the human perception of audio signals. More precisely, the perceived acoustic intensity level or loudness by humans, is logarithmic and thus, the algorithm compresses the signal using a logarithmic-response operational amplifier based on the Weber-Fechner law. It effectively reduces the dynamic range of the signal, thus increasing the coding efficiency, biasing the signal with the aim to obtain a signal-to-distortion ratio higher than the one obtainable adopting a linear encoding. The A-law instead, is described as:

$$F_A(x) = \text{sign}(x) \begin{cases} \frac{A|x|}{1+\log(A)}\ , & |x| < \frac{1}{A} \\ \frac{1+\log(A|x|)}{1+\log(A)}\ , & \frac{1}{A} \leqslant |x| \leqslant 1 \end{cases} \tag{2.2}$$

where A is a compression parameter. The tolerance on that rate should be ± 50 parts per million (ppm). The number of quantized values results from the encoding law, and digital paths between countries which have adopted different encoding laws, should carry signals encoded in accordance with the A-law. Where both countries have adopted the same law, that law should be used on digital paths between them. Any necessary conversion will be done by the countries using the m-law. The rules for conversion are given in some tables provided in the standard [61]. About the encoding, in telephony usually are used 256 intervals of quantization for the representation of all the possible sample values and thus 8 bits are used to represent each interval.

2.1.2 Speech coder: G.729

Recommendation ITU-T G.729 [58] contains the description of the conjugate-structure algebraic-code-excited linear prediction (CS-ACELP) algorithm for the coding of speech signals. In its basic mode, the G.729 coder consists of a mono-rate speech coder at 8 kbit/s using fixed-point arithmetic operations [58]. The standard provides also some annexes to extend its functionality for providing a reduced complexity version, defining a source-controlled rate operation, providing multi-rate operation, rate-switching mechanisms and further functionalities to extend the original encoder. The CS-ACELP coder bases on the CELP coding model and operates on speech frames of 10 ms, corresponding to 80 samples at a sampling rate of 8000 samples per second. Every frame of 10 ms is analyzed and the parameters of the CELP model are extracted, encoded and transmitted. At the decoder side, these parameters are used to retrieve the excitation and synthesis filter parameters. The speech is reconstructed by filtering this excitation through the short-term synthesis filter, as is shown in Fig. 2.2 [58].

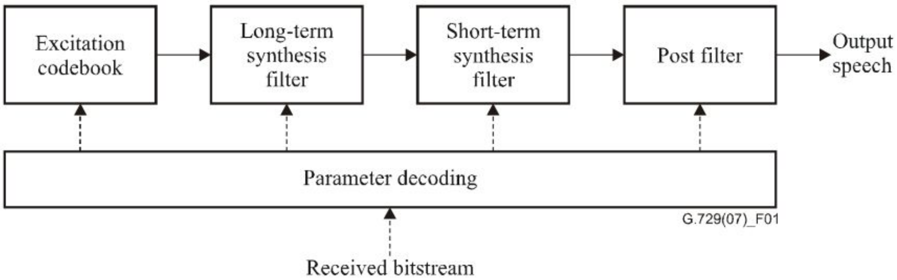

FIGURE 2.2: Block diagram of conceptual CELP synthesis model [58].

The short-term synthesis filter bases its functions on a 10th order linear prediction (LP) filter and the pitch synthesis filter is implemented using the adaptive-codebook approach. The short-term synthesis filter is further enhanced by a post-filter, after the computation of the reconstructed speech [58]. The encoding principle is shown in Fig. 2.3 [58]. The input signal, before being scaled in the preprocessing block is high-pass filtered and then used as input signal for all the performed analysis. Every 10 seconds an LP analysis is done aiming to compute LP filter coefficients which are converted to line spectrum pairs (LSPs) and quantized using predictive two-stage vector quantization (VQ) with 18 bits.

The excitation signal is chosen by using an analysis-by-synthesis search procedure where the error between the original and reconstructed speech is minimized. This operation is performed filtering the error signal with a perceptual weighting filter, whose coefficients are derived from the unquantized LP filter. The minimization procedure is performed in accordance with a perceptually weighted distortion measure. An open-loop pitch delay is estimated for every 10 ms interval accounting for the perceptually weighted speech signal. For each subframe the following operations are repeated. The target signal is computed by filtering the LP residual through a weighted synthesis filter and the initial states of these filters are updated by filtering the error between LP residual and excitation.

The impulse response of the weighted synthesis filter is then computed and closed-loop pitch analysis is performed followed by the encoding of pitch delay, with 8 bits in the first subframe and with 5 bits in the second subframe. The target signal is updated by subtracting the filtered adaptive-codebook contribution, and the new target signal is used in the fixed-codebook search to find the optimum excitation [58]. An algebraic codebook with 17 bits is used for the fixed-codebook excitation. The gains of the adaptive and fixed-codebook contributions are vector quantized with 7 bits and finally the filter memories are updated using the determined excitation signal [58]. The decoder

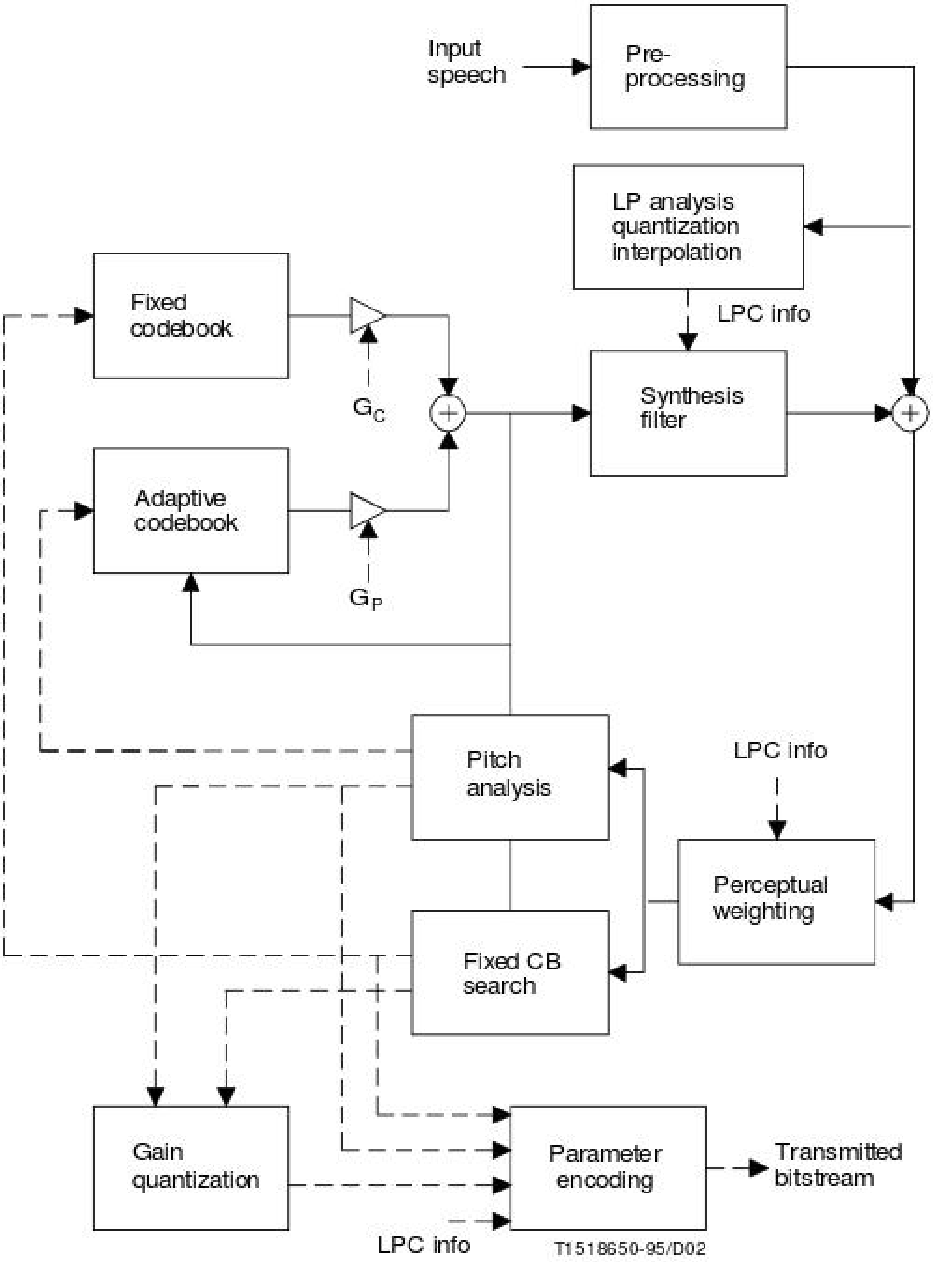

FIGURE 2.3: Principle of the CS-ACELP encoder [58].

principle is shown in Fig. 2.3. The parameter's indices are firstly extracted from the bit stream received and they are decoded to obtain the coder parameters corresponding to each 10 ms speech frame. These parameters represent the LSP coefficients, the two fractional pitch delays, the two fixed-codebook vectors, and the two sets of adaptive and fixed-codebook gains [58]. The LSP coefficients are interpolated and then converted to

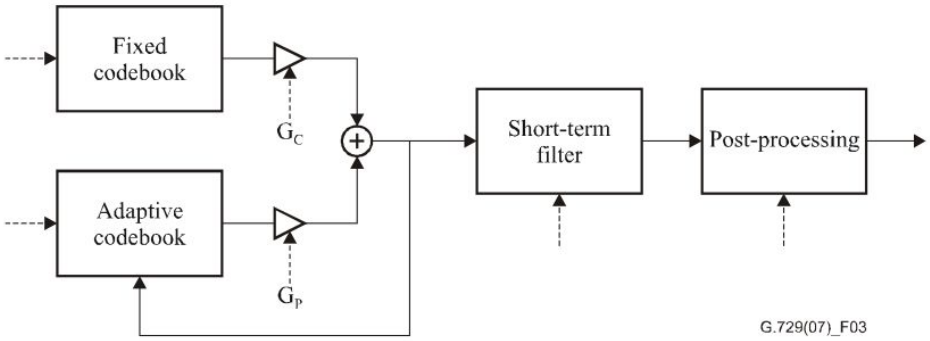

FIGURE 2.4: Principle of the CS-ACELP decoder [58].

LP filter coefficients for each subframe. Then, for each 5 ms subframe, the excitation is evaluated by adding the adaptive and fixed-codebook vectors scaled by their respective gains. Furthermore the speech is rebuilted by filtering the excitation through the LP synthesis filter and finally, the reconstructed speech signal is passed through a post-processing stage [58].

2.1.3 Moving Picture Experts Group (MPEG) layer I, II and III audio

Moving Picture Experts Group within the International Organization of Standardization (ISO) introduced in 1991 the standard MPEG-1 (MPEG layer I). MPEG combines features of coding algorithms such as Adaptive Spectral Perceptual Entropy Coding (ASPEC) [62] and MUSICAM [63] coding algorithms. MPEG-1 offers a subjective reproduction quality that is equivalent to CD quality (16-bit PCM) and the basic structure follows that one of perception-based coders [64]. Adopting an analysis filterbank, the audio signal is converted into its spectral components, which are then quantized and coded with the aim to maintain the quantization noise below the masking threshold. In the decoder the synthesis filterbank reconstructs a block formed by 32 audio output samples obtained from the demultiplied bitstream [64]. The standard describes three

layers of increasing complexity, delay and subjective performances and an MPEG-1 decoder is capable to decode a compressed audio bitstream coded with any of these three layers. The MPEG-1/Audio is capable to support 4 modes: mono, stereo, dual with two separate channels and joint stereo. An adaptive bit-allocation algorithm is controlled by a psychoacoustic model, which is adapted only at the encoder, aiming to reduce the complexity of the decoder. The standard defines the decoder and the structure of the bitstream, leaving space for improving the encoder.
Layers I and II of the MPEG standard are similar to each other also if the second one can reach slightly better performance due to the reduction of the overall scalefactor information. The number of quantization levels for the spectral components are obtained from a dynamic bit-allocation rule controlled by a psychoacoustic model [64]. The decoding process is straightforward and the subband sequences are reconstructed considering blocks of 12 subband samples and accounting for the bit-allocation information and the scalefactors [64]. Each MPEG-1 and 2 frame has a header containing bits about informations on the system, bits for synchronizations and cyclic redundancy check code bits. After the header a second part contains informations about bit allocation and scalefactor used in the encoding process (in MPEG-2 also scalefactor select information). The main part instead carries 32x12 sub-band samples or 32x36 ones, respectively in the first or second layer. Furthermore, the standard defines a multiplexed structure of audio and video data, in one whole stream. The MPEG-1 frame, which can be of variable size, is broken in packets. 188 bytes packets with 4 bytes for header are followed by 184 bytes of payload. These audio packets are then inserted in the stream, together with video and data packets 2.5.

MPEG-2 includes in the audio two multichannel audio-coding standards with one of them being forward compatible with MPEG-1/Audio, which means that an MPEG-2 audio coder is capable to properly decode MPEG-1 mono or stereophonic signals. Furthermore, it is also backward compatible with the MPEG-1, which means that an

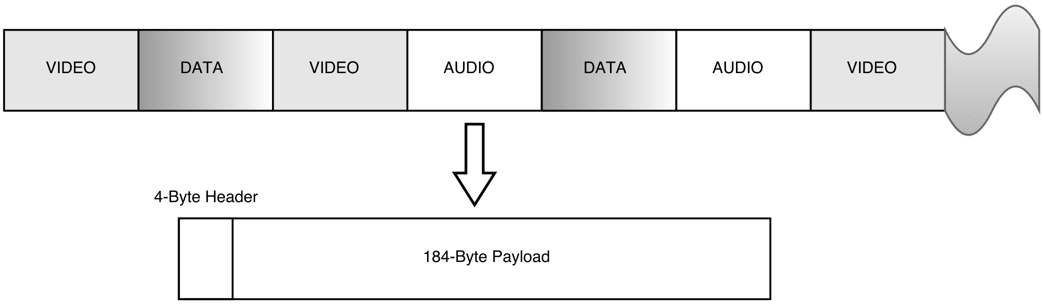

FIGURE 2.5: MPEG-1 frame structure.

MPEG-1 decoder can decode a basic 2/0 stereo signal from an MPEG-2 multichannel bit stream [64]. In Fig. 2.6, it is shown the structure of an MPEG-2 audio frame compatible with MPEG-1 format. The backward compatibility is reached through the adoption of compatibility matrices and transmitting two compatible stereo signals in the MPEG-1 format over the channel.

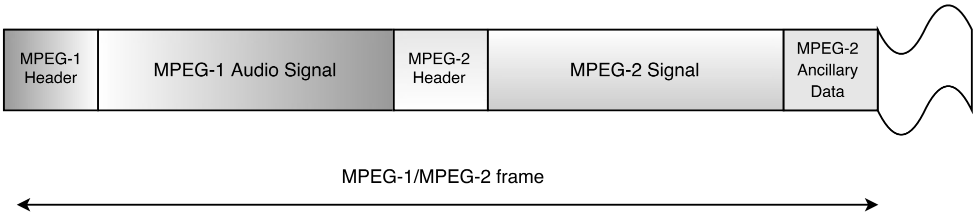

FIGURE 2.6: MPEG-2 frame structure.

MPEG-1/2 Layer III, known also as MPEG-3 or MP3, is the most complex, but capable to produce the highest compression between the three layers. This layer is a much more refined version derived from APSEC and Optimal Coding in the Frequency domain (OCF). In Fig. 2.7, it is shown the structure of the encoder and decoder of the Layer III [65]. MPEG-3 compensates for some filter-bank deficiencies by processing the filter outputs with a Modified Discrete Cosine Transform (MDCT) [65].

The MDCTs subdivide the subband outputs in frequency to provide better spectrum resolution. Then the encoder can partially cancel some aliasing caused by the polyphase

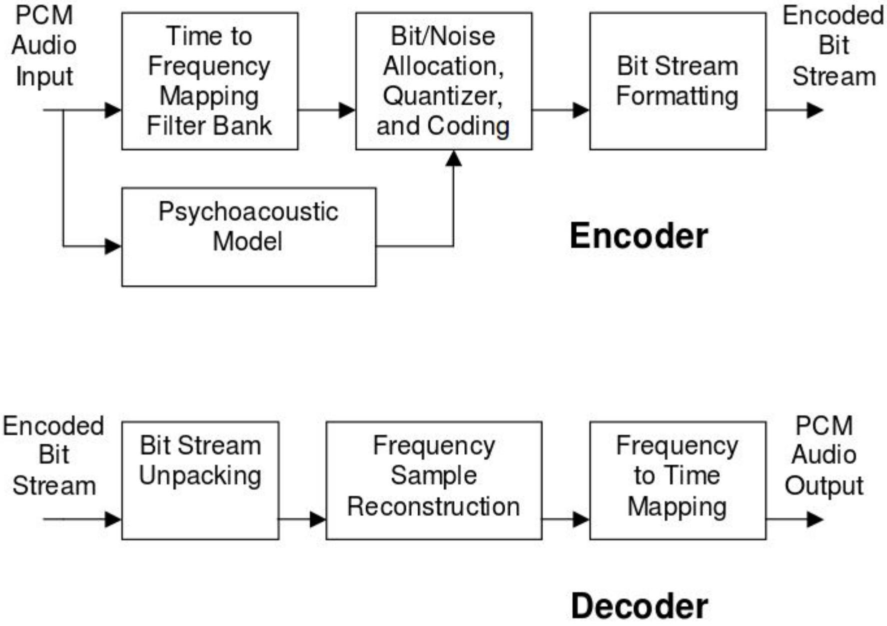

FIGURE 2.7: MPEG-3 encoder and decoder structure [65].

filter, cancellation that needs to be undone at the decoder. Two different MDCT length can be used. Layer III incorporates several measures to reduce pre-echo. Furthermore the layer III introduces some aspects such as alias reduction, nonuniform quantization, scale factor bands, entropy coding of data values and the use of a bit reservoir [65].

2.1.4 Vorbis

Vorbis is an open source software project [66] by the Xiph.Org Foundation. The project developers have created an audio coding format for lossy audio compression which is mostly used in conjunction with the Ogg container format [67], created too by the Xiph.Org Foundation. The Vorbis audio CODEC proposes a channel coupling mechanism designed with the aim to reduce the effective bitrate. Interchannel redundancy and audio information considered inaudible or undesirable are eliminated. Vorbis has two mechanisms that may be used alone or in conjunction to implement channel coupling. The first is channel interleaving via residue backend type 2, which pre-interleaves a

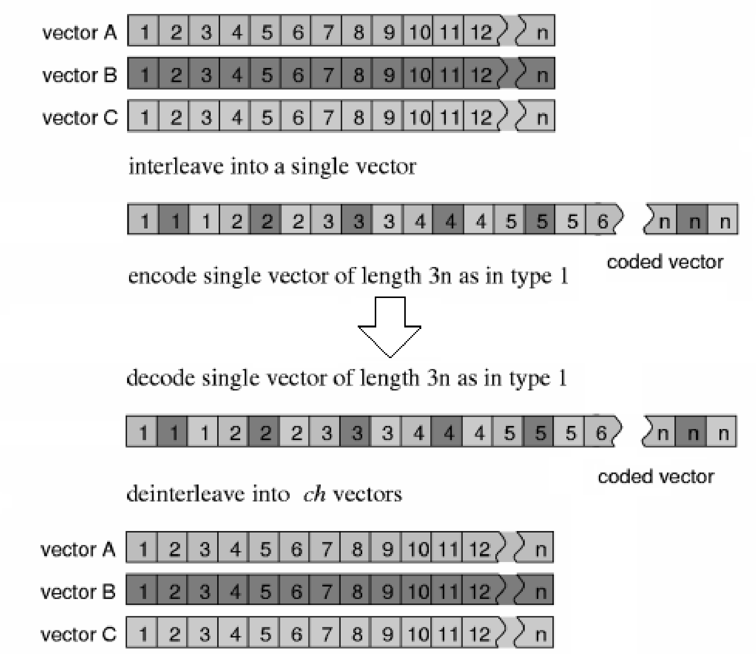
vector A
vector B
vector C
1 2 3 4 5 6 7 8 9 10 11 12 n
interleave into a single vector
1 1 1 2 2 2 3 3 3 4 4 4 5 5 5 6 n n n
coded vector
encode single vector of length 3n as in type 1
decode single vector of length 3n as in type 1
coded vector
deinterleave into ch vectors
vector A
vector B
vector C

FIGURE 2.8: Channel interleaving in Vorbis.

number of input vectors (A and B in stereo mode) into a single output vector before entropy encoding (Fig. 2.8). Thus each vector consists of matching magnitude and angle values [66]. The second mechanism is the square polar mapping, which accounts for the correlation between left and right audio channels which are normalized in energy across the spectrum to maximize correlation before coupling [66].

These two general mechanisms are particularly well suited to coupling, due to the structure of Vorbis encoding. Using both, one may implement both totally lossless stereo image coupling as well as various lossy models that seek to eliminate inaudible or unimportant aspects of the stereo image, in order to enhance bit rate [66]. Vorbis provides different stereo models, such as dual, lossless and phase stereo. Furthermore, raw packets are grouped and encoded into contiguous pages of structured bitstream data called logical bitstreams, which can be combined into a single physical bitstream. Groups of concurrently multiplexed bitstreams may be chained consecutively, so that a physical bitstream obeys all the rules of both grouped and chained multiplexed streams [66].

2.1.5 Advanced Audio Coding (AAC)

Advanced Audio Coding (AAC) follows the main ideas of the MPEG Layer III, such high frequency resolution filterbank, non-uniform quantization, Huffman coding and iteration loop structure adopting analysis-by-synthesis, but improves the Layer-III in many aspects and uses new coding tools [68]. A block digram of an AAC encoder is depicted in Fig. 2.9.

To enhance the coding performances of the MPEG Layer III, AAC introduces an higher frequency resolution, an optional backward prediction which achieves better coding efficiency, an improved joint stereo coding wich allows to reduce the bit-rate more frequently and an improved Huffman coding. AAC improves also the audio quality adopting an enhanced block switching and a temporal noise shaping technique, which shapes the noise in the time domain, performing an open loop prediction in the frequency domain [68].

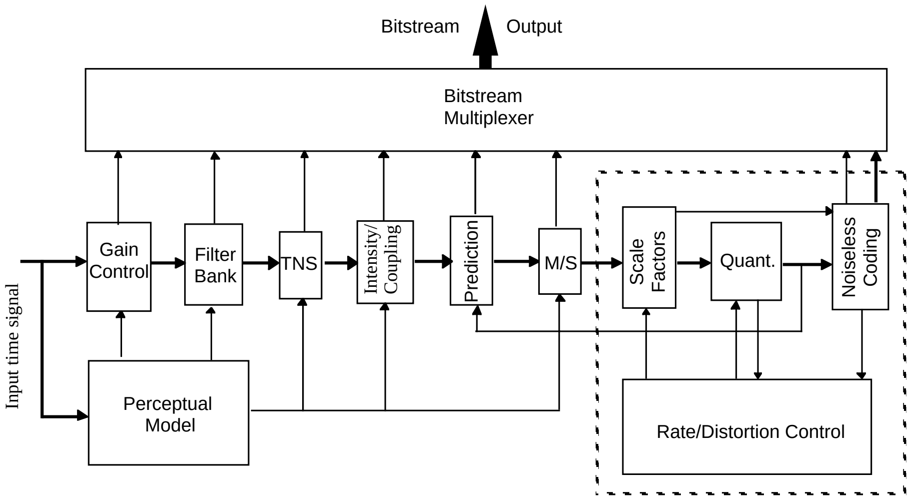

FIGURE 2.9: Block diagram of an AAC encoder.

2.1.6 Windows Media Audio (WMA)

Windows Media Audio (WMA) provides different versions of audio coding, like the Windows Media Audio 9, the Windows Media Audio 10 Professional, the Windows Media Audio 9 Lossless and the Windows Media Audio 9 Voice. Windows Media Audio 9 codec samples the audio at 44.1 or 48 kHz using 16 bits, providing CD quality at data rates that range from 64 to 192 kilobits per second. This allow one to reach an improvement up to 20 percent better than audio sampled with Windows Media Audio 8 when compared at the same data rate [69]. Windows Media Audio 10 Professional is the most flexible Windows Media audio codec available. It supports coding profiles that include everything from full-resolution 24-bit/96 kHz audio in stereo, 5.1 channel, or even 7.1 channel surround sound, to highly efficient mobile capabilities at 24 Kbps to 96 Kbps for stereo, and 128 Kbps to 256 Kbps for 5.1-channel sound. It supports streaming, progressive download, or download-and-play delivery at 128 to 768 Kbps [69]. The lossless version of Windows Media Audio creates a bit-for-bit duplicate of the original audio file aiming to not loose any data, resulting in lossless compression. Instead, the Windows Media Audio 9 Voice codec offers high quality for low-bit-rate streaming scenarios and the voice codec can also compress content to as low as 4 Kbps at 8 kHz [69].

2.2 Video coding

Digital video data requires a lot of storage or transmission capacity and with the always increasing request of video contents with higher quality, reducing the amount of data is fundamental. Video compression or video encoding is the process of reducing the amount of data required for representing a digital video sequence before being transmitted or stored [70], accounting for the statistical and subjective redundancy within and between frames. Digital video is the representation of a natural or real-world scene, sampled spatially and temporally. A scene is typically sampled at a certain point in time to produce a frame, which represents the complete visual scene in that specific moment or a field, formed by a number of lines of spatial samples [70]. During the coding process of a visual scene, many aspects such as texture variation, number of objects and their shapes or color needs to be considered and accounted. For representing a video scene in digital form, spatial sampling and temporal sampling are necessary. In the spatial sampling, the scene in a particular moment in time is sampled spatially, by a grid formed by sampling points, like that showed in Fig. 2.10.

The temporal sampling instead generates a sequence of frames. The higher is the sampling frequency, the better will be the appearences of the motion in the video. A further aspect is represented by the frames and fields. A video signal in fact may be sampled as a series of frames, which is called as progressive sampling, or as a sequence of interlaced fields, which instead is named as interlaced sampling. Another important thing is the colour. More precisely, while a monochromatic image requires only one value to represent the brightness or luminance of each spatial sample, an image with colors requires three number for pixel position, in order to represent brightness, luminance or luma [70]. In the Red, Green and Blue (RGB) color space, each color sample is

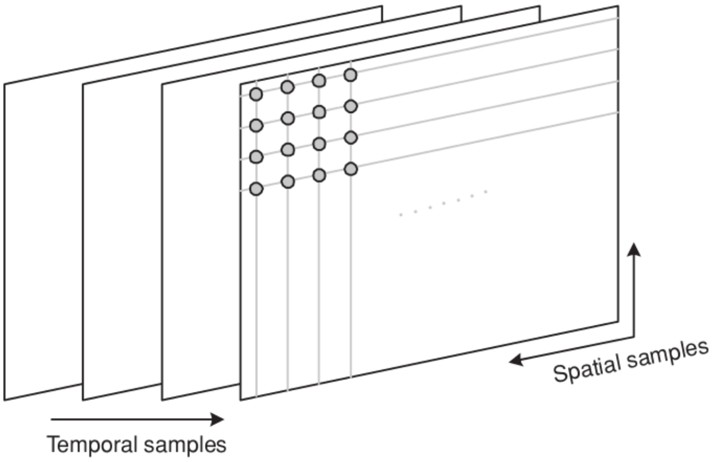

FIGURE 2.10: Spatial and temporal sampling of a video sequence [70].

represented with three values indicating the proportion of red, green and blue. In the Y:Cr:Cb color space instead, Y is the luminance component and can be obtained by the RGB values as:

$$Y = k_r\,R + k_g\,G + k_b\,B \tag{2.3}$$

where k_r, k_g and k_b represent weighting factors. To obtain the whole representation of the image in the Y:Cr:Cb color space, other three values are required, representing the difference between R, G and B components and the luminance Y:

$$C_r = R - Y\,, \tag{2.4}$$

$$C_b = B - Y\,, \tag{2.5}$$

$$C_g = G - Y\,. \tag{2.6}$$

Furthermore, different YCrCb sampling formats exist. The 4:4:4 (Y:Cr:Cb) format means that the three corresponding components are present for each pixel position. In the format 4:1:1 instead, for every 4 luminance samples, there are only one sample for Cr and one for Cb components.

In the video coding, many aspects need to be considered. A video encoder consists of three functionality units: a prediction model, an image model and an entropy model. The prediction model attempts to reduce redundancy by exploiting the similarities between neighbours video frames [70]. It can be of two kinds: temporal and spatial prediction. The image model, instead, focuses on de-correlating images or residual data and converting them into a form which can be easily compressed by an entropy coder. Finally, the latter one focuses on converting a series of symbols representing elements of the video sequence into a compressed bitstream.

In the following of this chapter, a quick overview of the most common video compression standards is presented.

2.2.1 Moving Picture Experts Group (MPEG) video

MPEG has been established in 1988 to compress audio and video contents and later has been extended to provide MPEG-2 video and associated audio compression algorithm for higher bit rates compared to the MPEG-1. MPEG-2 provides video quality not lower than National Television System Committee/Phase Alternating Line (NTSC/-PAL) and up to Consultative Committee for International Radio (CCIR) 601 quality with bit rates targeted between 2 and 10 Mbit/s [64]. The MPEG digital video-coding techniques are statistical and analyze the statistical redundancies in temporal and spatial directions. MPEG compression adopts DCT coding techniques for exploiting spatial correlation and achieve high data compression employing a combination of temporal motion-compensated prediction followed by transform coding of the remaining spatial information [64]. While the base idea behind sampling is to reduce the size of the image, the motion-compensated prediction reduces temporal redundancies between frames. In particular, the motion compensation focuses on estimating the motion between different frames, and describes it with motion parameters. Finally, the transform coding is employed to de-correlate the intra-frame or inter-frame error image content and to encode the transform coefficients. MPEG-1 provides frame based random access of video, fast-forward reverse searches through compressed bit stream, reverse playback of video and editability [64]. Based on macroblock structure, motion compensation and conditional replenishment of macroblocks, the MPEG-1 video compression technique encodes the first frame of a sequence as intraframe (I-frame), while the others are coded as interframe prediction pictures (P-frame). Furthermore the color input frames are partitioned into nonoverlapping macroblocks which contain blocks of data from luminance and chrominance bands. At the encoder the DCT is applied to each 8x8 luminance and chrominance block and then each of the 64 DCT coefficients are uniformily quantized. The lowest DCF coefficient, called DC coefficient, is encoded with a differential DC prediction method. The DC coefficient represents the average intensity of the component block. The nonzero quantizer values of the remaining DCT coefficients are then zig-zag scanned and run-length entropy coded using variable length code tables [64]. The decoder must perform the reverse operations extracting and decoding the variable-length coded words to obtain the locations and quantizer values of the nonzero DCT coefficients for each of the block [64]. MPEG-1 is capable to update macroblock information at the decoder if necessary and it allows one to tail the bit rate of a specific application constraint by adjusting the quantizer step-size. Furthermore, the MPEG-2 video has been designed to be backward compatible with MPEG-1 video and to add new features, like the capability to support interlaced video and scalability in the encoded video.

2.2.2 H.264

The H.264 [71] is a video compression standard co-produced by the ITU-T and the ISO/IES international standard bodies. The encoder is likely to mirror the steps of the decoding process and it lies on the concepts of previous standards such MPEG-2, but significantly offering better compression performance [70]. The H.264 encoder performs three main operations to obtain an encoded bitstream, and the decoder carries out the inverse operations. Generally the decode version is not equal to the original one, since the H.264 is a lossy compression standard. The three main operations are: prediction, transforming and encoding. The structure of a typical H.264 encoder is depicted in Fig. 2.11.

The encoder processes the data in groups of macroblocks which are used, together with the prediction macroblock, to generate the so called residual macroblock, which is tranformed, quantized and encoded. At the same time the content is re-scaled and added to the prediction macroblock after being inverse tranformed. In the decoder the inverse

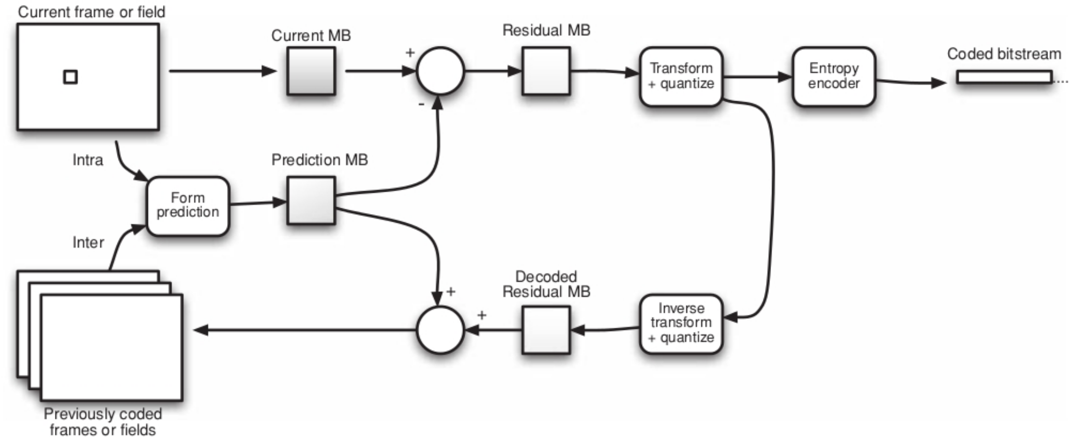

FIGURE 2.11: H.264 encoder [70].

operations are performed. The encoder generates a prediction of the macroblock based on previous data which are present in the current frame (intra prediction) or in other frames coded with intra prediction. The difference between the current macroblock and the predictions provides the residual (Fig. 2.12).

The block residual is then transformed using a 4 x 4 or 8 x 8 approximation of DCT, called integer transform, which provides a set of coefficients which, when combined, re-create the block of residual samples. The block of transform coefficients is then quantized according with quantization parameters. When QPs are high in value, the

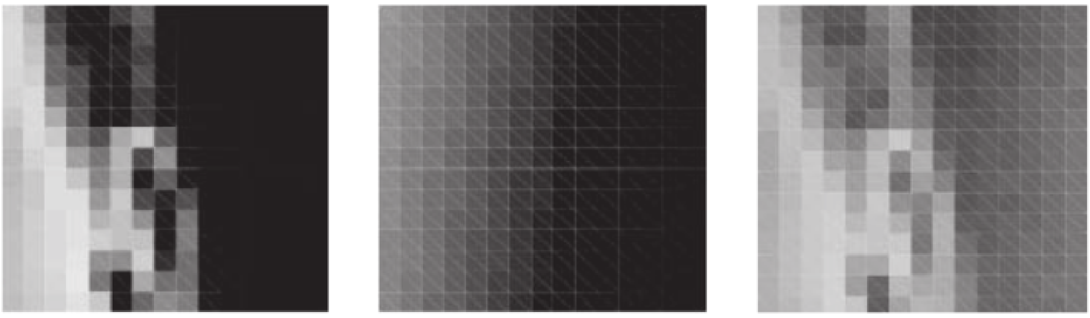

FIGURE 2.12: From left to right: original macroblock, intra prediction and residual macroblock [70].

compression will be higher with consequent reduction in the quality of the video once decoded. Finally the encoding process provides values, such as quantized transform coefficients, information to re-create the prediction, information on the structure of the compressed data and on the complete video sequence [70]. The decoder needs to perform the inverse operations, like decoding the bistream, rescaling the quantized transform coefficients and performing inverse transform and finally, reconstructing the decoded macroblocks which can be displayed as part of the frame. The H.264 standard supports many tools and profiles, and each profile defines a specific subset of tools. An encoded bitstream that conforms to a specific profile can only contain video data coded with some or all the tools present in that profile. Furthermore, an H.264 sequence is formed by a series of packets, called Network Adaptation Layer Units (NAL Units, or NALUs). Inside these NALUs, parameter sets which the decoder needs to know to correctly decode video slices and data, are sorted. A slice instead represents a frame or part of it, and consists of a number of coded macroblocks.

An extension of this standard is represented by the Scalable Video Coding (SVC). The term scalability can refer to the capability of removing parts of the video bit stream in order to adapt it to the various needs or preferences of end users as well as to varying terminal capabilities or network conditions. The objective of the SVC standardization has been to enable the encoding of a high-quality video bit stream that contains one or more subset bit streams that can themselves be decoded with a complexity and reconstruction quality similar to that achieved using the existing H.264/AVC design with the same quantity of data as in the subset bit stream [72]. One of the principle of the SVC extension is that all the new tools introduced by it can be added only if necessary for supporting the required type of scalability. There are three kinds of scalability: temporal, spatial and quality scalability. A bit stream provides temporal scalability when the set of corresponding access units can be partitioned into a temporal base layer and one or more temporal enhancement layers with the following property. Considering the temporal layers being identified by T_γ, with γ equal to 0 for the base layer and increasing of one unit for each enhancement layer, the remove of all the NALUs with temporal layer greater than γ, provides another valid bitstream for the given decoder [72]. Temporal scalability can be efficiently provided with the concept of hierarchical prediction structures and the coding efficiency is highly dependent on how the quantization parameters are chosen. Another scalability is represented by the spatial scalability. Each layer corresponds to a spatial resolution and is referred to a spatial layer or dependency identifier equal to zero for the base layer and increased by one from one spatial layer to another one [72]. Additionally, inter-layer prediction mechanisms are added, which enable the usage of as much as possible information from the lower layers, aiming to improve coding efficiency. Furthermore, the SVC introduces also the possibility of adopting inter-layer motion prediction and inter-layer residual prediction. Finally, the third scalability is the quality scalability which can be considered as a special case of spatial scalability with identical pictures size for base and enhancement layers. More precisely, the SVC standard defines three kind of quality scalability, the coarse-grain quality scalable coding (CGS), medium-grain quality scalable coding (MGS) and fine-grain quality scalable coding (FGS).

2.2.3 High Efficiency Video Coding (HEVC)

High Efficiency Video Coding (HEVC) [73], is a video compression standard developed by the Joint Collaborative Team on Video Coding (JCT-VC). HEVC has been developed aiming to provide twice the compression efficiency of the previous standard, H.264. More precisely, HEVC enables a video sequence to be compressed with half of the size or half of the bit rate compared to the H.264/AVC at an identical level of visual quality. The main idea behind HEVC relies on a technique known as motion compensated prediction. Blocks of pixels are encoded by making reference to another area in the same frame (intra-prediction), or in another frame (inter-prediction). Where H.264/AVC defines macroblocks up to 16x16 pixels, HEVC can describe a range of block sizes up to 64x64 pixels and it allows the predicted blocks to be coded in different block sizes than the residual error. Each of the top level coding unit is first coded as a prediction quad-tree, where at each depth the encoder decides whether to encode with merge/skip, inter, or intra coding. The residual from those predictions, is then coded with a second quad-tree [73].

Chapter 3

Distributed wireless networks

Wireless Local Area Networks (WLANs) represent an indispensable adjunct to traditional wired LANs, making possible to meet the always growing requests for mobility, relocation ad hoc networking and coverage. Early WLANs have been introduced in late 1980's targeting to substitute the wired LANs. The absence of wires allows to introduce higher mobility, reducing the costs for installations and maintenance, most of all in locations with not easy access. Wireless LANs needs to meet some requirements about throughput, number of nodes, battery power consumption, transmission robustness, security and dynamic configuration [74]. The nodes of a WLAN can differ from each other in computational power, mobility support, protocols adopted and connectivity. These parameters depend from many aspects of the nodes, like capability to support increasing traffic loads, processing operations and capabilities to perform routing functionalities. With the rise of new standards, nodes can act as bridges between different protocols allowing higher flexibility and network coverage. Considering the wireles access, a particular LAN is the Distributed Wireless Network (DWN) where there is not a central authority that manages the nodes. In these networks the nodes have knowledge of the characteristics of the neighbours nodes but not necessary of the other ones in the network. Examples of DWNs are the Mobile Ad-hoc Networks (MANETs), the

Vehicular Ad-hoc Networks (VANETs) and the Wireless Mesh Networks (WMNs). In particular, MANETs, which can be connected to the Internet or work for itself, are formed by a group of devices, each of whose are free to move independently and are router themselves. A particular MANET is a VANET. VANETs can be utilized for a broad range of safety and non-safety applications, such as vehicle safety, automated toll payment, traffic management, enhanced navigation and location-based services [75]. Furthermore WMNs are particular DWNs, characterized by self-configuration, scalability, self-organization and are capable to interoperate with other networks. Typically in these networks there are some nodes, called mesh clients, that constitute the backbone of the network and that provide connectivity to the mobile nodes, and other nodes, which are called mesh clients, that are usually less powerful and have less functionalities. All these families before mentioned have in common the lack of synchronization in the nodes access to the medium. For their importance and heavy use, all these networks are matter of interest for many industries and research activities.

3.1 The IEEE 802.11 standard

The IEEE 802 project can be modeled in three main layers. In particular, there is firstly the Physical layer, which focuses on the connection between the station and the medium, managing functionalities like encoding/decoding operation of the transmitted/received bits. Then, there is the MAC layer, which control the medium access functionalities and finally, there is the Logical Link Control (LLC) that focuses on the logical connections at the higher level of the datalink layer of the open Systems Interconnection (OSI) model. The IEEE 802 project defines different LAN architectures, that differ between each others in different aspects of the PHY and MAC implementation, being the LLC layer common to all the networks. These standards are known as IEEE 802.x, reported in Tab. 3.1 with the structure depicted in fig. 3.1.

IEEE 802 Standards	
802.1	Bridging and Management
802.2	Logical Link Control
802.3	Ethernet - CSMA/CD Access Method
802.4	Token Passing Bus Access Method
802.5	Token Ring Access Method
802.6	Distributed Queue Dual Bus Access Method
802.7	Broadband LAN
802.8	Fiber Optic
802.9	Integrated Services LAN
802.10	Security
802.11	Wireless LAN
802.12	Demand Priority Access
802.14	Medium Access Control
802.15	Wireless Personal Area Networks
802.16	Broadband Wireless Metro Area Networks
802.17	Resilient Packet Ring

TABLE 3.1: IEEE 802 family of standard.

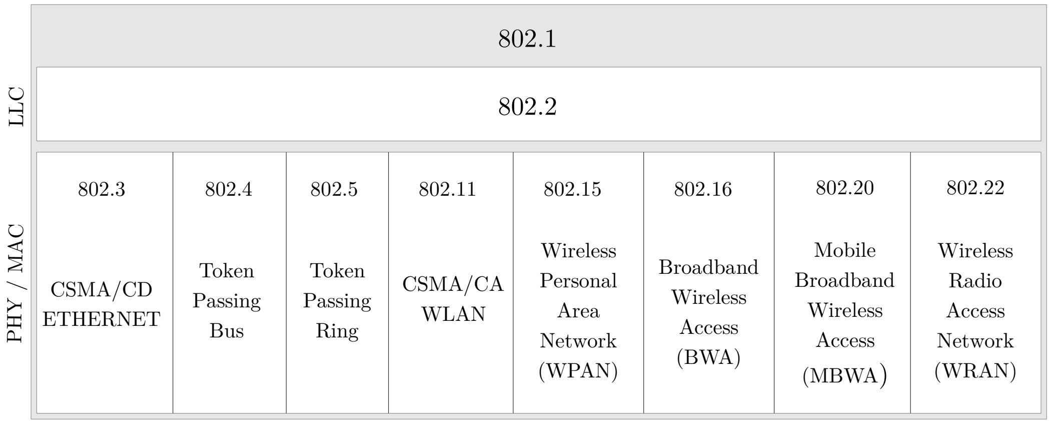
LLC
PHY / MAC
802.1
802.2
802.3
CSMA/CD ETHERNET
802.4
Token Passing Bus
802.5
Token Passing Ring
802.11
CSMA/CA WLAN
802.15
Wireless Personal Area Network (WPAN)
802.16
Broadband Wireless Access (BWA)
802.20
Mobile Broadband Wireless Access (MBWA)
802.22
Wireless Radio Access Network (WRAN)

FIGURE 3.1: IEEE 802 model.

In 1990 the 802.11 working group created the IEEE 802.11 standard, aiming to develop a MAC protocol and physical medium specifications for a WLAN. The 802.11 is very popular in the wireless communication field and it is the most spread standard adopted by DWNs. Among all the developed extensions, the standards aim improving the available rate [76], introducing QoS features [77] and providing security in the communication.

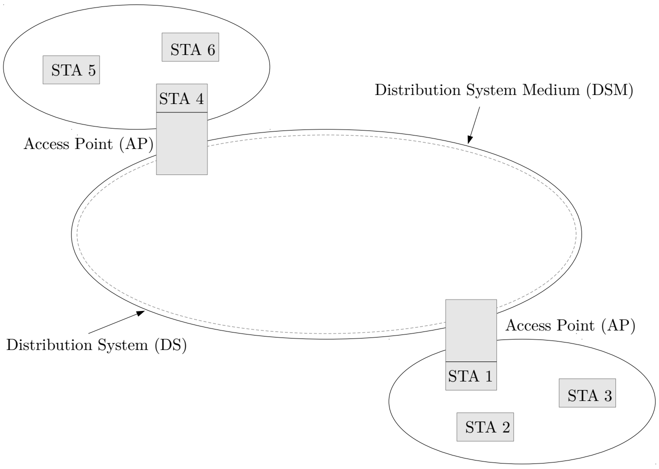

FIGURE 3.2: IEEE 802.11 components.

3.2 Architecture and services

The basic building block of a WLAN is represented by a Basic Service Set (BSS), constituted by a group of stations adopting the same MAC protocol, that compete to access the wireless medium. A BSS can stay for its own or can be connected by an Access Point (AP) to a backbone Distribution System (DS) fig.3.2.

In a BSS a number of wireless stations are associated to an AP and all the communications pass through this AP. When the stations can communicate directly, without the need of an AP, the BSS is called as Independent Basic Service Set (IBSS). This structure allows more freedom allowing the creation of ad-hoc networks. The MAC protocol can be fully distributed, or it can be controlled by a central coordination function present in the AP. Two or more BSSs connected to each other by a Distribution System (DS), such as a wired LAN, form an Extended Service Set (ESS). The AP is part of a node itself and a portal, such a router or a bridge, is used to connect the IEEE 802.11 structure with a wired LAN. The IEEE 802.11 defines 4 main services, which are resumed in Tab. 3.2 and an overview is provided in the following of this chapter.

Service	Provider
Distribution and Integration	Distribution system
Association, Disassociation and Reassociation	Distribution system
Authentication, Deauthentication and Privacy	Station
MSDU delivery	Station

TABLE 3.2: IEEE 802.11 services.

3.2.1 Distribution and integration system

Distribution and integration are two services that are necessary to distribute a message inside a DS. Distribution is a service used by each node for exchanging MAC frames that needs to go across the DS, reaching another node placed in a different BSS. If instead the receiving node is placed inside the same BSS of the transmitting one, this service is handled by the AP inside that specific BSS. Finally, the integration service allows data transfer between the 802.11 LAN and another possibly integrated 802.x LAN connected by a router or a bridge. It takes care of any address translation required for the transfer between the two LANs [78].

3.2.2 Association, disassociation and reassociation

The primary target of the MAC layer is to allow the transfer of MAC Protocol data units (MSDUs) between MAC layers of different nodes. For this purpose the mobility of a node needs to be considered and three scenarios are assumed: no transition, BSS transition and ESS transition [78]. The association consists in creating an initial association between a node and an AP. In this way the node can transmit or receive packets in the WLAN. To transmit and receive, the addresses need to be known and for this purpose, the AP establishes an association and get the address from the nodes. This address can be shared with other APs of different BSS, in the same ESS. When the association is concluded, a disassociation notification is sent, allowing the conclusion of it. Finally, to allow the move between different BSS, a reassociation procedure is performed, which move the established association to the AP of the new BSS.

3.2.3 Authentication, deauthentication and privacy

Differently from a wired LAN where a node needs to be physically connected to the LAN, in a WLAN each node can transmit and receive just if it is positioned in the wireless range of the nodes of that BSS. For this reason a WLAN requires an authentication procedure and a degree of privacy. In order to provide these services, the IEEE 802.11 supports many authentication schemes, that requires a successful authentication before establishing an association with the AP in that BSS. The authentication schemes can range from simple hand-shaking procedure, to public-key encryption schemes [78].

3.2.4 MAC Protocol data units (MSDUs) delivery

The MSDU is a block of data, fragmented if necessary in more MAC frames, passed by the MAC user to the MAC layer. MAC frames can be mainly of three kind: control frames, data frames and management frames. Control frames, can be Power Save-Pool (PS-Pool) Request to Send (RTS), Clear to Send (CTS), Acknowledgment (ACK), Contention-Free (CF)-End and CF-End+CF-Ack. They are used to assist the delivery of data frames. Data frames instead, can be associated with a CF-acknowledgment, a CF-Poll or both of them . Finally, management frames are used to manage communications between stations and APs [78].

3.3 Physical layer

Initially the original 802.11 standard defined at the PHY layer three physical media: direct-sequence spread spectrum operating in the 2.4 Ghz band using data rates of 1 or 2 Mbps, frequency-hopping spread spectrum operating in the 2.4 Ghz band using data rates of 1 or 2 Mbps, and infrared at 1 or 2 Mbps operating at a wavelength between 850 and 950 nm. In the first case, up to seven channels can be used with a data rate of 1 or 2 Mbps. The number of channels available, each one with a bandwidth of 5MHz, depends from the available bandwidth, that differs from country to country (13 in Europe to 1 only in Japan) [78]. The encoding scheme adopted is the differential binary phase shift keying (DBPSK) and the Differential Quadrature Phase-Shift Keying (DQPSK), depending on which bit rate is used.

3.3.1 IEEE 802.11b

Aiming to increase the throughput of the original standard the 802.11b amendment [55] has been proposed. This amendment maintains the media access of the original one but extend the DS-SS scheme to reach up to 11 Mbps of bit rate. The 802.11b adopts the modulation scheme known as complementary code keying (CCK). Input data are treated in blocks of 8 bits at a rate of 1.375 MHz and 6 of these bits are further mapped into one of 64 code sequences and then fed to a quadrature phase shift keying (QPSK) modulator [78].

3.3.2 IEEE 802.11a

This standard [79], which is not interoperable with the 802.11b, has been rectified in 1999. Basing on the same core of the original 802.11, this standard is the first one to have introduced the 5 GHz band. The maximum data rate reachable is of 54 Mbit/s thanks to the adoption of a 52-subcarriers orthogonal frequency division multiplexing (OFDM). The adoption of the OFDM represents a good advantage since the OFDM has very good propagation advantages in the presence of environments characterized by high multipath and the higher frequencies allows to adopt smaller antennas. Furthermore a convolutional code is adopted providing forward error correction.

3.3.3 IEEE 802.11g

With the start of 2003 the standard 802.11g [76] has started to be adopted on consumer devices. It works on the 2.4 GHz band like the IEEE 802.11b, but adopts the OFDM scheme used in the IEEE 802.11a standard, which allows to arrive to a bit rate up to 54 Mbit/s. At hardware level, this standard is fully compatibile with the b one and it suffers from the same interferences due to the high usage of the 2.4 GHz band.

3.3.4 IEEE 802.11n

This amendment defines modifications to both the 802.11 PHY and the 802.11 MAC layers. These modifications allow one to reach much higher throughputs, with a maximum throughput of at least 100Mb/s. The 802.11n standard introduces a number of new mechanisms aiming to increase the available bandwidth. Networks based on 802.11n can theoretically reach a data rate of 600 Mbps with 4 data streams [80]. Furthermore the g amendment improves the reliability of the communications also through a stronger wireless signal between AP and stations. 802.11n uses the OFDM scheme, like in the a amendment, and can use up to 52 carriers signals. The most important introduction is the adoption of the Multiple Input Multiple Output (MIMO) technology.

MIMO allows the transmitters and receivers to transmit up to four parallel data streams using the same transmission channel, with consequent increase in the data throughput and improved wireless coverage.

3.3.5 IEEE 802.11ac

In 2014 a new standard has been rectified, the IEEE 802.11ac [81]. This amendment aims to improve the maximum throughput mainly thanks to three fundamental contributions: larger channel width (80/160 MHz), support for denser modulation (256 QAM) and increased number of spatial streams for MIMO [81]. The 802.11ac amendment operates in the 5 GHz band only and present a denser modulation, adopting a 256 QAM modulation. A big change between 802.11n and 802.11ac is that the beamforming has been dramatically simplified, adopting the specification called Null Data Packet (NDP) sounding.

3.4 Medium Access Control layer (MAC)

In a WLAN the transmission of multimedia contents, requires a periodic access to the shared medium. The access procedure is controlled by the MAC layer. There are two main techniques to access the shared medium, the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF).

3.4.1 The Distributed Coordination Function (DCF)

The DCF basic method is based on the Carrier Sense Multiple Access Collision Avoidance (CSMA/CA). Before transmitting a frame, the station performs a carrier sensing, which means that it checks that the medium is in idle mode, i.e., no other stations are

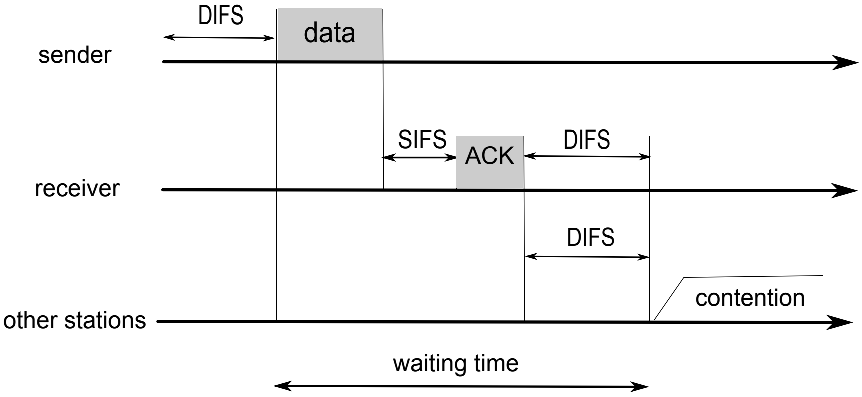

FIGURE 3.3: DCF basic access mechanism

transmitting. The carrier sensing can be of two kinds: virtual or physical. In the first one the sensing is performed at the MAC layer. If a packet received by a station is not for that station, its MAC layer reads the duration time present in the header of the frame, and accounting for the time required for transmitting the frame and receiving the ACK, it defers the access to the medium of the corresponding time. The Physical approach instead is performed directly at the PHY layer, sensing the medium. Considering a new packet, after the sensing procedure, if the medium stays in idle for a period of time equal to a DCF Inter-Frame Space (DIFS) and no other nodes are sensed, the station starts to transmit and all the other stations wait until the medium comes back in idle mode again for at least a DIFS period. When a station receives correctly a frame, it send a ACK message after waiting a Short Inter-Frame Space (SIFS). The SIFS period, that is shorter then the DIFS one, allows the stations to avoid a collision after a succesfull transmission. In fact the stations that completed the transmission with success, waits a SIFS before sending the ACK message, while the other stations have to wait a DIFS period, longer then the SIFS one. Thus, they keep waiting the idle mode on the medium. An example of this procedure is showed in Fig. 3.3.

When an PC/AP is present in the BSS, it does not use the DIFS period but another shorter one, called PCF Inter-Frame Space (PIFS). This shorter time is adopted by the

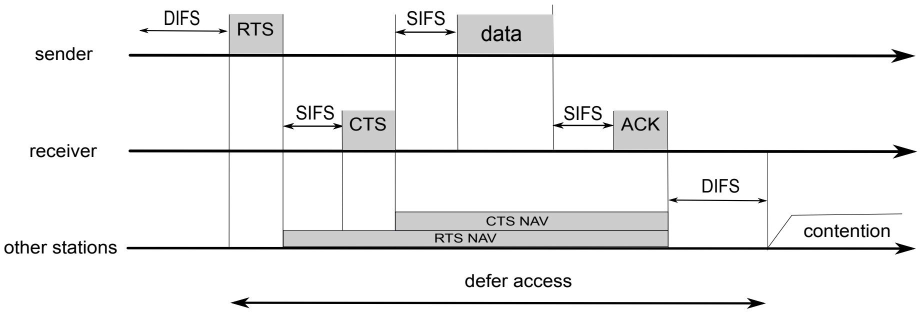

FIGURE 3.4: RTS/CTS handshaking mechanism

PC/AP aiming to have a higher priority compared to the other nodes in the BSS, since the AP waits a PIFS instead of a DIFS before transmitting after the medium passes in idle mode. SIFS, PIFS and DIFS are the group of Inter-Frame Space (IFS) and are defined by the underlying PHY layer adopted. If two or more stations transmit at the same time, a collision can occur. To avoid these collisions, when one of them occurs, each station has to wait an additional time before trying the access again. Each station waits until the medium becomes idle and generates a backoff, which represents a time period, that the station has to wait after the channel is idle for a DIFS period. During the idle mode, the node decrements the backoff counter. If the medium becomes busy due to a trasmission of another node, the station stops the decrease until the medium is again idle for a DIFS. After a transmission of another node, a station mantains its backoff counter, while the node that attempted the transmission will generate a new backoff. The generated backoff is randomly chosen in an interval between 0 and the Contention Window (CW). At every transmission attempt, if the transmission does not succeed, the contention window is doubled until it reaches the maximum size. The DCF defines also a retry limit, which defines how many attempts the station can retry the transmission before the frame is discarded and a new one, if available, is taken from the transmission queue. If, after a succesfull transmission, a node has another frame to transmit, it waits until the medium is free for another DIFS and then generates a backoff. This implies that between two frames from the same node, there is always a backoff time period, which allows other nodes to try to access the medium.

An additional optional mechanism used to solve the problem of hidden terminals is represented by the RTS/CTS mechanism. With the adoption of this mechanism, after the wait of a DIFS period during which the channel stays idle, before sending the frame, the node performs the RTS/CTS procedure. It sends an RTS message and the receiver sends a CTS message after a SIFS period. This handshaking procedure allows to reserve the medium to that particular transmission, and the other nodes in the wireless range set their Network Allocation Vector (NAV), which works similarly to a backoff counter. An example of this procedure is depicted in Fig. 3.4.

3.4.2 The Point Coordination Function (PCF)

PCF is an optional channel access mechanism for the support of delay-sensitive applications. This function is characterized by contention free access. A point coordinator (PC) manages a procedure of poll and response aiming to eliminate the contention between the nodes for accessing the medium. The PC takes control of the medium periodically. Within this period, it starts Contention Free Period (CFP), during which the medium access is granted and controlled only by the same PC. During the CFP, the PC can communicate with the nodes without any contention. Furthermore, it can send a contention-free poll (CF-Poll) to a station which, when receive it, can send a MAC frame to the PC for any CF-Poll received. If the polled node does not have anything to send in uplink, it will send to the PC a Null data packet. Once the CFP ends, a mandatory DCF period starts. The PCF procedure is often used for wireless multimedia streaming as it provides a guaranteed time to the applications that require real-time transmissions.

3.5 QoS at MAC layer: the Hybrid Coordination Function (HCF)

Aiming to introduce QoS at the MAC layer, in the 2005 the task group e (TGe) developed the 802.11e amendment. IEEE 802.11e supports QoS by adopting priority mechanisms and introducing a new coordination function, called Hybrid Coordination Function (HCF). This function extends the functionalities of the DCF by adopting the Enhanced Distributed Channel Access (EDCA), and the functionalities of the PCF by adopting the HCF Controlled Channel Access (HCCA).

3.5.1 Enhanced Distributed Channel Access (EDCA)

The EDCA provides differentiated access to the medium defining four access categories (Fig. 3.5), due to the type of data traffic. The four access categories, in decreasing order of medium access priority, are Voice (VO), Video (VI), Best Effort (BE) and BacK ground (BK), as showed in Tab. 3.3.

Priority	Access Category (AC)
1	Voice (VO)
2	Video (VI)
3	Best Effort (BE)
4	BacK ground (BK)

TABLE 3.3: Access categories defined by the IEEE 802.11e amendment.

The EDCA introduces a new access function, the Enhanced Distributed Channel Access Function (EDCAF). The EDCAF is an extension of the DCF, and basing on the same principles of CSMA/CA and backoff, adopts different parameters for each AC. Each AC is characterized by the Arbitration Inter-Frame Spacing (AIFS), the CW used for the backoff procedure, the maximum contention window and the Transmission Opportunity (TXOP) limit.

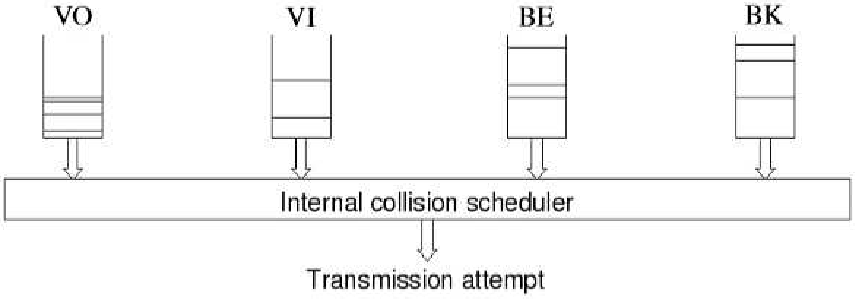

FIGURE 3.5: Access categories.

The AIFS represents the minimum time during which the medium must be sensed idle before a station attempts the access. In DCF this time is represented by the DIFS, but in EDCA, because of the presence of different ACs, this time must be differentiated for each class. To this purpose, it is introduced the Arbitration Inter-Frame Spacing Number (AIFSN), which is multiplied for the slot time and is summed to the SIFS to generate the AIFS for each class. The AIFS is different from the DIFS used in the DCF and its size change for each access category. For the AC with higher priority in the medium access the AIFS is low, granting a faster access to that AC, while it becomes larger in accordance with the decrease of the medium access priority of the corresponding AC. The lower priority AC suffers from longer wait time due to the low priority in the medium access. Another parameter that characterizes an AC is the contention window. CW is defined by its minimum and its maximum value and they depend from the AC. Higher priority implies lower values of minimum and maximum contention windows, granting a lower value of random backoff. Furthermore, the maximum contention windows for the ACs with higher priority are set in such a way that their value results to be equal or less then the minimum contention window of lower priority ACs.

Finally, another important parameter is the TXOP. It represents the time during which a station may transmit after gaining access to the medium. Its maximum value, called TXOP limit covers the time corresponding the SIFS, ACK and RTS/CTS if used. When the TXOP limit is non zero, the transmission of multiple frames of the same AC during the TXOP period is possible, and this is called Contention Free Bursting (CFB). TXOP can change due to the use of different PHY layers (Frequency-Hopping Spread Spectrum (FHSS), Direct Sequence Spread Spectrum (DSSS), OFDM or OFDM/MIMO).

Part II

Original Results

Chapter 4

Fast retry limit adaptation for video distortion/delay control

The possibility to manage video traffic in 802.11 distributed wireless networks represents a challenging task due to the unreliability of the wireless medium and the presence of contention-based mechanisms [82]. These aspects are particularly relevant for streaming applications, which have often to fulfill stringent quality of service (QoS) requirements for satisfying the user's demand [83, 84]. Therefore, to introduce QoS control at the MAC layer of an 802.11 network, task group e (TGe) developed the 802.11e amendment, which extends the functionalities of the legacy DCF by adopting the EDCA [85]. The EDCA enables a prioritization of the traffic during the contention period by defining four ACs: VO, VI, BE, and BK, whose differentiation is based on four parameters: transmission opportunity, AIFS, minimum and maximum contention windows. The 802.11e extension establishes the values of these parameters according to the AC and the PHY layer technology available among those adopted in the 802.11a/b/g/n amendments.

The content of this chapter is based on F. Babich, M. Comisso, and R. Corrado, "Fast Retry Limit Adaptation for Video Distortion/Delay Control in IEEE 802.11e Distributed Network", *Elsevier Ad Hoc Networks*, Vol. 36, No. 1, pp. 229 - 243, January 2016.

Even if the EDCA settings are specified to provide a higher priority to the VO and VI ACs, collisions involving audio/video packets may still occur, thus making necessary a proper policy of management of the retransmissions. Accordingly, several studies have investigated this issue, by proposing useful methods for optimizing the retry limit associated to each video packet [42, 86–95]. The main objective of these methods is the adaptation of the number of retransmissions to the acceptable delay and the perceived distortion. This leads to the derivation of elaborate optimization strategies, able to provide significant performance improvements with respect to those achievable using the 802.11e default settings. However, two relevant aspects are often neglected in these proposals. First, the presence of the higher priority VO AC, which can considerably reduce the access opportunities for the video packets. Second, the complexity of the conceived solution, which may be difficult to implement on the commercially available 802.11 network interface cards, that are characterized by low computational resources. Therefore, an alternative approach may be developed by moving from a more realistic model in which both the VO and VI ACs can be active, and considering the satisfaction of the QoS requirements together with a minimization of the necessary calculations.

The retry limit adaptation method proposed in this chapter deals with these two issues. The method, which is derived by relating the drop probability to the video distortion, and the packet delay to the expiration time, explicitly accounts for the impact of higher priority VO AC on the evolution of the video transmission. Additionally, both the distortion estimation algorithm and the retransmission strategy are developed with the purposes of limiting the computational cost and of not requiring feedback distortion/delay information from the destination. A theoretical evaluation of the optimum retry limit and an existing retransmission strategy are used as benchmarks for validating the performance of the presented method, which is implemented in a network simulation platform including the physical communication chain and the 802.11e EDCA.

The chapter is organized as follows. Section 4.1 presents a brief literature overview about the topic of this chapter. Section 4.2 introduces the analyzed system. Section 4.3

describes the adopted theoretical model and the conceived adaptation algorithm. Section 4.4 discusses the numerical results. Section 4.5 summarizes the most relevant conclusions.

4.1 Related work

The interest in the development of an optimization strategy for the retry limit derives, firstly, from the influence of this parameter on the performance figures of the network (throughput, successful packet delay, drop probability), and, secondly, from the absence of mandatory specifications for its setting. In particular, this second aspect guarantees a certain flexibility to the designer, which instead is not guaranteed for the other EDCA parameters (TXOP, AIFS, minimum and maximum contention windows), whose values are specified by the 802.11e standard according to the adopted PHY layer extension [85]. Moreover, this flexibility becomes more relevant when streaming applications are involved, since the possibility to associate a different number of retransmissions to a different packet allows the designer to better match the QoS requirements in the presence of video traffic flows.

Accordingly, several retry limit adaptation methods have been proposed in the research literature [42, 86–95]. The QoS strategy presented in [42] adopts a priority queueing also at the network layer, in order to relate the adaptation to the tradeoff between drop probability and buffer overflow rate. The optimal retry limit estimation in [86] derives from a minimization of the total expected distortion relying on classification and machine learning techniques. In [87] the conventional count-based retransmission scheme is replaced by a time-based one, in which the deadline is determined by the expiration time and the importance of the inter-coded frames. A retry limit adaptation method for scalable videos is developed in [88] by considering the collision probability as a load indicator. The reciprocal influence among the nodes and the ACs on the selection of the retry limit is analyzed in [89], where an adaptive algorithm is derived from the numerical solution of a nonlinear system. A cross-layer content-aware scheme for scheduling the retransmissions is proposed in [90] by considering the estimated backoff time and the macroblock-level loss impact. Closed-form estimations of the retry limit in the presence of collisions and buffer overflows are obtained in [91] by modeling the 802.11 MAC layer as an M/G/1 queueing system. The concept of virtual buffer size is introduced in [92] to develop an adaptation strategy for delay-critical video transmissions in lossy networks. A video-coding aware MAC layer is proposed in [93], with the purpose of delivering a video stream in which the retry limit is adjusted to guarantee a delay reduction and a satisfactory peak signal-to-noise ratio (PSNR). A fragment-based retransmission scheme suitable for video traffic is developed in [94], where the aim is to decrease the duration of the retransmission attempts in the presence of channel errors. In [142] the mean square error and the structural similarity are compared as video quality assessments for developing adaptive retransmission strategies. A tradeoff between energy efficiency and satisfaction of the QoS requirements for centralized operations is obtained in [95] by a joint dynamic adjustment of the contention window and of the retry limit.

This brief overview shows that the available adaptation policies for the retry limit of the VI AC in distributed environment are developed with the aim of satisfying two main objectives: management of the distortion and control of the delay. Except for [89], the proposals are conceived assuming the absence of the VO AC, and are not focused on the limitation of the computational complexity. The aim of the strategy presented in this chapter is to provide an adaptive algorithm able to account for the distortion/delay requirements of the transmitted video sequence, considering, as additional purposes, limitation of the processing time and possibility to operate in the presence of higher priority traffic.

4.2 System description

Consider the MAC layer of an 802.11e distributed network, and hence a single-hop scenario involving N sources and the corresponding N destinations. All the $2N$ nodes operate using the EDCA basic access mechanism combined with an 802.11g PHY layer. Each source S contends with the other sources for gaining access to the wireless medium in order to deliver its packets to the intended destination D. Except for the mandatory ACK packet, the destination does not provide any feedback information concerning the distortion and the delay, which hence must be estimated by the source on its own. In particular, S can support four ACs, which are numbered according to $q=1$ (VO), $q=2$ (VI), $q=3$ (BE), $q=4$ (BK), thus indicating that a lower q value identifies a higher priority. Assume that each AC of each source remains nonempty once a packet is successfully transmitted, hence considering, as in [88, 90], saturated traffic conditions. For the BE and BK ACs, the saturation assumption is widely accepted, since it derives from usual file transfer applications. For the VO and VI ACs, the saturation hypothesis is justified by the transmission policy usually adopted by many common streaming services, such as YouTube, according to which the packets corresponding to the first 40 seconds of a requested stream are immediately sent, while the sending rate adopted for the rest of the stream must be slightly higher than the playback rate, since the objective is to avoid the interruption of the reproduction [96]. This policy implies that a large amount of packets may be considered already present also at the transmission queues corresponding to the VO and VI ACs, thus allowing to assume a saturated scenario.

Among the entire load that must be delivered to D, S has to transmit a video sequence $\mathcal{V} = \{\nu_l : l = 1, ..., L\}$, which includes L frames $\nu_1, ..., \nu_L$. The source-destination model is reported in Fig. 4.1, where four operations are considered: the encoding of $\mathcal{V}$, the estimation of the distortion and of the expiration time, the adaptation of the retry limit, and the decoding of the received video. The first three operations are carried out by the source, while the latter one by the destination. The three following subsections

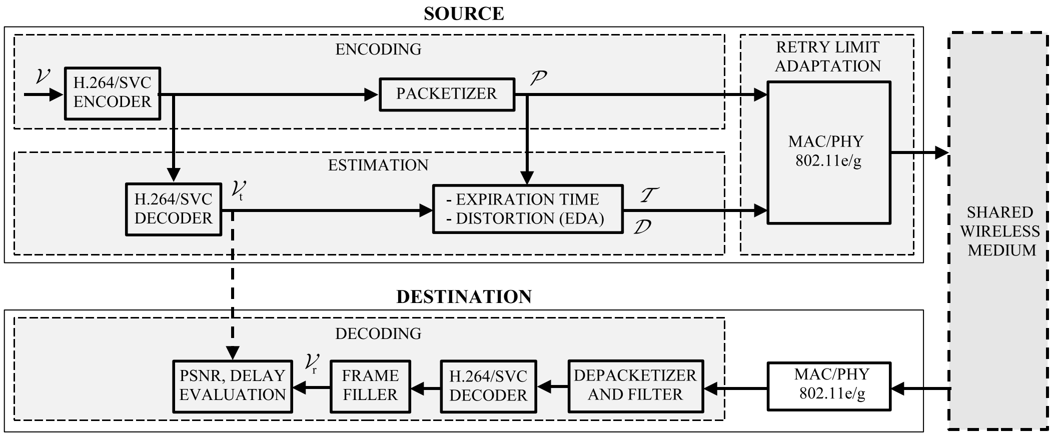

FIGURE 4.1: Model for the generic source-destination pair.

describe the encoding, the estimation, and the decoding operations, while the retry limit adaptation is presented in Section 4.3.

4.2.1 Encoding

The video $\mathcal{V}$ is encoded using the H.264 SVC standard developed by the joint video team (JVT) [97]. Thus, $\mathcal{V}$ is subdivided into groups of pictures (GOPs) of size α, and encoded to obtain a set of NALUs. Each NALU, which is created considering the dependencies within a GOP, is classified according to the type of the corresponding frame: Intra-coded (I), Predictively-coded (P), and Bipredictively-coded (B), and is generated to encode the video as an independently decodable base layer and a certain number of enhancement layers [98]. The set of NALUs, which are of different size, is then packetized to obtain a set $\mathcal{P}$ of K packets $\pi_1, ..., \pi_K$ of equal size that are transmitted over the network. Thus, at the end of the packetization process, the encoded version of a generic frame ν_l may be fragmented in a certain number of 802.11 packets. For calculation purposes, it may be then useful to define, for each $\nu_l \in \mathcal{V}$, the set $\mathcal{P}_l(\subset \mathcal{P})$ of the packets containing the NALUs of ν_l. In particular, $\mathcal{P}_l$ has k_l elements and hence the overall number of packets K that derives from the encoding of the original video sequence $\mathcal{V}$ can be expressed as $K = \sum_{l=1}^{L} k_l$. By consequence, the set $\mathcal{P}_l$ contains the packets having indexes k between $K_{l-1}+1$ and K_l, where $K_l = \sum_{l'=1}^{l} k_{l'}$, thus $\mathcal{P}_l = \{\pi_k : k = K_{l-1}+1, ..., K_l\}$.

4.2.2 Estimation

Since each NALU may have a different impact on the overall video quality, the retry limit for each packet π_k should be selected according to its expiration time T_{e_k} and to the distortion D_k produced by its possible loss. It is useful to first relate these two quantities to the frames, which represent the real video content perceived by the final user, and subsequently refer them to the packets. As a first step, the NALUs generated by the H.264/SVC encoder are decoded to obtain the sequence $\mathcal{V}_{\mathrm{t}} = \{f_l : l = 1, ..., L\}$, which may be considered as a reference sequence that is physically transmitted over the network. This sequence will be compared with the video received at the destination (Fig. 4.1), in order to enable a performance evaluation of the proposed framework that accounts for the losses due to the sole access procedure, and not for the lossy compression, whose effects are out of the scope of this chapter.

Let's now consider the expiration time. To this aim, one may observe that usually the player at the destination awaits the reception of a certain number of frames $\bar{l}$ before starting the play of the video. From now on in this chapter, $\bar{l}$ will be referred to as the expiration time index. Therefore, one can assume that the requirement on the expiration time holds for the frames successive to a frame $f_{\bar{l}}$, while the frames previous to $f_{\bar{l}}$ may be associated to an infinite expiration time. Accordingly, the expiration time for the l-th frame can be evaluated as [87, 92]:

$$\tilde{T}_{\mathrm{e}_l} = \begin{cases} +\infty & l = 1, ..., \bar{l} \\ (l + M_l)T_{\mathrm{f}} & l = \bar{l}+1, ..., L \end{cases} \tag{4.1}$$

where M_l is the number of frames inter-coded with f_l in the same GOP and T_f is the inter-frame interval. The quantity in (4.1) is considered to control the delay at the destination, in order to limit the interruptions of the playback of the video due to the wait of the arriving frames.

From a practical point of view, the most accurate method for estimating the impact of the loss of a packet on the corresponding GOP would require the removal of the packet itself and the subsequent decoding of the entire GOP according to the adopted error concealment strategy [90]. Since this method would be computationally too expensive, alternative approaches have been derived [99, 100]. In particular, the existing distortion estimation techniques for H.264 encoded videos may be classified in two families: lightweight methods and sophisticated methods [101]. The lightweight methods are computationally cheap, since they just distinguish between key and non-key frames [102]. However, they do not provide a fine estimation of the distortion effect determined by the loss of a frame, thus making preferable the sophisticated methods when more accurate estimations are necessary. Among this second family of distortion estimation techniques [48, 50–52, 103], which are often characterized by a high computational cost, the exponential distortion algorithm (EDA), presented in [6, 51, 104], is one of the few sophisticated algorithms that enables to model the distortion due to the loss of a frame f_l maintaining a low complexity. For this reason, the EDA is adopted in this chapter.

The EDA assumes the adoption of a frame copy error concealment at the decoder, since the lost frame f_l is replaced by the previous received one f_{l-1}. Therefore, considering the frame f_l and a succeeding one $f_{l'}$, both belonging to the same GOP, the EDA estimates the distortion suffered by $f_{l'}$ because of the loss of f_l as the product $\mathrm{MSD}[f_l - f_{l-1}]e^{-\xi(l'-l)}$, where $\mathrm{MSD}[f_l - f_{l-1}]$ is the mean square difference between f_l and f_{l-1} that estimates the actual mean square error at the decoder, and ξ is a parameter dependent on the encoded video that accounts for the error propagation effect. Using this approach, the distortion on the entire GOP of size α due to the loss of the frame f_l can be evaluated as:

$$\tilde{D}_l = \sum_{l'=l}^{\lceil \frac{l}{\alpha} \rceil \alpha} \mathrm{MSD}[f_l - f_{l-1}] e^{-\xi(l'-l)}, \tag{4.2}$$

where $\lceil \cdot \rceil$ denotes the ceiling function. Further details concerning the EDA can be found in [6, 51, 104].

The two sequences of estimations $\tilde{T}_{\mathrm{e}_1}, ..., \tilde{T}_{\mathrm{e}_L}$ and $\tilde{D}_1, ..., \tilde{D}_L$, which are related to the frames, must be then related to the packets. To this aim, one may observe that, if a frame f_l, with $l > \bar{l}$, is characterized by the expiration time $\tilde{T}_{\mathrm{e}_l}$ and by the set $\mathcal{P}_l$, the corresponding k_l packets would not all be associated to the same $\tilde{T}_{\mathrm{e}_l}$ value. In fact, in this case the transmission of the first packet of $\mathcal{P}_l$ might use all the time margin, subsequently forcing the remaining packets, having the same expiration time, to adopt a retry limit equal to zero. To avoid the occurrence of this event, the interval $\tilde{T}_{\mathrm{e}_l} - \tilde{T}_{\mathrm{e}_{l-1}}$, theoretically available for the transmission of the k_l packets corresponding to the l-th frame, is subdivided into k_l equal subintervals. Therefore, recalling (4.1), the expiration time associated to $\pi_k \in \mathcal{P}_l$ remains infinite for $k = K_{l-1}+1, ..., K_l$ and $l = 1, ..., \bar{l}$, while it is evaluated as:

$$T_{\mathrm{e}_k} = \frac{\tilde{T}_{\mathrm{e}_l} - \tilde{T}_{\mathrm{e}_{l-1}}}{k_l} (k - K_{l-1}) + \tilde{T}_{\mathrm{e}_{l-1}}, \tag{4.3}$$

for $k = K_{l-1}+1, ..., K_l$ and $l = \bar{l}+1, ..., L$. The linearized approach in (4.3) aims to fairly subdivide the time available to transmit a frame among all packets containing NALUs that belong to that frame.

The distortion corresponding to $\pi_k \in \mathcal{P}_l$ can be calculated by considering that associated to the frame f_l normalized to the maximum, thus:

$$D_k = \frac{\tilde{D}_l}{\max\limits_{l \in \{1,...,L\}} \tilde{D}_l}, \tag{4.4}$$

for $k = K_{l-1}+1, ..., K_l$ and $l = 1, ..., L$. Observe that, since the indexing in k is related to the indexing in l, all packets $\pi_k \in \mathcal{P}_l$ associated to a frame f_l have an identical normalized distortion. For this reason, the index k does not explicitly appear in the right hand side of (4.4). The motivation for the normalization in (4.4) can be explained observing that D_k will be related to the drop probability, thus it is useful to identify a measure of the distortion lying between 0 and 1. As it will explained in Subsection 4.3.2, D_k may be further scaled according to the specific application, if required. However, the availability of a normalized quantity may represent a reasonable starting point for the subsequent exploitation of the distortion, also in the case in which other estimation techniques are adopted. Summarizing, the process of estimation carried out at the source S provides, for the set of packets $\mathcal{P}$, the two sets of estimations $\mathcal{T} = \{T_{\mathrm{e}_1}, ..., T_{\mathrm{e}_K}\}$ and $\mathcal{D} = \{D_1, ..., D_K\}$ that will be used at MAC layer to adapt the retry limit of the VI AC.

4.2.3 Decoding

At the destination, the received packets are depacketized to derive the set of the received NALUs, which are filtered to remove, firstly, all NALUs that have been at least partially lost due to the loss of the corresponding packets and, secondly, the NALUs relative to frames whose base layer has not been received and hence cannot be decoded [98] (Fig. 4.1). The set of remaining NALUs is passed to the H.264/SVC decoder and the result is filled with the lost frames, thus obtaining the received video $\mathcal{V}_{\mathrm{r}}$, which is compared to the transmitted video $\mathcal{V}_{\mathrm{t}}$. The filling and comparison operations, which would not be carried out in a real network, are performed just for modeling purposes, in order to enable a frame-by-frame comparison between the reference video $\mathcal{V}_{\mathrm{t}}$ and the received one $\mathcal{V}_{\mathrm{r}}$, so as to evaluate the PSNR and the delay for the decodable frames. As discussed at the beginning of Subsection 4.2.2, this comparison enables to isolate the effects due to the 802.11 dropped packets from those due to the lossy compression.

4.3 Retry limit adaptation

A reliable model for a distributed network may be derived adopting a Markov approach [105, 106], which has been extensively used to investigate the performance of an 802.11-based uncoordinated network in several contexts, including the presence of non-saturated conditions [107, 108], directional communications [109], multiple ACs [110], and heterogeneous traffic sources [111]. Accordingly, the here developed adaptation strategy for the retry limit associated to each packet $\pi_k \in \mathcal{P}$ is based on a Markov model of the 802.11e EDCA [110]. This model, which has been validated by experimental measurements realized using a real testbed, is properly re-elaborated to obtain a reduced set of simplified equations that enable to reliably estimate the network behavior with a low computational cost. The next subsection introduces this Markov model with the purpose of briefly summarizing the approach presented in [110], so as to better identify the mathematical context from which the proposed algorithm is derived. Subsequently, Subsection 4.3.2 presents, as the main contribution, the developed retry limit adaptation strategy.

4.3.1 Theoretical model

According to the 802.11e EDCA specifications and the network scenario described in Section 4.2, in the presence of equal transmission opportunities the q-th AC of the generic source can be characterized by the AIFS AIFS_q, the minimum contention window W_q, the retry limit m_q, and the maximum backoff stage m'_q, which determines the maximum contention window. The function of these parameters can be explained by describing the EDCA backoff procedure.

When a packet belonging to the q-th AC is ready for transmission, the source monitors the medium for a time AIFS_q. If the medium is sensed idle during this time, the packet is immediately transmitted, otherwise a random backoff is generated as the product

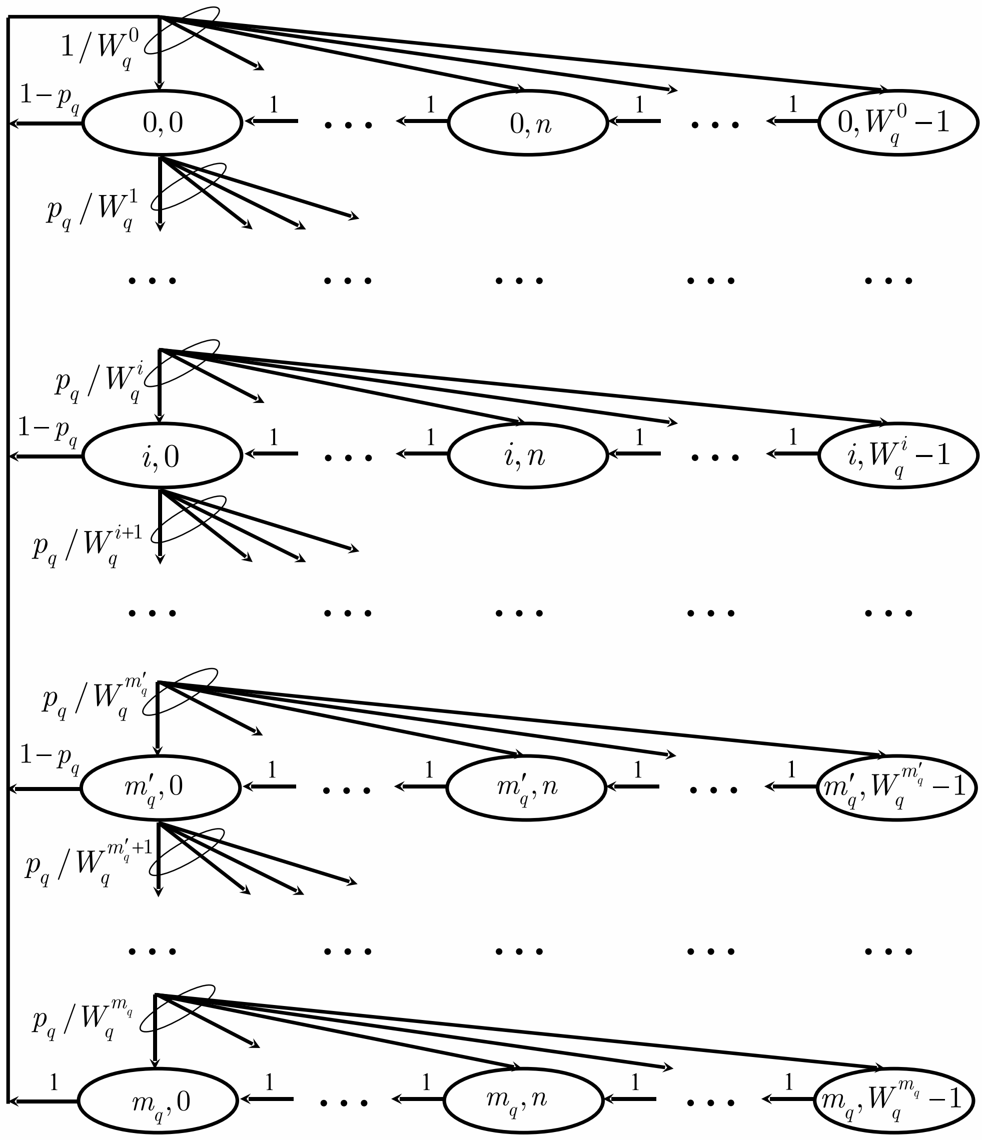

FIGURE 4.2: Markov model for the q-th AC of the generic source.

between a constant slot time and a random integer n uniformly distributed in the interval $[0, W_q - 1]$. This backoff is inserted in a reverse counter, which is decreased when the medium is sensed idle, frozen when the medium is sensed busy, and reactivated when the medium is sensed idle again for an AIFS_q. When the counter reaches the zero value the packet is transmitted. If the transmission is successful, the source receives an ACK packet from the destination. Otherwise, a retransmission is scheduled by updating the retry counter and the contention window. In particular, at the i-th retransmission attempt, the contention window is evaluated as:

$$W_q^i = 2^{\min(i,m'_q)} W_q, \tag{4.5}$$

and the backoff decrease process is repeated adopting a random integer n uniformly distributed in the interval $[0, W_q^i - 1]$. When the retry counter reaches the retry limit m_q, the packet is discarded. With reference to the q-th AC and assuming identical AIFS values for the four ACs, this mechanism can be modeled considering the Markov chain in Fig. 4.2, where p_q denotes the conditional collision probability. This figure describes the backoff procedure by a two-dimensional process in which the generic state (i, n) identifies, for a generic packet, a residual backoff of n slots at the i-th transmission attempt. According to the scenario introduced in Section 4.2, the model assumes saturated traffic conditions, since, once a packet is successfully transmitted or is discarded due to the achievement of the maximum number of retransmissions, a novel packet is immediately available. Observe that the assumption of identical AIFS values, which may seem to limit the applicability of the analysis, is acceptable in the case considered in this study. The suitability of this assumption will be justified is detail in Subsection 4.3.2.

Analyzing the chain in Fig. 4.2, one can express the generic steady-state probability $\eta_{i,n}$ as a function of $\eta_{0,0}$, hence obtaining [110]:

$$\eta_{i,n} = \left(1 - \frac{n}{W_q^i}\right) p_q^i \eta_{0,0}, \tag{4.6}$$

for $n \in [0, W_q^i - 1]$ and $i \in [0, m_q]$. Therefore, using (4.6) and imposing the normalization condition, one obtains:

$$\eta_{0,0} = \left[\sum_{i=0}^{m_q} \sum_{n=0}^{W_q^i - 1} \left(1 - \frac{n}{W_q^i}\right) p_q^i\right]^{-1}. \tag{4.7}$$

The probability τ_q that the source attempts the transmission can then be evaluated by summing over all the steady-state probabilities with backoff equal to zero, thus:

$$\tau_q = \sum_{i=0}^{m_q} \eta_{i,0}. \tag{4.8}$$

Using (4.5)-(4.7) in (4.8) and performing some algebra, one can therefore obtain the first set of equations of the system:

$$\begin{cases} \tau_q = \dfrac{2(1-2p_q)\left(1-p_q^{m_q+1}\right)}{(1-2p_q)\left[1-p_q^{m_q+1}+p_q W_q 2^{m'_q}\left(p_q^{m'_q}-p_q^{m_q}\right)\right]+W_q(1-p_q)\left[1-(2p_q)^{m'_q+1}\right]} \\ p_q = 1-\displaystyle\prod_{q'=1}^{4}(1-\tau_{q'})^{N-1}\prod_{q'=1}^{q-1}(1-\tau_{q'}) \end{cases} \tag{4.9}$$

which is defined for $q=1,...,4$, and hence consists of $2{\times}4=8$ equations. The second set of equations in (4.9) expresses the conditional collision probabilities p_q for $q = 1, ..., 4$, according to the fact that a packet belonging to the q-th AC of a given source S collides in two cases. First, if S and at least another source transmit their packets at the beginning of the same slot time (external collision). Second, if, at the source S, the backoff of the elaborated packet and that of a packet belonging to an AC with a higher priority reach the zero value at the same time (internal collision). In this second case the collision is directly resolved at the source S by allowing the transmission of the packet with the higher priority and considering as collided the packet with the lower priority. Further mathematical details for the derivation of (4.9) can be found in [110]. The nonlinear system of eight equations in (4.9) represents the core of the model, since it enables the calculation of the transmission and collision probabilities for the ACs of interest. The parameters W_q and m'_q for $q = 1, ..., 4$ are assumed known, since they are specified in the 802.11e standard for a given PHY layer [85]. Instead, the parameter m_q and the quantities τ_q and p_q for $q = 1, ..., 4$ are assumed unknown, thus $3 \times 4 = 12$ unknowns are present in (4.9). For the case $q = 2$, corresponding to the VI AC that is of interest in this study, m_2 depends also on the specific video packet π_k. However, to simplify the notation, this dependence will be explicitly introduced afterwards, thus currently considering the network behavior for a given video packet.

Moving from (4.9), the proposed approach for the derivation of a retry limit adaptation algorithm first considers the collision and transmission probabilities of the active ACs, from which the drop probability and the packet delay are estimated. Subsequently, the retry limit for each packet is evaluated by relating the drop probability to the distortion, and the packet delay to the expiration time.

4.3.2 Adaptation algorithm

The first approximation introduced to simplify (4.9) relies on the practical observation that the main impact on the collision probability of a given AC is due to the ACs having a higher or equal priority, and hence a higher or equal transmission probability [110]. The 802.11e standard states that, when a PHY layer specifies a minimum contention window and a maximum backoff stage, these parameters hold for the BK AC. Thus, they must be intended as W_4 and m'_4, respectively, and must be used to obtain the corresponding parameters for all the ACs as $W_1 = W_2/2 = W_3/4 = W_4/4$, $m'_1 = m'_2 = 1$, $m'_3 = m'_4$ [85], in order to provide a higher priority to the VO and VI ACs. For the same reason, in the 802.11e extension, the AIFS values are selected as $\mathrm{AIFS}_1 = \mathrm{AIFS}_2 > \mathrm{AIFS}_3 > \mathrm{AIFS}_4$ [85]. Since the higher the AIFS, the minimum contention window, and the maximum backoff stage, the lower the transmission probability, these settings imply that $\tau_3, \tau_4 < \tau_2 < \tau_1$. This allows one to neglect, on first approximation, the impact of the BE and BK ACs on the remaining ones, thus assuming $\tau_3, \tau_4 \cong 0$, and the impact of the VI AC on the VO one. Besides, being in this study the interest focused on the VI AC, the equations expressing p_3 and p_4 as functions of the transmission probabilities in (4.9) can no longer be considered. Observe that, since $\mathrm{AIFS}_1 = \mathrm{AIFS}_2$, the Markov chain in Fig. 4.2 properly describes the behavior of both the VO and VI ACs without the need of additional states, which instead would be required if the interest had been focused also on the BE and BK ACs, in order to account for the larger AIFS_3 and AIFS_4 values [112]. Regarding the parameter settings specified in the 802.11e amendment for the VO and VI ACs, a final aspect that is worth noticing concerns the low number of possible contention windows enabled by the unity backoff stage, that is, $W_1^0 = W_1$ and $W_1^1 = 2 \cdot W_1$ for the VO AC, and $W_2^0 = W_2 = 2 \cdot W_1$ and $W_2^1 = 2 \cdot W_2 = 4 \cdot W_1$ for the VI AC, which lead to just three possible contention windows. This has a relevant consequence, since, once a given source has established its retry limit, the retry limits selected by the other sources have a limited influence on the collision probability of that source, because, after the first transmission attempt, the contention window remains identical for all the subsequent attempts. To better clarify this issue, consider, as a theoretical reference, the limiting case in which just the VI AC is active and the backoff stage is equal to zero. This is a perfectly homogeneous case, since, regardless of the retry limit value selected by each source, the backoff time is randomly selected in an identical interval. Hence, the collision probability of the single source is insensitive to the retry limits selected by the other sources. The unity backoff stage, even if not guarantees this complete insensitivity, however maintains the sensitivity very low, thus justifying the implicit assumption adopted in the formulation developed in (4.9), according to which each source selects its m_2 value considering that the other sources adopt the same one. Furthermore, to deepen this issue, it have been performed a set of simulations to include the heterogeneous case in which each source may adopt its own setting for the retry limit, where S sources has to access the channel using the EDCAF. In particular, it have been considered a network with $N = 4$ source-destination pairs, where the collision probability $p_2^{(1)}$ for a video packet of the source 1 has been monitored. The retry limit $m_2^{(1)}$ of the source 1 (in general the notation $m_2^{(j)}$ denotes the retry limit for the node j) is fixed at a value equal to 32. The retry limits of the other three sources are selected considering all the possible combinations generated by the set $[2^0, 2^3, 2^6] = [1, 8, 64]$. These combinations (reported in the rows from 2 to

Case	1	2	3	4	5	6	7	8	9	10
$m_2^{(2)}$	1	1	1	1	1	1	8	8	8	64
$m_2^{(3)}$	1	1	1	8	8	64	8	8	64	64
$m_2^{(4)}$	1	8	64	8	64	64	8	64	64	64
$p_2^{(1)}$ (VI, $m_2'=0$)	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.51
$p_2^{(1)}$ (VO,VI, $m_{1,2}'=1$)	0.81	0.80	0.80	0.80	0.79	0.79	0.79	0.79	0.79	0.79
$p_2^{(1)}$ (VO,VI, $m_{1,2}'=5$)	0.62	0.59	0.60	0.56	0.58	0.57	0.54	0.54	0.53	0.53

TABLE 4.1: Different combinations of retry limits and corresponding simulated collision probabilities for a video packet transmitted by source 1 in a heterogeneous scenario with $N = 4$ sources.

4 of Table 4.1 lead to ten possible cases. For example, in the case 8, the set of retry limits adopted by the four sources is $[32, 8, 8, 64]$. The effect of each combination of retry limits on the collision probability of a video packet of the first source is studied in three scenarios. In the first scenario the sole VI AC is active and the maximum backoff stage is equal to zero, thus, regardless of the retry limit, the contention window remains fixed and equal to its minimum value for the entire simulation. This first scenario is substantially homogeneous, and, as expected, provides identical collision probabilities in all cases (row 5 of the table). In the second scenario the VO and VI ACs are active and the maximum backoff stage is equal to one. This scenario coincides with the 802.11e standard settings, and the corresponding results reveal that the sensitivity of the collision probability $p_2^{(1)}$ on the retry limits of the other sources is very low (row 6 of the table). In the third scenario the VO and VI ACs are active and the maximum backoff stage is equal to five. Now, the sensitivity of $p_2^{(1)}$ on the retry limits of the other sources increases (row 7 of the table). The results in the table seem to suggest that the sensitivity of the single-source performance on the retry limits selected by the other sources becomes higher when the number of possible contention windows increases, that is, when the maximum backoff stage increases. In the 802.11e standard settings, this value is maintained very low for the VO and VI ACs, in order to avoid too large backoff times that would be in contradiction with the delay constraint that usually characterizes

the audio/video traffic. In summary, these simulations suggest that the low maximum backoff stage specified in the 802.11e amendment for the VO and VI ACs may be the main reason that allows one to assume, on first approximation, a negligible sensitivity of the single-source performance on the retry limits selected by the other sources. Additionally, the reliability of this hypothesis will be further explored in Subsection 4.4.2.4, where a network scenario involving sources transmitting different video sequences, and hence adopting different retry limits, will be considered.

Exploiting the above arguments, whose reliability will be also investigated in Section 4.4 considering scenarios in which all the four ACs are active, (4.9) can be replaced by a reduced nonlinear system of four equations:

$$\begin{cases} \tau_q = \dfrac{2\left(1-p_q^{m_q+1}\right)}{(2W_q+1)\left(1-p_q^{m_q+1}\right)-W_q(1-p_q)}, \quad q=1,2 \\ p_1 = 1-(1-\tau_1)^{N-1} \\ p_2 = 1-(1-\tau_1)^{N}(1-\tau_2)^{N-1} \end{cases} \tag{4.10}$$

Then, in the first two equations of (4.10), τ_q may be approximated by a second-order polynomial passing through the points $(0,\tau_{q_1})$, $(1/2,\tau_{q_2})$, and $(1,\tau_{q_3})$, where:

$$\tau_{q_1} = \tau_q\big|_{p_q=0} = \frac{2}{W_q+1}, \tag{4.11a}$$

(4.11b)

$$\tau_{q_2} = \tau_q\big|_{p_q=1/2} = \frac{2^{m_q}\cdot 4-2}{2^{m_q}(2+3W_q)-2W_q-1} \underset{m_q\to\infty}{\longrightarrow} \frac{4}{3W_q+2}, \tag{4.11c}$$

(4.11d)

$$\tau_{q_3} = \lim_{p_q\to 1}\tau_q = \frac{2m_q+2}{m_q(2W_q+1)+W_q+1} \underset{m_q\to\infty}{\longrightarrow} \frac{2}{2W_q+1}, \tag{4.11e}$$

are evaluated for $m_q \to \infty$ when a dependence on m_q is present. Thus, τ_q can be approximated by:

$$\tau_q \cong a_q p_q^2 + b_q p_q + c_q, \tag{4.12}$$

where the coefficients:

$$a_q = \frac{4W_q^2}{6W_q^3 + 13W_q^2 + 9W_q + 2}, \tag{4.13a}$$

$$b_q = -\frac{2W_q(5W_q + 2)}{6W_q^3 + 13W_q^2 + 9W_q + 2}, \tag{4.13b}$$

$$c_q = \frac{2}{W_q + 1}, \tag{4.13c}$$

depend only on the minimum contention window of the q-th AC. These coefficients are obtained by fitting (4.12) through the three points $(0, \tau_{q1})$, $(1/2, \tau_{q2})$, and $(1, \tau_{q3})$, that is, substituting the p_q and τ_q coordinates in (4.12) for each point, and then solving the resulting linear system of three equations in the three unknowns a_q, b_q, and c_q. Concerning the approximations in (4.11) and (4.12) for the evaluation of τ_q, it is worth remarking that the aim of the proposed strategy is to estimate, first, the drop probability and the delay for the VI AC through p_2, and, subsequently, the retry limit according to the video distortion and the expiration time. Solving directly (4.10) would lead to a p_2 value dependent on m_2 and hence to a higher computational cost. Instead, the solution of a polynomial equation, obtained removing the dependence on m_2, can rely on efficient root-finding techniques. Moreover, one can easily prove, by performing a simple derivative, that τ_q in (4.10) is a monotonically decreasing function of m_q. Hence, since (4.12) is obtained for $m_q \to \infty$, it provides an estimate from below of τ_q, which leads to an underestimation of p_q and, in turn, of the drop probability $p_q^{m_q+1}$. This behavior, combined with the requirement that the drop probability of a packet be inversely proportional to the distortion introduced by its loss, guarantees a small (conservative) overestimation of the distortion's effect, thus explaining the reason for the use of an approximation based on an infinite retry limit in (4.11c) and (4.11e).

Given $\tau_q < 1$, each term $(1 - \tau_q)^{N-1}$ in the two latter equations of (4.10) can be approximated by truncating the corresponding binomial expansion to a suitable value $\bar{N}(\leq N - 1)$. Using this approximation and substituting (4.12) for $q = 1$ in the third equation of (4.10), one obtains the polynomial equation:

$$\sum_{j=0}^{\bar{N}} \begin{pmatrix} \bar{N} \\ j \end{pmatrix} (-1)^j (a_1 p_1^2 + b_1 p_1 + c_1)^j + p_1 - 1 = 0, \tag{4.14}$$

whose solution $\bar{p}_1 \in [0,1]$ is the approximated conditional collision probability for the VO AC. Similarly, substituting (4.12) for $q = 2$ in the fourth equation of (4.10), and defining $\bar{\tau}_1 = a_1\bar{p}_1^2 + b_1\bar{p}_1 + c_1$, one can use the binomial approximation to derive a second polynomial equation:

$$(1-\bar{\tau}_1)^N \cdot \sum_{j=0}^{\bar{N}} \begin{pmatrix} \bar{N} \\ j \end{pmatrix} (-1)^j (a_2 p_2^2 + b_2 p_2 + c_2)^j + p_2 - 1 = 0, \tag{4.15}$$

which provides, for the VI AC, the approximated conditional collision probability $\bar{p}_2 \in [0,1]$ and subsequently the approximated transmission probability $\bar{\tau}_2 = a_2\bar{p}_2^2 + b_2\bar{p}_2 + c_2$. Thus, the original problem of evaluating the transmission and collision probabilities for the VO and VI ACs by the nonlinear system in (4.9) is considerably simplified by the replacement with the two polynomial equations in (4.14) and (4.15). Observe that this simplification may be maintained even if the number of nodes is high. In fact, the degree of the polynomial in (4.14) and (4.15) depends of the parameter $\bar{N}$, which may be selected lower than $N-1$ maintaining an acceptable accuracy, since $\bar{\tau}_q^j$ becomes less significant as j becomes larger. Furthermore, it is worth noticing that the current 802.11 PHY layer extensions lead to very low minimum and maximum contention windows for the VO and VI ACs. Hence, the number of collisions grows rapidly with the increase of N, thus making difficult the support of many contending video flows of acceptable quality. By consequence, the selection of the maximum value $\bar{N} = N-1$ in (4.14) and (4.15), which provides the highest accuracy, may be acceptable in practical scenarios, where, realistically, the number of contending video flows is limited.

Once the four probabilities $\bar{p}_q, \bar{\tau}_q$ for $q = 1, 2$ are estimated, one can derive the performance figures of the network as a function of the retry limit. Remembering that $m_2' = 1$ and introducing the notation $m_{2,k}$ to identify the dependence of the retry limit for the VI AC on the generic video packet $\pi_k \in \mathcal{P}$, the average packet delay can be expressed as [110]:

$$T(m_{2,k}) = \mathrm{E_s} \cdot \mathrm{E_{ns}}(m_{2,k}), \tag{4.16}$$

where $\mathrm{E_s}$ is the average time required for a decrease of the backoff counter and:

$$\mathrm{E_{ns}}(m_{2,k}) = \sum_{i=0}^{m_{2,k}} \frac{W_2^i - 1}{2} \bar{p}_2^i = \frac{2W_2 - 1}{2} \cdot \frac{1 - \bar{p}_2^{m_{2,k}+1}}{1 - \bar{p}_2} - \frac{W_2}{2}, \tag{4.17}$$

is the average number of backoff decreases for the $m_{2,k}$ retransmissions of π_k. Recalling that, using the basic access, the transmission time $\bar{T}$ for a success and a collision is the same, $\mathrm{E_s}$ is given by the sum of the fractions of time wasted because of inactivity and used for transmission (successful or not), thus:

$$\mathrm{E_s} = \varsigma + \left\{1 - [(1 - \bar{\tau}_1)(1 - \bar{\tau}_2)]^N\right\} (\bar{T} - \varsigma), \tag{4.18}$$

where ς is the slot time specified by the adopted PHY layer standard and the term $1 - [(1 - \bar{\tau}_1)(1 - \bar{\tau}_2)]^N$ denotes the probability that at least one packet is transmitted [110]. Furthermore, the transmission time $\bar{T}$ in (4.18) can be calculated as:

$$\bar{T} = \frac{\bar{\Lambda}}{R} + \frac{\mathrm{H} + \mathrm{ACK}}{R_\mathrm{c}} + \mathrm{SIFS} + \mathrm{AIFS}_2, \tag{4.19}$$

where $\bar{\Lambda}$ is the length of the payload averaged over the VO and VI ACs, R is the data rate, H is the length of the MAC/PHY headers of the DATA packet, ACK is the length of the ACK packet, R_c is the control rate, SIFS is the short inter-frame space, and $\mathrm{AIFS}_2 = \mathrm{SIFS} + 2\varsigma$ [85]. The second fundamental performance figure that is required to estimate the network behavior when a video sequence has to be transmitted is the drop probability, which can be evaluated as:

$$p_{\mathrm{drop}}(m_{2,k}) = \bar{p}_2^{\,m_{2,k}+1}. \tag{4.20}$$

Now, the problem's requirements can be imposed by relating the drop probability to the distortion and the delay to the expiration time. In particular, the packets that lead to a higher distortion in the case of loss should be associated to a lower drop probability and hence to a higher retry limit, while the packets determining a lower distortion should be associated to a higher drop probability and hence to a lower retry limit. Moreover, the distortion should be not only inversely proportional to the drop probability, but should recall the exponential relationship in (4.20) between p_{drop} and $m_{2,k}$, in order to provide an effective adaptation [142]. Therefore, imposing as a further requirement that the sum of the delays derived by (4.16)-(4.18) for the first k packets be lower than the expiration time of the k-th packet, the following minimization problem in the unknown $m_{2,k}$ can be formulated for each $\pi_k \in \mathcal{P}$:

$$\underset{m_{2,k}\in\mathbb{N}}{\arg\ \min} \left| p_{\mathrm{drop}}(m_{2,k}) - 10^{-\zeta D_k} \right|, \tag{4.21}$$

$$\text{subject to}: \sum_{k'=1}^{k} T(m_{2,k'}) \leq T_{\mathrm{e}_k}, \tag{4.22}$$

where $\zeta (> 0)$ is a parameter introduced to better manage the relationship between drop probability and distortion, whose impact on the estimation process will be discussed at the beginning of the next section. The objective of the problem in (4.21) and (4.22) is to find, for each $\pi_k \in \mathcal{P}$ from $k = 1$ to $k = K$, the retry limit $m_{2,k}$ that provides the drop probability closest to the reciprocal of the exponentially weighted distortion, simultaneously verifying that the reception time remains lower than the expiration time. One of the main advantages of this formulation relies on the possibility to obtain a closed-form expression for the estimated retry limits, thus considerably limiting the computational burden. To achieve this result, (4.21) and (4.22) may be separately solved. More precisely, one may first use (4.20) in (4.21), and then solve the corresponding equation in the integer retry limit recalling that a conservative overestimation of the distortion has been adopted, thus obtaining the quantity:

$$m_{2,k}^{D} = \left\lceil \frac{\log(10^{\zeta D_k}\,\bar{p}_2)}{\log(1/\bar{p}_2)} \right\rceil, \tag{4.23}$$

which accounts for the sole distortion. Subsequently, since the retry limits are evaluated following the order $k = 1, ..., K$, (4.22) can be usefully rewritten as:

$$T(m_{2,k}) \leq T_{\mathrm{e}_k} - T_{\mathrm{a}_{k-1}}, \tag{4.24}$$

where, using (4.16) and (4.17), the delay accumulated by the $k-1$ packets previous to the k-th one can be expressed as:

$$T_{\mathrm{a}_{k-1}} = \sum_{k'=1}^{k-1} T(m_{2,k'}) = (k-1)\hat{T} - \left(\hat{T} + \frac{\mathrm{E_s} W_2}{2}\right) \sum_{k'=1}^{k-1} \bar{p}_2^{\,m_{2,k'}+1}, \tag{4.25}$$

with:

$$\hat{T} = \frac{\mathrm{E_s}}{2}\left(\frac{2W_2 - 1}{1 - \bar{p}_2} - W_2\right). \tag{4.26}$$

The novel formulation in (4.24) for the time requirement in (4.22) allows the exploitation of the knowledge of the retry limits already evaluated for $k' < k$. Now, using (4.16) and (4.17) in (4.24), one can solve the corresponding inequality in the integer retry limit that guarantees the satisfaction of the requirement on the expiration time, thus identifying the limiting value:

$$m_{2,k}^{T} = \left\lfloor \log\left[\frac{\left(\hat{T} - T_{\mathrm{e}_k} + T_{\mathrm{a}_{k-1}}\right)^{+}}{\bar{p}_2(\hat{T} + \mathrm{E_s} W_2/2)}\right] \cdot \frac{1}{\log \bar{p}_2} \right\rfloor, \tag{4.27}$$

where $(\cdot)^{+}$ is the positive part and $\lfloor \cdot \rfloor$ is the floor function. In particular, (4.27) selects the largest integer that allows the maintenance of the reception time of the k-th packet below its expiration time. The positive part is introduced for mathematical purposes to include in a unique expression also the cases in which the term $\hat{T} - T_{\mathrm{e}_k} + T_{\mathrm{a}_{k-1}}$ is negative, and hence no requirement on the delay is present in practice. The absence of a delay requirement characterizes, for example, the packets corresponding to the frames indexed from 1 to $\bar{l}$, which, as explained in Subsection 4.2.2, are associated to an infinite expiration time, since the play of the video starts after the elaboration of the frame $f_{\bar{l}}$. Finally, $m_{2,k}$ can be evaluated by taking the minimum between the value in (4.23), accounting for the video distortion, and that in (4.27), accounting for the expiration time, hence obtaining:

$$m_{2,k} = \min\left(m_{2,k}^{D}, m_{2,k}^{T}\right). \tag{4.28}$$

4.3.3 Summary and remarks

The presented mathematical derivation allows the development of a very fast retry limit adaptation algorithm, which requires just a limited number of operations. These operations are summarized in Fig. 4.3. Firstly, one evaluates (in order): $\bar{p}_1$ by solving (4.14), $\bar{\tau}_1$ by (4.12) and (4.13) for $q = 1$, $\bar{p}_2$ by solving (4.15), $\bar{\tau}_2$ by (4.12) and (4.13) for $q = 2$, E_{s} by (4.18), and finally $\hat{T}$ by (4.26). Observe that all these quantities do not depend on the video packet, thus they can be calculated just once for the entire stream, and, if desired, they might be inserted in a lookup table to avoid their re-calculation when the source has to manage different videos in the same network scenario. Secondly, for each $\pi_k \in \mathcal{P}$ and using the estimations D_k and T_{e_k}, one evaluates (in order): $m_{2,k}^{D}$ by (4.23), $T_{\mathrm{a}_{k-1}}$ by (4.25), $m_{2,k}^{T}$ by (4.27), and finally $m_{2,k}$ by (4.28). In general, the algorithm requires just the solution of two polynomial equations and the evaluation of expressions available in closed-form.

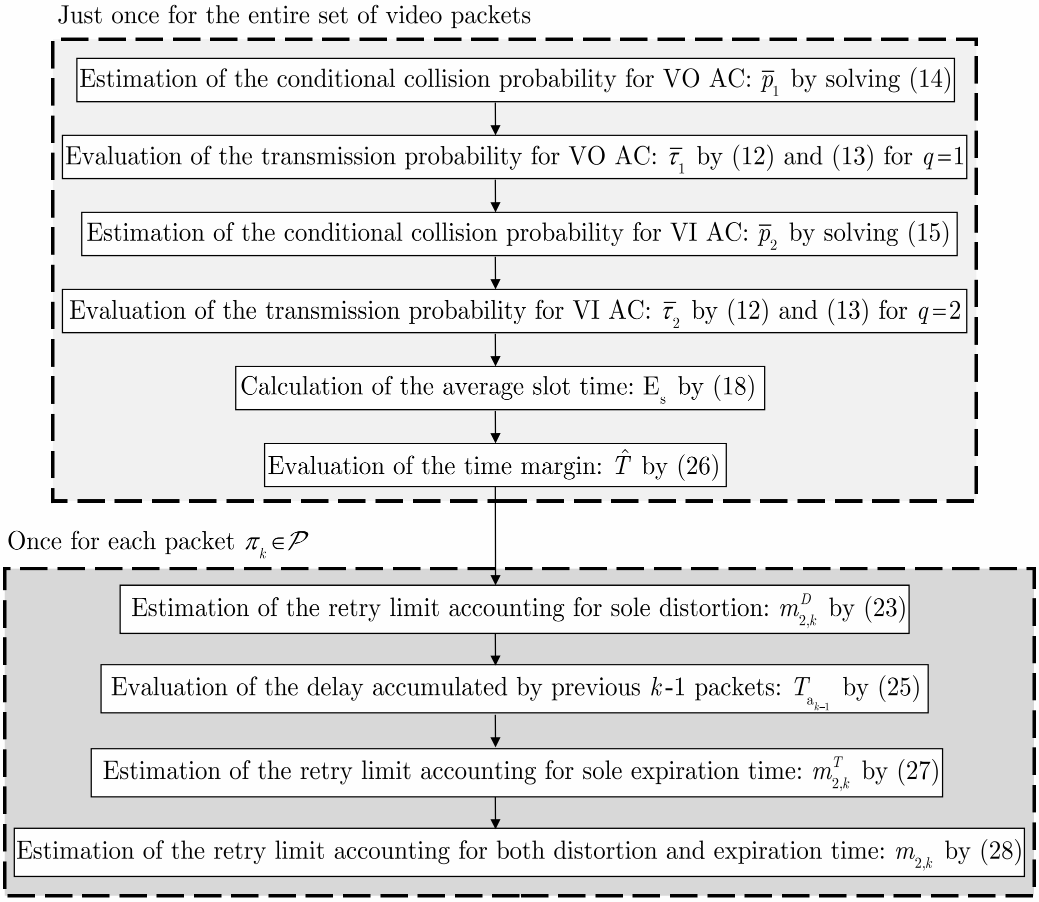
Just once for the entire set of video packets
Estimation of the conditional collision probability for VO AC: $\overline{p}_1$ by solving (14)
Evaluation of the transmission probability for VO AC: $\overline{\tau}_1$ by (12) and (13) for q=1
Estimation of the conditional collision probability for VI AC: $\overline{p}_2$ by solving (15)
Evaluation of the transmission probability for VI AC: $\overline{\tau}_2$ by (12) and (13) for q=2
Calculation of the average slot time: E_s by (18)
Evaluation of the time margin: $\hat{T}$ by (26)
Once for each packet $\pi_k \in \mathcal{P}$
Estimation of the retry limit accounting for sole distortion: $m^D_{2,k}$ by (23)
Evaluation of the delay accumulated by previous k-1 packets: $T_{\mathrm{a}_{k-1}}$ by (25)
Estimation of the retry limit accounting for sole expiration time: $m^T_{2,k}$ by (27)
Estimation of the retry limit accounting for both distortion and expiration time: $m_{2,k}$ by (28)

FIGURE 4.3: Retry limit adaptation algorithm.

It may be useful to observe that the entire framework proposed in this chapter, consisting of the estimation process for the video distortion and the subsequent retry limit adaptation, may be viewed as a modular procedure, in the sense that the retransmission strategy could be also exploited using different measures of the distortion, if desired. In fact, one may notice that (4.23) requires the D_k value, but is not constrained on how this normalized value is obtained. In this chapter, D_k has been derived using the EDA to maintain an overall low computational cost. However, the procedure for solving the problem in (4.21) and (4.22), and hence the steps of the proposed algorithm, remain identical if a different estimation of the video distortion is adopted. This implies that the proposed retry limit adaptation may be applied not only to distortions obtained using different estimation techniques, but also in the presence of video encoders different

H.264/SVC (*Bus* sequence)		**Adaptation algorithm** (Basic access; 802.11g)			
Number of frames	$L=65$	Distortion parameter in (4.21)	$\zeta=3$	Slot time	$\varsigma=20\,\mu$s
				SIFS	SIFS $=$ $10\,\mu$s
GOP size	$\alpha=16$	EDA parameter	$\xi=1/6$	AIFS numbers	AIFSN$_{1,2}=$ 2, AIFSN$_3$ $=$ 3, AIFSN$_4=7$
Expiration time index	$\bar{l}=17$	Polynomial degrees in (4.14)-(4.15)	$2\bar{N}=2(N-1)$		
Inter-frame interval	$T_\mathrm{f}=1/15$ s	Minimum contention windows	$4W_1 = 2W_2 = W_{3,4}=16$	MAC/PHY header length	H $=24$ bytes
Number of layers	$V=4$	Maximum backoff stages	$m'_{1,2}=1,$	ACK length	ACK $=$ 14 bytes
			$m'_{3,4}=6$	Data rate	R $=$ 54 Mbits/s
Payload sizes	$\Lambda_{1,2,3,4}$ $=$ 1400 bytes	Default retry limits	$m_{1,2,3,4}=7$	Control rate	R_c $=$ 2 Mbits/s

TABLE 4.2: Adopted parameters.

from the H.264/SVC, provided that a normalized measure of the distortion is available in some way. Furthermore, just the delay estimation and the retry limit adaptation result necessary if a provider of video services supplies an estimation of the distortion together with the video sequence. The applications enabled by this second possibility, which are not limited to the sole MAC layer, are discussed in [6, 51], and may also involve the routing and the transport layers. Similar arguments may hold in the presence of traffic with time constraints much more stringent than those assumed to derive (4.1) and then (4.3). In fact, for such kind of traffic, (4.27) still holds and provides a value that follows the possible stringent requirement specified by T_{e_k}. As shown in [113, 114], the VI AC may be selected not only to deliver video sequences, but, more in general, to manage the access of other kinds of traffic, including multimedia contents. Thus, the proposed adaptive algorithm may be applied to any traffic that a network operator decides to associate with the VI AC, provided that a measure of the normalized distortion D_k and of the time constraint T_{e_k} are available for each packet.

4.4 Results and discussion

This section evaluates the performance obtained from the proposed method. The results are derived using the parameters in Table 4.2 by assuming that the video playback begins after the elaboration at the receiver of at least the first GOP, thus $\bar{l} = 1 + \alpha = 17$ (the first frame is the frame I). Each AIFS value can be derived as $\text{AIFS}_q = \text{SIFS} + \text{AIFSN}_q\varsigma$, where AIFSN_q denotes the AIFS number for the q-th AC. In the following of this section, when one of the parameters in Table 4.2 will be set to a different value, for example to specifically study its impact on the performance of the developed algorithm, it will be explicitly declared. The transmission buffer of the VI AC of each source is assumed to be sufficiently large to contain all the packets belonging to a given video sequence, in order to avoid losses due to queue overflow, whose modeling is out of the scope of this chapter. All routines and the developed 802.11e network simulator are implemented in Matlab. The presented numerical values are obtained using one core of an Intel Core2 Quad Q9300 @2.50 GHz Sun Ultra 24 workstation.

4.4.1 Estimated retry limits

To provide a clarification of the behavior of the proposed adaptation algorithm, Fig. 4.4 reports, for each frame of the adopted video sequence (Bus), the distortion in (4.2) evaluated by the EDA and the corresponding retry limits for two values of the parameter ζ in the presence of $N = 4$ sources when both the VO and VI ACs are active. In this case no requirements on the expiration time are imposed in order to better outline how the distortion is managed. Consider first the case $\zeta = 3$ (circle marker). For this case the figure shows that the estimated retry limit accurately follows the curve representing the distortion (dashed line), thus revealing that the approximations adopted in (4.3.2) to develop the algorithm by reducing the complexity of the calculations have a very limited impact on the capability of the algorithm to reliably account for the distortion. A more

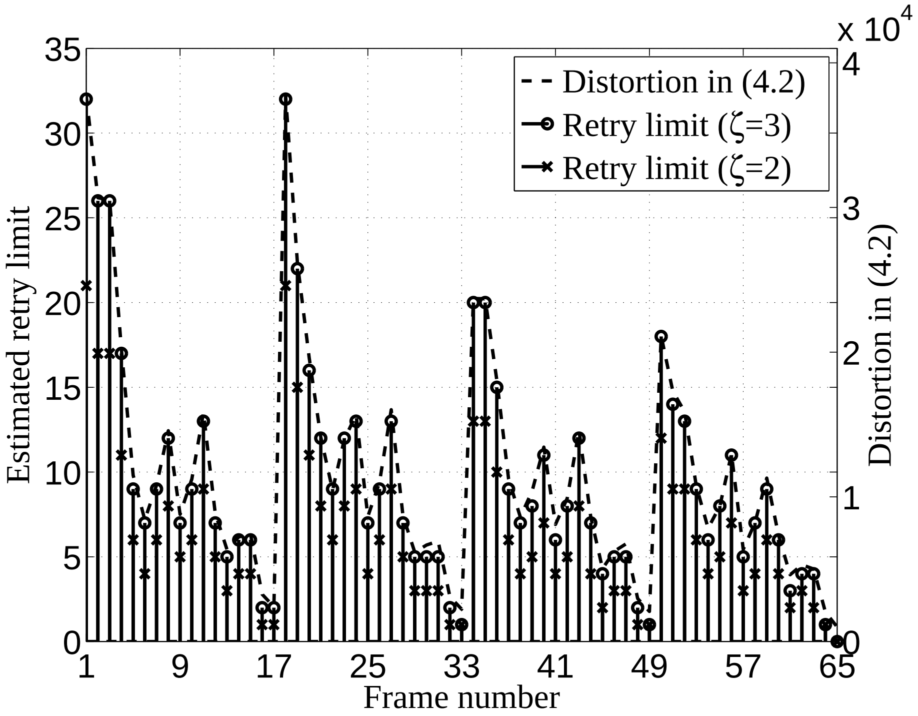

FIGURE 4.4: Retry limits estimated for each frame according to the distortion provided by the EDA for $N = 4$ sources when the VO and VI ACs are both active.

detailed view of this figure, involving even the case $\zeta = 2$ (cross marker), additionally reveals that also the set of retry limits obtained using $\zeta = 2$ is shaped according to the distortion, but at a different scale. This difference between the two sets of retry limits is useful to understand the impact of the parameter ζ, which in practice can be used to control the overall number of retry limits associated to the sequence in absence of requirements on the expiration time. When these requirements on the delay are instead present, they may reduce the retry limits estimated according to the sole distortion, as it can be inferred from (4.28).

A final aspect that may be observed from Fig. 4.4 concerns the high distortion values, and the corresponding high retry limits, that may be noticed for some specific indexes. These indexes identify the I frame, which contains the fundamental encoder settings, and the P frames, which represent the most important frames for the respective GOPs. In fact, all B frames of a given GOP depend on the P frame of that GOP. Thus, while the loss of a B frame has an impact on just a subset of the other B frames of the same

GOP, the loss of a P frame has an impact on all the B frames of the same GOP, and hence the loss of a P frame usually results highly detrimental. This damage is reliably managed by the proposed retransmission strategy, since, for a P frame, the estimated distortion and the corresponding retry limit are both high. Of course, possible stringent requirements on the delay may reduce the retry limit also for a P frame.

4.4.2 Network simulations

Now that the basic behavior of the proposed algorithm has been introduced, the subsequent results aim to further test its performance in a distributed network. Each test is carried out by running 20 network simulations for each considered network scenario, namely for each combination of active ACs and number of contending sources N. Besides, the single network simulation is run for 10 seconds in order to complete the access procedure for all video packets of all sequences. Each simulation has been carried out at packet-level, that is, the retry limits, once obtained, are used in a packet-level 802.11e/g simulator, which is implemented in Matlab as a state machine. Then, for each simulation, the trace corresponding to the correctly received packets is used to derive the drop probability and the reception time, and is physically elaborated by the H.264/SVC decoder to derive the PSNR by comparing the transmitted video $\mathcal{V}_\mathrm{t}$ with the received one $\mathcal{V}_\mathrm{r}$.

4.4.2.1 Preliminary results: two ACs

Figs. 4.5 and 4.6, which are obtained for $N = 4$ and $N = 8$ when the VO and VI ACs are active, compares, for each frame of the video sequence, the PSNR (Fig. 4.5) and the playback reception time (Fig. 4.6) obtained using the proposed algorithm with those derived using the 802.11e/g default settings. The vertical bars present on each curve represent the 95% confidence intervals, which are reported at steps of five frames to

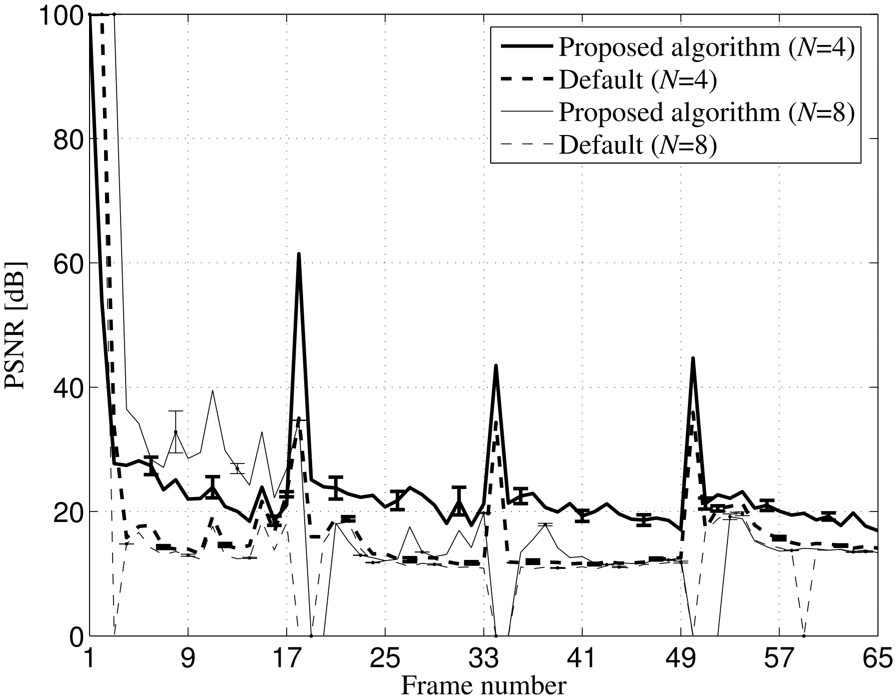

FIGURE 4.5: PSNR for $N = 4$ and $N = 8$ when the VO and VI ACs are active using the proposed algorithm and the default settings.

maintain the readability of the figures. The playback reception time is evaluated starting from the frame $\bar{l}+1 = 18$, for which the requirement on the expiration time becomes finite. Therefore, for a frame f_l with $l > \bar{l}$, $T_{\rm rx}$ is given by the difference between the reception time of the frame f_l and the reception time of the frame $f_{\bar{l}}$, both obtained from the packet-level simulation. This representation, which adopts the instant of starting of the playback as reference, is in agreement with the requirement expressed by the expiration time, and is suitable to verify if, once the playback of the video is started (after the reception of the frame number $\bar{l} = 17$), the reproduction would be interrupted or not. Each curve is derived by averaging the corresponding quantity (PSNR or playback reception time) over all the simulations and the sources. The figure shows that the presented algorithm is preferable to the default settings for both values of N. In particular, the default settings achieve acceptable results for $N = 4$, while, when the number of contending sources increases to $N = 8$, the playback reception time largely exceeds the expiration time. The closeness between the curve corresponding to the expiration

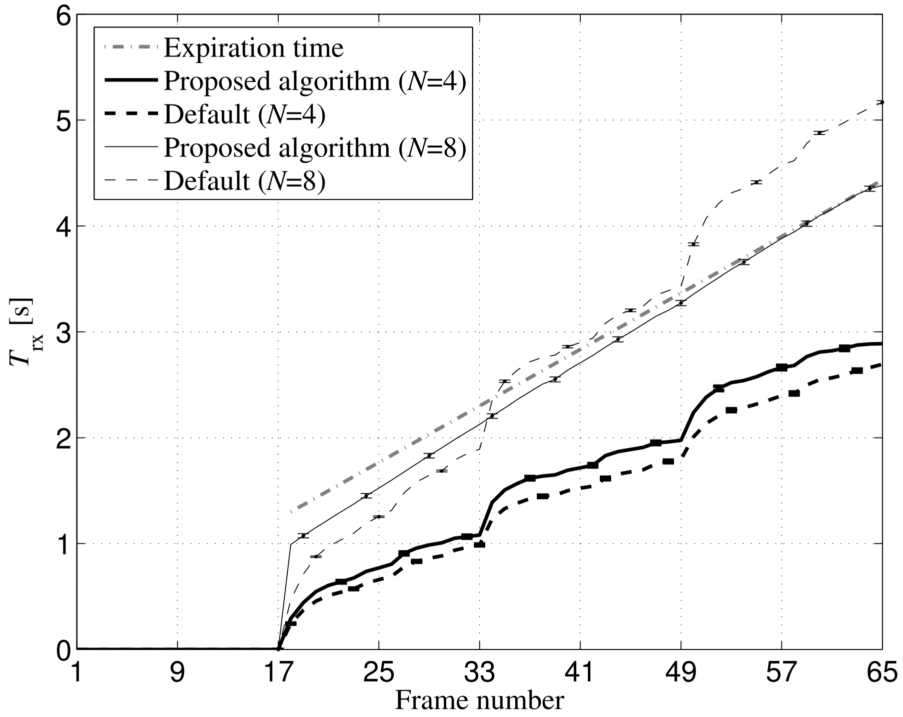

FIGURE 4.6: Playback reception time for $N = 4$ and $N = 8$ when the VO and VI ACs are active using the proposed algorithm and the default settings.

time and the curve corresponding to the playback reception time for the case $N = 8$ when the proposed algorithm is used represents an interesting confirmation that the time available for video transmission is efficiently exploited by the presented retransmission strategy. Recalling that the retry limit estimation process has been developed by operating on average quantities and that the curves are obtained by averaging the results over the simulations and the nodes, one may expect, as a proof of the correctness of the analysis, that, when the scenario becomes highly congested, the playback reception time approaches the expiration time for the last frames of the video sequence. From the provided confidence intervals, one may also observe that, in a single simulation, the playback reception time may be even slightly higher or slightly lower than this average value, exactly because the developed analysis is based on a predictive approach for estimating the evolution of the transmissions.

One may notice that, when $N = 4$, high PSNR values are achieved for the P frames.

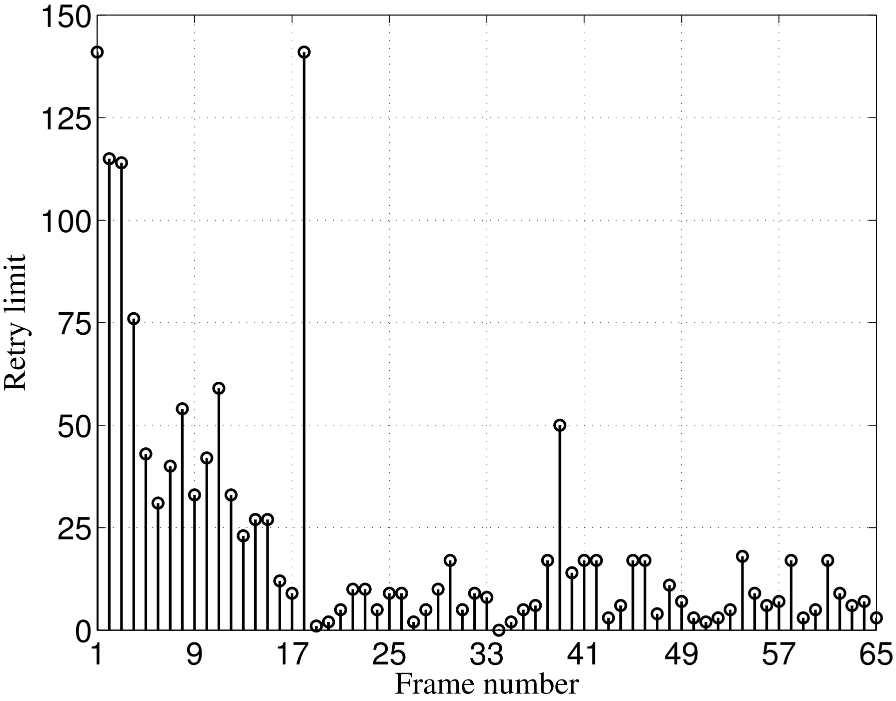

FIGURE 4.7: Estimated retry limits for $N = 8$ sources when the VO and VI ACs are both active.

This may be again explained recalling the H.264 encoding structure. As previously discussed, all B frames of a GOP depend on the P frame of that GOP. On the other hand, a P frame does not depend on these B frames, and hence it is immune to their loss. This leads to a situation in which, once a P frame is received, the PSNR for that P frame may be very high. Differently, the achievement of high PSNR values for a B frame requires not only the correct reception of that B frame, but also the correct reception of all the B and P frames from which that B frame depends. Thus, the presence of a high PSNR for a B frame is an event less frequent than the presence of a high PSNR for a P frame, which, being more important (i.e., associated to a higher distortion), is protected by a high retry limit.

Concerning the scenario corresponding to $N = 8$, it is worth remarking, firstly, that the contention does not only involve 8 VI ACs, but also 8 higher priority VO ACs, and, secondly, that the minimum contention windows and the maximum backoff stages established by the 802.11e/g standard for the VO and VI ACs are very low (Table 4.2).

This leads to a very constrained scenario, in which the proposed algorithm remains able to guarantee an acceptable video playing within the expiration time in the presence of a necessarily high number of collisions. In these kinds of scenarios, the PSNR referred to some frames may drop to very low values. To this purpose, it is worth to remark that the objective of the proposed algorithm is not to ensure that all frames are received, but to ensure that, in the presence of distortion/delay requirements and contention-based mechanisms, the highest possible quality level (in those network conditions) is achieved for the overall video sequence. To provide more details on the behavior of the proposed adaptation mechanism, Fig. 4.7 presents the estimated retry limits for the scenario with $N = 8$ sources. Considering this figure together with Fig. 4.4, one may observe that the retry limits of the frames having index lower than $\bar{l} + 1 = 18$ remain shaped according to the distortion, since the expiration time is infinite for $l < 18$, but they become much higher than those corresponding to the case $N = 4$, because for $N = 8$ the collision probability considerably increases and hence more retransmissions are statistically necessary. For $l \geq 18$, instead, the retry limits are no more exactly shaped according to the distortion, since the requirement on the expiration time becomes dominant.

4.4.2.2 General results: two and four ACs

While the previous results have shown that the proposed algorithm is able to operate in the presence of many contending flows, a second set of simulations is carried out to deepen some further aspects, which have been fundamental during the development of the method. These aspects concern the performance of the proposed algorithm when all the four ACs are active. The aim is to investigate if the approximation in (4.10) of the system in (4.9) is acceptable, simultaneously evaluating the computational time required to estimate the retry limit. The results of this second set of simulations are presented in Table 4.3, which reports the drop probability and the PSNR (both averaged over the simulations, the sources, and the frames), the maximum playback reception

N	Q	$\bar{p}_{\text{drop}}$ [%]			$\overline{\text{PSNR}}$ [dB]			$T_{\text{rx}_{\max}}$ [s]			$\bar{S}$ [Mbits/s]			CPU time [s]	
		Def	Opt	Alg	Def	Opt	Alg	Def	Opt	Alg	Def	Opt	Alg	Opt	Alg
4	2	40.8	30.9	29.7	18.5	24.2	24.4	2.7	2.9	2.9	0.5	0.8	0.8	65.1	1.1
	4	39.0	29.6	30.3	18.6	24.2	25.4	2.7	2.9	2.9	0.5	0.8	0.8	110.9	1.2
6	2	72.0	41.9	42.4	18.6	22.6	22.5	4.1	4.6	4.4	0.2	0.3	0.3	136.4	1.4
	4	71.8	42.5	43.0	17.1	22.0	21.9	4.1	4.3	4.4	0.2	0.3	0.3	228.9	1.4
8	2	88.8	65.9	65.2	16.5	21.7	22.7	5.2	4.3	4.4	0.1	0.2	0.2	226.7	1.3
	4	88.7	65.5	65.9	16.4	22.9	21.4	5.2	4.2	4.4	0.1	0.2	0.2	393.9	1.4
10	2	94.3	74.9	73.9	16.4	27.0	27.8	5.9	4.2	4.3	0.0	0.1	0.1	406.7	1.4
	4	94.1	73.9	74.5	16.7	29.6	25.9	5.9	4.1	4.3	0.0	0.1	0.1	792.6	1.5

TABLE 4.3: Drop probability, PSNR, maximum playback reception time, single-node throughput of the VI AC, and processing time for different retry limit setting policies: default (Def), optimum (Opt), and proposed algorithm (Alg).

time $T_{\text{rx}_{\max}}$, corresponding to the playback reception time of the last frame of the video sequence (averaged over the simulations and the sources), the single-node throughput $\bar{S}$ of the VI AC (averaged over the simulations and the sources), and the central processing unit (CPU) time required by Matlab to estimate all the retry limits of the video sequence. From now on, the symbol Q is used to denote the number of active ACs. In particular, when $Q = 2$, the sole VO and VI ACs are active, while, when $Q = 4$, all the four ACs are active. For a given source and a given simulation, the throughput is evaluated as the ratio between the sum of the bits correctly received by each destination and that may be used for decoding purposes, and the time instant corresponding to the end of the processing of the last packet of the video sequence. This time is the reception time, when the last packet is correctly received, while it is the discarding time, when the last packet is dropped. To have a reliable term of comparison, Table 4.3 includes the performance corresponding to the set of optimum retry limits, which are derived by numerically solving the problem in (4.21) and (4.22) using the complete system in (4.9). In particular, the eight equations in (4.9) and the requirement in (4.21) have been implemented in a unique Matlab function. Then, the resulting system has been solved using the Matlab function `fsolve`, reducing, if necessary, the derived retry limit until the constraint in (4.22) is satisfied.

The results in the table confirm that both the optimum retry limit setting and the proposed algorithm provide a better performance with respect to the default settings. As expected, when the number of sources increases, the drop probability increases. However, in these cases, both the optimum setting and the algorithm are able to provide satisfactory PSNR values. It is furthermore interesting to observe that, for each scenario, the drop probability, the PSNR, the maximum playback reception time, and the throughput obtained from the optimum setting and the developed method are very close to each other. In particular, this closeness holds also when all the four ACs are active, thus confirming the reliability of the approximations adopted during the development of the algorithm. This element becomes more relevant when considered together with the CPU time, since the results in Table 4.3 reveal that less than two seconds are sufficient to evaluate all the retry limits using the proposed algorithm, while some minutes are required by the policy relying on the optimum setting. Concerning this latter aspect, a further noticeable advantage of the presented solution may be signaled. As explained in Subsection 4.3.3, the set of operations performed by the algorithm may be subdivided in two subsets: the subset of the operations required to estimate the network evolution, such as the evaluation of the collision probabilities and of the average slot duration $\mathrm{E_s}$, which can be performed just once for the entire stream, and the subset of the operations required to subsequently estimate the retry limit, which must be performed packet by packet (Fig. 4.3). The CPU times reported in the last column of Table 4.3 are mainly due to the first subset of operations, while the second subset requires considerably lower computational times.

4.4.2.3 Comparison

To further test the proposed solution, the obtained results are compared to those achievable using the retry limit adaptation algorithm presented in [88]. This algorithm relies

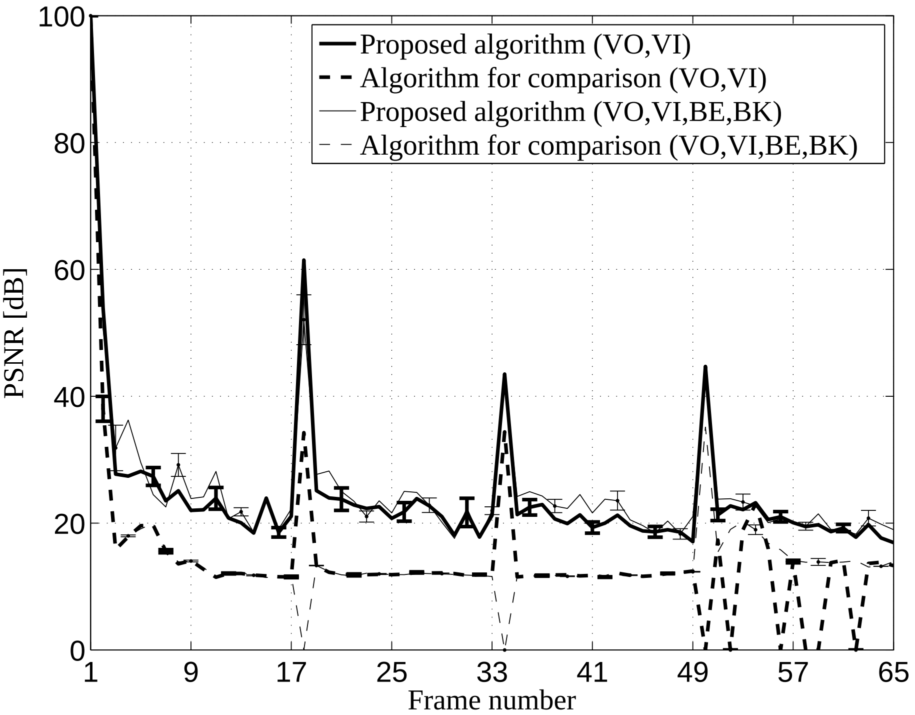

FIGURE 4.8: PSNR for $N = 4$ using the proposed algorithm and the method presented in [88], used for comparison.

on an unequal loss protection approach developed according to the collision probability and to a strategy of differentiation of the packets into groups, in which the most important packets are associated to higher retry limits. The solution in [88] has been selected for its very low computational cost, thus making significant the comparison with the algorithm proposed in this chapter. Since [88] assumes that the collision probability is available and does not consider the problems relative to its estimation, the quantity $\bar{p}_2$, estimated by the proposed algorithm, is used as the collision probability in [88], in order to guarantee a fair comparison between the two algorithms. Furthermore, since [88] does not accounts for possible time constraints, the results concerning the delay are not considered.

Fig. 4.8 shows the PSNR as a function of the frame number obtained for $N = 4$ when two (VO and VI) and four ACs are active, while Table 4.4 reports the average drop probability, the PSNR, the single-node throughput for the VI AC, and the CPU time also for

N	Q	$\bar{p}_{\text{drop}}$ [%]		$\overline{\text{PSNR}}$ [dB]		$\bar{S}$ [Mbits/s]		CPU time [s]	
		Alg	Alg [88]	Alg	Alg [88]	Alg	Alg [88]	Alg	Alg [88]
4	2	29.7	76.0	24.4	15.8	0.8	0.2	1.1	1.1
	4	30.3	76.7	25.4	15.4	0.8	0.2	1.2	1.2
6	2	42.4	99.5	22.5	21.5	0.3	0.0	1.4	1.4
	4	43.0	99.5	21.9	18.1	0.3	0.0	1.4	1.4
8	2	65.2	99.7	22.7	-	0.2	0.0	1.3	1.3
	4	65.9	99.7	21.4	-	0.2	0.0	1.4	1.4

TABLE 4.4: Drop probability, PSNR, single-node throughput of the VI AC, and processing time obtained using the proposed algorithm (Alg) and that presented in [88] (Alg [88]).

the network scenarios corresponding to larger values of N. The results confirm the satisfactory performance of the proposed algorithm, which remains capable of sustaining the video traffic even in a highly congested environment. In particular, one may observe from the last two columns of Table 4.4, that the CPU times for the two solutions appear as identical. In practice, differences are present just in the not reported less significant decimals. The similarity is due to the use of the same procedure of estimation for the collision probability, which, as previously discussed, has the larger impact on the computational cost. Thus, once this estimation is available, the remaining calculations have a negligible cost for both compared algorithms. Combining this characteristic with the satisfactory performance achievable by the proposed method, one may conclude that the here presented algorithm is able to provide a really satisfactory tradeoff between performance and complexity.

4.4.2.4 Different video sequences

As a final set of results, Table 4.5 reports, for each source, the performance obtained when $N = 4$ sources transmit different video sequences. The table considers the optimum retry limit setting, the proposed algorithm, and the solution presented in [88].

	Optimum retry limit setting				Proposed algorithm				Algorithm in [88]			
Source	1	2	3	4	1	2	3	4	1	2	3	4
$\bar{p}_{\mathrm{drop}}$ [%]	21.2	11.2	27.2	10.4	19.5	12.4	26.4	9.2	64.2	30.3	34.7	32.0
$\overline{\mathrm{PSNR}}$ [dB]	40.4	86.9	33.2	72.8	36.9	81.9	33.7	76.8	17.6	32.8	24.6	42.6
$T_{\mathrm{rx}_{\max}}$ [s]	2.1	0.4	0.7	0.5	2.0	0.4	0.7	0.4	1.8	0.4	0.6	0.4
$\bar{S}$ [Mbits/s]	1.1	0.9	0.9	0.9	1.1	0.9	0.9	0.9	0.4	0.7	0.8	0.8
CPU time [s]	114.5	33.2	37.1	28.9	1.3	0.2	0.3	0.3	1.3	0.2	0.3	0.3

TABLE 4.5: Drop probability, PSNR, maximum playback reception time, single-node throughput of the VI AC, and processing time when $N = 4$ sources transmit different video sequences (source 1: *Bus*, source 2: *Container*, source 3: *Foreman*, source 4: *News*).

The values are derived allowing that, at each source, all the four ACs are active. Observe that, in some cases, similar throughput values may appear in conjunction with different drop probabilities. In particular, a direct comparison between the proposed algorithm and that presented in [88] for the fourth video sequence (*News*) shows that the two throughput values are close, but the drop probabilities are considerably different. The reason of this behavior may be explained, firstly, remembering that the drop probability is referred to the frames, while the throughput is referred to the packets usable for decodable frames, and, secondly, recalling the characteristics of the two retransmission strategies. In this specific case, the two algorithms allow the reception of a similar number of packets, but the packets received adopting the proposed algorithm enable the decoding of a number of frames that is larger than that enabled by the packets received adopting the algorithm in [88]. This is confirmed by the different PSNR values, and further outlines the importance of adopting a sophisticated estimation of the distortion, and, in turn, of the retry limits.

One may notice from the table that also in this case the developed algorithm provides values for the drop probability, the PSNR, the maximum playback reception time, and the throughput that are very close to those achievable using the optimum setting, simultaneously maintaining a very low CPU time, which remains identical to that required by the method conceived in [88]. The similarity of the performance provided by the optimum setting and the proposed algorithm in this heterogeneous (in terms of videos) scenario confirms the limited impact of the simplifying hypothesis adopted in Subsection 4.3.2, according to which each source selects its retry limit assuming that the other sources select the same one.This latter result, combined with the other validations reported in this section, confirms that the adopted modeling approach represents a suitable solution for reaching the purposes considered at the beginning of this study: the development of a retry limit adaptation strategy that, moving from the initial objective of guaranteeing a low computation cost, be able to satisfy distortion and delay requirements for video transmission over 802.11e distributed networks.

4.5 Conclusions

A fast and simple retry limit adaptation method for video streaming applications over 802.11e distributed networks in the presence of distortion and delay requirements has been presented in this chapter. The method has been derived by carefully modeling the evolution of the network and by introducing proper approximations, whose reliability has been validated by numerical simulations, which have allowed to considerably reduce the computational cost of the conceived solution.

The results have shown that the presented algorithm is able to accurately account for the impact of the higher priority VO AC on the video transmission, while simultaneously maintaining the frame delay below the video expiration time. The satisfactory performance has been reached maintaining a really low processing time for the retry limit estimation process. This latter advantage reveals that the developed algorithm may be also of interest for possible implementations on network devices characterized by very limited computational resources.

Acknowledgements

The work in this chapter has been supported in part by the National Inter-University Consortium for Telecommunications (CNIT) within the project “Multiple access improvements in high-capacity 802.11 networks for video streaming support”, and by the Italian Ministry of University and Research (MIUR) within the project FRA 2013 (University of Trieste, Italy), entitled “Multi-packet communication in 802.11x heterogeneous mobile networks: models and antenna system algorithms.”

Chapter 5

Impact of video quality assessment on retry limit adaptation

Nowadays, the streaming of video content has become one of the most relevant areas in multimedia communications. In North America over 50 percent of the source traffic is generated by streaming of videos from web sites such as Netflix or YouTube, and in Europe video flows represent over 30 percent of the total traffic. The possibility of managing video traffic in 802.11-based wireless networks achieving a high level of acceptability is a challenging task due to the unreliability of the wireless channel and the losses due to the collisions. Packet losses can in fact lead to a relevant quality degradation due to the error propagation effect on the video stream. Thus, as previously seen, to guarantee QoS control at the MAC layer, the 802.11e amendment has been introduced. During the channel contention period, the video packets sent by the different nodes can collide, requiring to adopt a proper policy for their retransmission, with the purpose to provide the best video quality to the end user. Within this context, one may identify two relevant issues. The first issue concerns the selection of

The content of this chapter is based on F. Babich, M. Comisso, and R. Corrado, "On the Impact of the Video Quality Assessment in 802.11e Ad-Hoc Networks Using Adaptive Retransmissions", *IEEE IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Piran, Slovenia, pp. 47 - 54, 2 - 4 June 2014.

the retransmission policy, given an estimation of the distortion suffered by the video because of the loss of a packet. Several studies have been performed on this research area, adopting machine learning techniques [86], unequal loss protection mechanisms [88], and priority queueing [42, 43, 89]. The second relevant issue involves the video quality measurement methodologies on which a given retransmission policy can rely. In particular, in [115] the authors classify the possible methodologies in two categories, subjective quality assessments and objective quality assessments, comparing the main differences between the two approaches. In [116] a survey on the measurement methods for subjective video quality assessments is provided, while in [27] the authors perform a comparison between two of the most common indexes, the widely adopted MSE and the more recently introduced SSIM.

This chapter focuses on the latter of the above discussed issues, aiming to extend the comparison between MSE and SSIM by considering the application of these indexes to the field of 802.11e adaptive video retransmission. Precisely, the study presented in this chapter investigates the suitability of MSE and SSIM to provide, at MAC layer, a reliable estimation of the distortion and, subsequently, of the retry limit in a contention-based scenario. The results are derived using a Matlab simulation platform that models the 802.11e EDCA and is interfaced with the C++ implementation of the H.264 SVC standard. To obtain an exhaustive view of the achieved performance, the quality of each received video is evaluated in terms of PSNR, MSE, SSIM, and cumulative frame delay, regardless of the index adopted to estimate the distortion and hence the retry limit.

The chapter is organized as follow. Section 5.1 introduces the adopted video quality assessments. Section 5.2 describes the single-node and 802.11e network models. Section 5.3 presents and discusses the numerical results and section 5.4 summarizes the most important conclusions.

5.1 Video quality assessments

The study presented in this chapter considers two video quality assessments: the MSE, selected for its simplicity and its common use as a full-reference quality metric, and the SSIM, selected for its better consistence with the human eye perception. In particular, the MSE is computed by averaging the squared intensity differences of distorted and reference frame pixels [35], while the SSIM accounts for the structural information in a frame by considering those attributes that represent the structure of the objects in the scene [35, 117].

5.1.1 Mean Squared Error (MSE)

Consider two video sequences $\mathcal{V}_t = \{f_l \mid l = 1, ..., L\}$ and $\mathcal{V}_t' = \{f_l' \mid l = 1, ..., L\}$, having an identical number of frames L, which have been alredy encoded and decoded. This assumption allows one to not take into consideration the encoding distortion which is not in the interest of this chapter. A comparison between $\mathcal{V}_t$ and $\mathcal{V}_t'$ can be performed by considering each frame as a discrete signal, thus $f_l = \{v_{li} \mid i = 1, ..., Pix\}$ and $f_l' = \{v_{li}' \mid i = 1, ..., Pix\}$, where Pix is the number of signal samples (pixels) and v_{li} and v_{li}' are the values of the i-th sample of the l-th frame. The MSE between f_l and f_l' can be expressed as [27]:

$$\mathrm{MSE}(f_l, f_l') = \frac{1}{Pix}\sum_{i=1}^{Pix}(v_{li} - v_{li}')^2, \tag{5.1}$$

thus considering the difference between the two frames as an error signal $e_l = f_l - f_l'$. Usually, in the video processing literature, the MSE is converted to the PSNR by [27]:

$$\mathrm{PSNR} = 10\log_{10}\frac{(255)^2}{\mathrm{MSE}}, \tag{5.2}$$

where the range of the allowable pixel intensities in this chapter, is selected equal to 255, since 8 bits per pixel are adopted. The MSE assumes that the signal fidelity is independent of the temporal or spatial relationships between the samples of the original signal, and of any relationship between the original and the error signal. Furthermore, the MSE assumes that the signal fidelity is independent of the signs of the error signal samples [30]. Unfortunately, these assumptions often result too strict in a context of measure of visual perception of the video quality, and hence in the literature one can found several examples of situations in which the MSE fails in terms of the visual perception of the quality of the scene [27].

5.1.2 Structural SIMilarity (SSIM)

Since the human visual system is highly adapted to the natural visual environment, the modeling process of the transmitter (natural image source) and of the receiver (human visual system) can be thought as a dual problem [118]. Thus, an alternative to the MSE for the evaluation of the frame quality is the SSIM index. The basic ideas behind SSIM were introduced in [30] and more formally expressed in [35]. The SSIM is computed locally within a sliding window that moves pixel-by-pixel across the frame, providing an SSIM map that is used for averaging the SSIM values across the entire frame [35]. Considering w and w' two equally sized areas selected from the same location of two frames f_l and f_l' that are compared, the SSIM measures the similarity of three characteristics of these areas: the similarity $bv(w, w')$ of the local area luminance (brightness values), the similarity $ac(w, w')$ of the local area contrast, and the similarity $ps(w, w')$ of the local patch structure. In particular, the mean luminance $bv(w, w')$ and the similarity of the contrast $ac(w, w')$ can range from 0 to 1, being $bv(w, w') = 1$ when $\mu_w = \mu_{w'}$ and $ac(w, w') = 1$ when $\sigma_w = \sigma_{w'}$, where μ_w, $\mu_{w'}$ and σ_w, $\sigma_{w'}$ denote the means and the standard deviations, respectively. The correlation coefficient between the two frame areas $ps(w, w')$ can instead range from -1 to 1. These local similarities are then combined together providing the SSIM [35]:

$$\begin{aligned} SSIM(w,w') &= bv(w,w') \cdot ac(w,w') \cdot ps(w,w') \\ &= \frac{2\mu_w \mu_{w'} + C_{bv}}{\mu_w^2 + \mu_{w'}^2 + C_{bv}} \cdot \frac{2\sigma_w \sigma_{w'} + C_{ac}}{\sigma_w^2 + \sigma_{w'}^2 + C_{ac}} \cdot \frac{\sigma_{ww'} + C_{ps}}{\sigma_w \sigma_{w'} + C_{ps}}, \end{aligned} \tag{5.3}$$

where C_{bv}, C_{ac}, and C_{ps} are small constants introduced with the purpose to stabilize each term, that are specifically set as $C_{bv} = C_{ac} = 2C_{ps} = (C_o \cdot 255)^2$ with $C_o = 0.05$. The SSIM measures the degree of correlation between w and w' and its range lies between -1 and 1, being equal to one only when w' is linear with respect to w at all scales [119]. This index has the characteristic of being symmetric, which means that $SSIM(w,w') = SSIM(w',w)$. This implies that the same results will be provided regardless of the order of the two compared frames. Further details concerning the SSIM can be found in [27] and [35].

It is worth to notice that the SSIM is bounded, namely $|SSIM(w,w')| \leq 1$. However, several studies have revealed that in practical scenarios the SSIM often ranges between 0 and 1 [27, 35, 119, 120], since negative values of $SSIM(w,w')$ are due to negative values of the correlation coefficient, which, in turn, are due to negative $\sigma_{ww'}$ values, thus representing very particular situations of frame distortion. Therefore, in the following of this chapter, the distortion estimation for an SSIM-based approach will be performed assuming $0 \leq SSIM(w,w') \leq 1$.

5.2 Network model

Consider a single-hop 802.11e distributed network where N nodes contend for channel access using the EDCA basic access mechanism. Each source, in general, can support four ACs: VO, VI, BE, and BK. Since in this case the objective is to investigate the suitability of the video quality assessment measures, the sole VI AC will be considered active. The reason behind this choice is to avoid the contention with the other ACs, which may introduce difficulties in the interpretation of the results, since the aim is to specifically refer them to impact of the sole quality assessment measure. Let's assume a saturated traffic scenario, where a generic source has always a nonempty buffer once a packet has been successfully transmitted. Each source has to transmit the reference video sequence $\mathcal{V}_t$ to a destination that belongs to the same network. As shown in Fig. 5.1, $\mathcal{V}_t$ is encoded using the JVT H.264/SVC standard, thus the corresponding frames are classified as I, P or B [98]. The coding process provides a sequence of NALUs, that consists of a base layer and three enhancements layers. The base layer is decodable independently from the other layers, while the enhancement ones have to refer to the base layer. The set of NALUs, which are of different size, needs to be packetized. Through a packetizer it is possible to obtain a sequence of K packets $\pi_1, ..., \pi_K$ of equal size. Since the loss of a NALU can have a different impact on the final video quality, the corresponding packets need to be classified according to the possible distortion due to their loss [43, 86, 88, 89, 104]. To this purpose, the reference video sequence $\mathcal{V}_t$, obtained after encoding and then decoding the original sequence $\mathcal{V}$ (Fig. 5.1), is used at the source side to estimate the distortion due to the loss of a generic frame f_l. Precisely, for the frames whose loss implies the failure of the decoder, such as the I frames [121], the distortion is assumed infinite. Instead, for the frames whose loss does not imply the crash of the decoder, this estimation is carried out by evaluating the MSE and the SSIM between $\mathcal{V}_t$ and the video sequence $\mathcal{V}_t^{'}$. $\mathcal{V}_t^{'}$ is obtained decoding the sequence of NALUs obtained by encoding the original sequence $\mathcal{V}$, but deprived of the NALUs corresponding to the frame f_l, whose is replaced in the decoded sequence by its previous frame, namely f_{l-1}, thus applying a frame copy error concealment approach. More precisely, considering for example two frames $f_{l'}$ $(\in \mathcal{V}_t)$ and $f_{l'}^{'}$ $(\in \mathcal{V}_t^{'})$, the distortion in terms of MSE suffered by $f_{l'}^{'}$ $(\in \mathcal{V}_t^{'})$ because of the loss of $f_l^{'}$ $(\in \mathcal{V}_t^{'})$ can be evaluated as $\mathrm{MSE}[f_{l'}, f_{l'}^{'(l)}]$, and hence the distortion suffered by the entire video sequence $\mathcal{V}_t^{'}$ because of the loss of $f_l^{'}$ $(\in \mathcal{V}_t^{'})$ can be obtained by adding on all the MSE contributions, i.e. on the index l'. The apex (l) on $f_{l'}^{'(l)}$ is used to specify

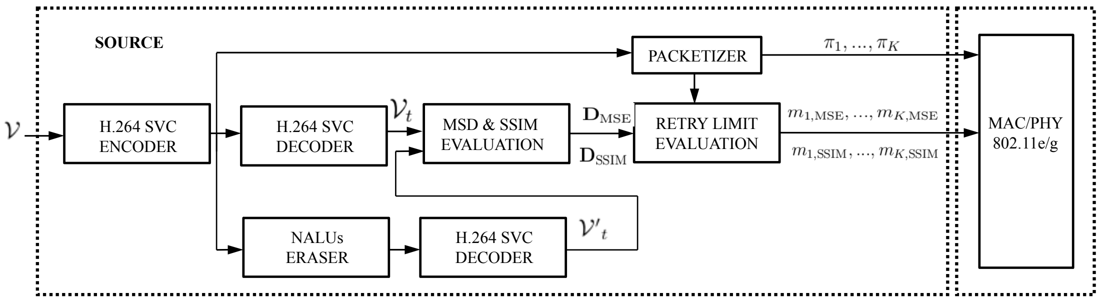

FIGURE 5.1: Single node model.

that the frame $f_{l'}^{'(l)}$ is affected by distortion due to the loss of the frame f_l'. A similar strategy can be adopted for the SSIM-based approach. In this way, one can define two measures of distortion for each frame f_l, one corresponding to an MSE-based scenario and the other to an SSIM-based scenario, thus obtaining:

$$\tilde{D}_{l,\mathrm{MSE}} = \sum_{\substack{l'=1 \\ l' \neq l}}^{L} \mathrm{MSE}[f_{l'}, f_{l'}^{'(l)}], \tag{5.4}$$

$$\tilde{D}_{l,\mathrm{SSIM}} = \sum_{\substack{l'=1 \\ l' \neq l}}^{L} \mathrm{SSIM}[f_{l'}, f_{l'}^{'(l)}], \tag{5.5}$$

with $f_{l'}$ $(\in \mathcal{V}_t)$ and $f_{l'}^{'(l)}$ $(\in \mathcal{V}_t')$. One may notice that in (5.4) and (5.5), the distortion estimations are referred to the frames. To refer them to the packets, one have to associate to each packet π_k the highest distortion value among those associated to the frames that are fully or partially contained in π_k. More precisely, for each π_k, one can identify a set of frames $\mathcal{L}_k$ for which at least a NALU is contained in π_k. Among the distortions corresponding to the frames belonging to $\mathcal{L}_k$, the maximum one is considered. Hence, the distortions associated to each packet π_k for the MSE-based and the SSIM-based scenarios can be defined, respectively, as:

$$D_{k,\mathrm{MSE}}' = \max_{l \in \mathcal{L}_k} \tilde{D}_{l,\mathrm{MSE}}, \tag{5.6}$$

$$D_{k,\mathrm{SSIM}}' = \max_{l \in \mathcal{L}_k} (L - 1 - \tilde{D}_{l,\mathrm{SSIM}}). \tag{5.7}$$

In (5.7) the complementary SSIM based distortion is evaluated in order to obtain coherent meanings for the two estimations, namely a high $D'_{k,\mathrm{MSE}}$ (high MSE) or a high $D'_{k,\mathrm{SSIM}}$ (low SSIM) identifies a high distortion. Once the distortion estimations referred to the packets are available, it is worth to discuss if, in this form, they are suitable for the setting of the retry limit in an 802.11e network. To this purpose, one can refer to the 802.11e Markov model proposed in [110], limiting the analysis to case of uniformly distributed backoff and identical nodes with activity in the sole VI AC, according to the scenario introduced at the beginning of this section. The use of a uniform random backoff is mandatory not only in the 802.11e extension, but in general in the 802.11x family of standards, and has the objective to provide fair access opportunities to all the contending nodes. In the specific case of 802.11e, each AC has its own interval in which the backoff must be generated, but, however, within this interval, the backoff distribution must be uniform. The effects of a nonuniform backoff are discussed in [122].

Since the Markovian approach for modeling an 802.11-based network is widely adopted in the literature [105, 123], this section recalls the sole fundamental equations of the model in [110] that are of interest for the study presented in this chapter. More precisely, the Markov chain analysis of the considered 802.11e network is based on the solution of the nonlinear system of $2Q$ equations reported in (4.9) in Ch.4, where p is the conditional collision probability, τ is the transmission probability, W is the minimum contention window, and m_k is the retry limit corresponding to the packet π_k. In this chapter, since the focus is placed only on the VI queue ($q = 2$), all the quantities are referred to it and thus, the subscript which specifies the VI queue, is omitted. Observe that, since m_k is unknown, (4.9) contains two equations in three unknowns: p, τ, and m_k. Using this model, the drop probability referred to π_k can then be calculated as [110]:

$$p_{\mathrm{drop}}(m_k) = p^{m_k+1}. \tag{5.9}$$

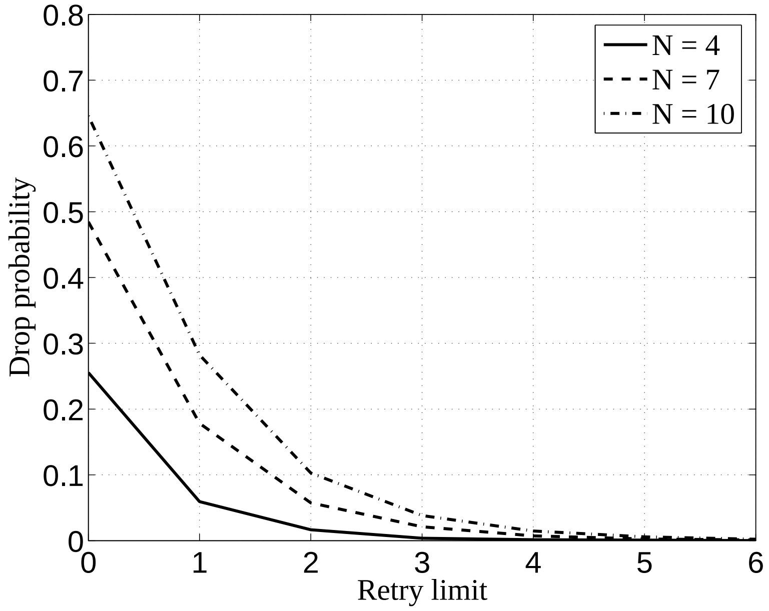

FIGURE 5.2: Drop probability as a function of the retry limit obtained for $W = 8$ and different values of N.

This latter equation shows that m_k appears also as an exponent of the function that provides $p_{\text{drop}}(m_k)$. If one intends to associate a distortion measure to the drop probability, this measure should maintain a certain adherence with (5.9). More precisely, the distortion should be not only inversely proportional to the drop probability, but should recall an exponential relationship in order to be properly exploited for setting the retry limit. As a support of this consideration, Fig. 5.2 reports a network simulation, realized by a Matlab implementation of the 802.11e EDCA, of the drop probability as a function of the retry limit for different values of the number of contending nodes. To this purpose, it worth to notice, according to (5.9), that the relationship between the drop probability and the retry limit is more complex than an exponential. However, Fig. 5.2 shows that the contribution of $m_k + 1$ as exponent may be assumed as the most relevant on that relationship.

A second aspect that should be taken into account concerns the ranges of the two considered quality assessment indexes, which are considerably different. In fact, $\tilde{D}_{l,\text{MSE}}$ can range from zero to $255^2(L-1)$, while $\tilde{D}_{l,\text{SSIM}}$ can range from zero to $L-1$. Thus, to obtain indexes with the same range that recalls the relationship in (5.9) between $p_{\text{drop}}(m_k)$

and m_k, the two distortion measures are reformulated as:

$$D_{k,\mathrm{MSE}} = 10^{\frac{\zeta \cdot D^{'}_{k,\mathrm{MSE}}}{\max\limits_{k \in \mathcal{P}} D^{'}_{k,\mathrm{MSE}}}}, \qquad k = 1, ..., K \tag{5.10}$$

$$D_{k,\mathrm{SSIM}} = 10^{\frac{\zeta \cdot D^{'}_{k,\mathrm{SSIM}}}{\max\limits_{k \in \mathcal{P}} D^{'}_{k,\mathrm{SSIM}}}}, \qquad k = 1, ..., K \tag{5.11}$$

where $\zeta(> 0)$ is a shaping parameter, whose impact on the final results will be discussed in the following section, and $\mathcal{P}$ denotes the set of all packets that can be dropped without determining the failure of the decoding process (namely those packet not associated with an infinite distortion). The drop probability for π_k must be inversely proportional to $D_{k,\mathrm{MSE}}$ (or $D_{k,\mathrm{SSIM}}$), whose reciprocal $1/D_{k,\mathrm{MSE}}$ (or $1/D_{k,\mathrm{SSIM}}$) is a quantity having as base $1/10$ and as exponent the product between ζ and the normalized value of $D^{'}_{k,\mathrm{MSE}}$ (or $D^{'}_{k,\mathrm{SSIM}}$). Thus, since the conditional collision probability is of course comprised between zero and one, $m_{k,\mathrm{MSE}}$ (or $m_{k,\mathrm{SSIM}}$) is selected to be approximately proportional to the normalized value of $D^{'}_{k,\mathrm{MSE}}$ (or $D^{'}_{k,\mathrm{SSIM}}$). This is the reason for the elaboration of the distortion adopted in (5.10) and (5.11). For compactness, these distortion measures can be joined in the vectors:

$$\mathbf{D}_{\mathrm{MSE}} = [D_{1,\mathrm{MSE}}, ..., D_{K,\mathrm{MSE}}], \tag{5.12}$$

$$\mathbf{D}_{\mathrm{SSIM}} = [D_{1,\mathrm{SSIM}}, ..., D_{K,\mathrm{SSIM}}]. \tag{5.13}$$

Now, the value of each m_k that has to be adopted to match the distortion requirement can be numerically calculated from:

$$\arg\min_{m_k \in \mathbb{N}} \left| p^{m_k+1} - \frac{1}{D_{k,*}} \right|, \tag{5.14}$$

where $D_{k,*} = D_{k,\mathrm{MSE}}$, when the MSE-based approach is adopted, and $D_{k,*} = D_{k,\mathrm{SSIM}}$, when the SSIM-based approach is adopted. In summary, as shown in Fig. 5.1, each source node that has to transmit a sequence of packets $\pi_1, ..., \pi_K$ obtains a set of retry limits ($m_{1,\text{MSE}}, ..., m_{K,\text{MSE}}$ or $m_{1,\text{SSIM}}, ..., m_{K,\text{SSIM}}$), which is used during the medium access procedure. Observe that, from a mathematical point of view, (5.14) holds also for the packets associated to an infinite distortion, since in this case (5.14) leads to an infinite retry limit. Of course, for these packets the calculation can be omitted by directly setting an infinite retry limit. Besides, using (5.14), the packets producing a lower distortion are associated to a higher drop probability, and hence to a lower retry limit. It is also worth to notice that a real time estimation of the distortion at the source node is not necessary, since this estimation can be performed offline as a quantity associated to the video sequence. An application of this approach is discussed in [124].

At the destination, a frame-copy approach is applied to the corrupted sequence, in order to derive the performance in terms of PSNR, MSE and SSIM, by comparing the sequence $\mathcal{V}_t$ and the corrupted one $\mathcal{V}_t^{\prime}$. Thus, using these three quantities, the result achieved using an adaptation based on $\mathbf{D}_{\text{MSE}}$ (or $\mathbf{D}_{\text{SSIM}}$) can be observed from different perspectives.

5.3 Results and discussion

To derive the results, the *Bus* sequence consisting of 65 frames encoded at a frame interval equal to 1/15 is adopted as the reference sequence $\mathcal{V}_t$. The rest of this section presents the performance obtained adopting the MSE-based and the SSIM-based approaches by comparing the results in terms of PSNR, MSE and SSIM. Even if, as explicitly stated in (5.2), the PSNR and the MSE are directly related, both are maintained in the presentation of the results, in order to provide a complete view of the achievable performance. All results are derived using a Matlab implementation of the 802.11e EDCA, considering $W = 8$ and carrying out 20 simulations for each investigated network scenario. The curves in the presented figures represent the values averaged over the simulations and the nodes.

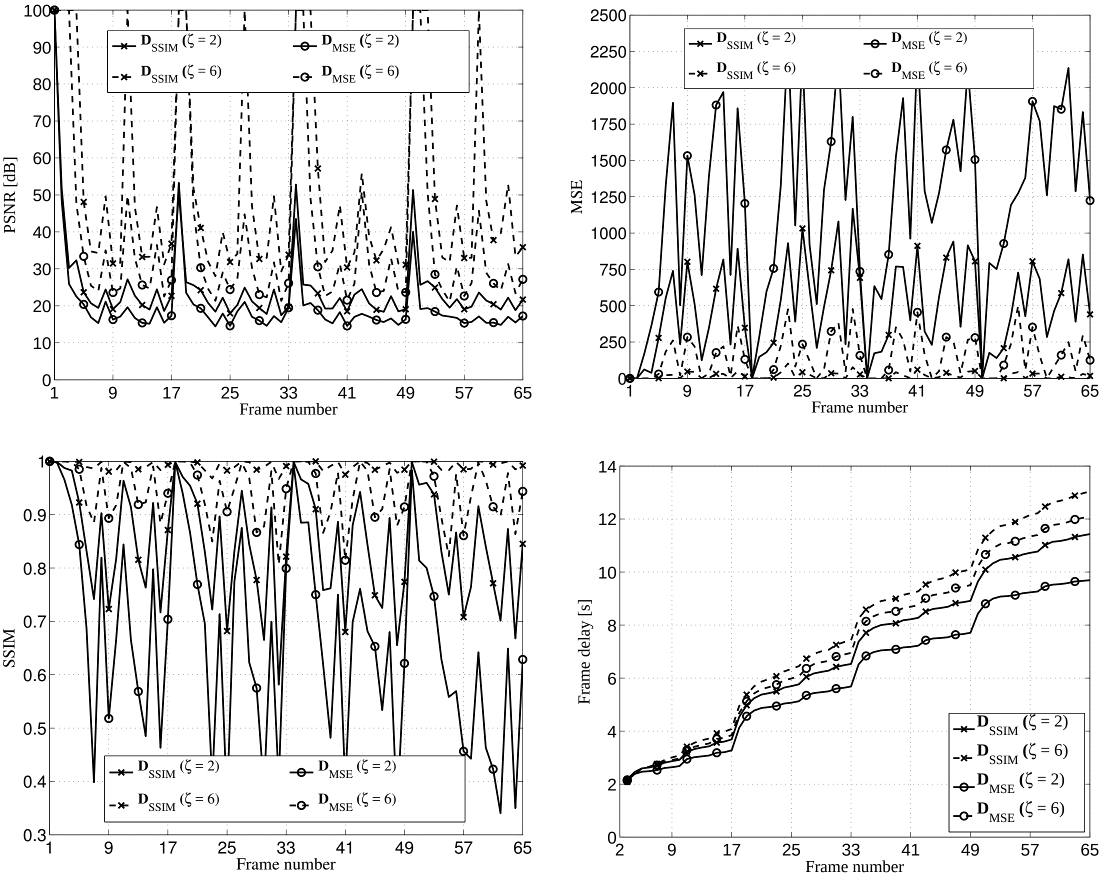

FIGURE 5.3: PSNR (upper left), MSE (upper right), SSIM (lower left) and cumulative delay (lower right) for $N = 7$ and using different ζ values for the MSE and SSIM-based scenarios.

Fig. 5.3 reports the PSNR, MSE, SSIM, and the cumulative frame delay in the presence of $N = 7$ contending nodes when the MSE-based and the SSIM-based approaches are adopted for setting the retry limit, considering different values of the parameter ζ. The typical oscillating behavior of the measures is due to the different impact of the loss of a frame on the other ones and depends on the characteristics of the specific video sequence [42]. These figures reveal that in all the tested cases, the adoption of the SSIM-based setting allows one to reach a better average quality. One can notice, in fact, that the PSNR, MSE, and SSIM agree in identifying the SSIM-based technique as that providing the most satisfactory performance. Moreover, the better results obtained using $\mathbf{D}_{\text{SSIM}}$ appear not only performing a comparison between the two approaches for the same ζ value, but also considering the best performance among all the adopted ζ values.

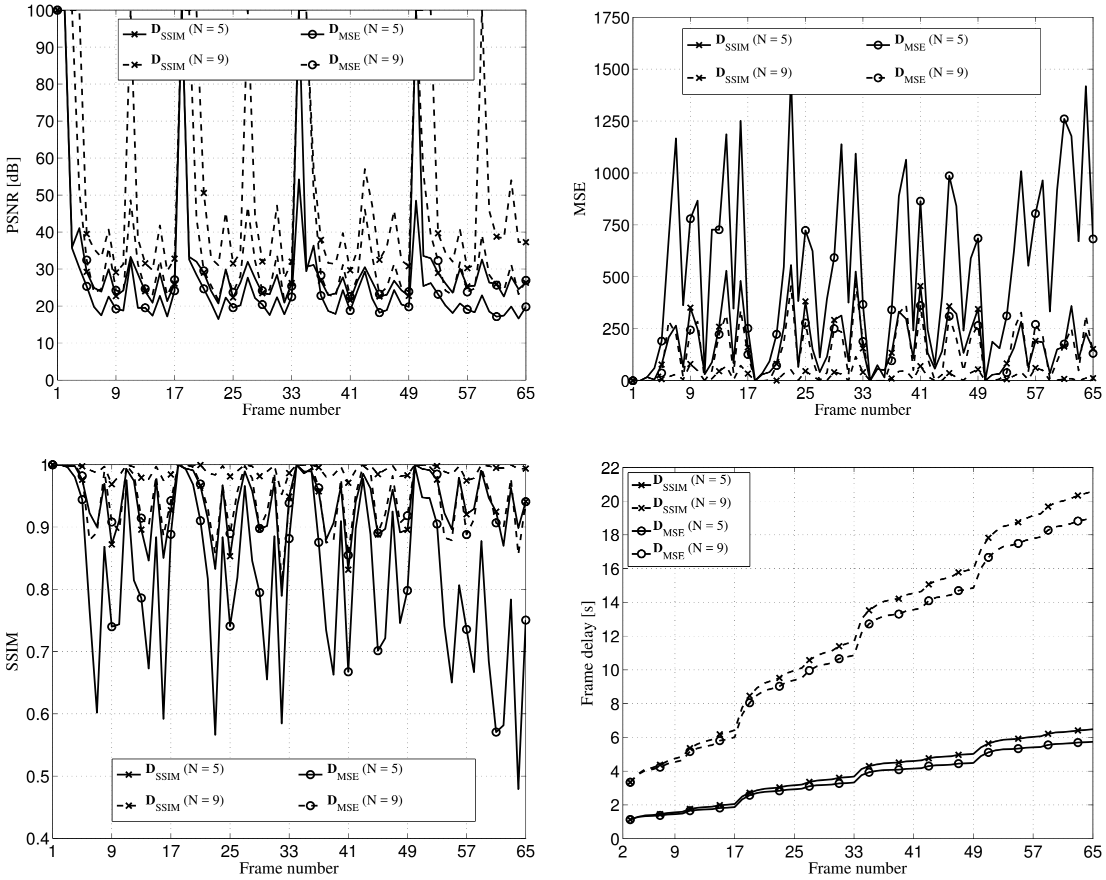

FIGURE 5.4: PSNR (upper left), MSE (upper right), SSIM (lower left) and cumulative delay (lower right) for the MSE and SSIM-based scenarios, using $\zeta = 5$ and different values of N.

This behavior is confirmed by Fig. 5.4, in which the comparison is repeated in the presence of a different number of contending nodes N and a fixed value of ζ. In particular, this figure reveals that, when N is high and hence the number of collisions is high too, the two approaches lead to a similar performance showing better results anyway with the adoption of the SSIM. As N gets lower, the SSIM-based approach becomes even more robust with respect to the MSE-based one. A combined observation of Figs. 5.3 and 5.4, with particular reference to the cumulative frame delay, enables to more deeply discuss the impact of N and ζ on the results. More precisely, as expected, the increase of the number of contending nodes leads to an increase of the delay, due to a larger number of collisions, regardless of the adopted approach (MSE or SSIM-based). When N is fixed, one can notice that an increase of ζ leads to an increase of the delay. As

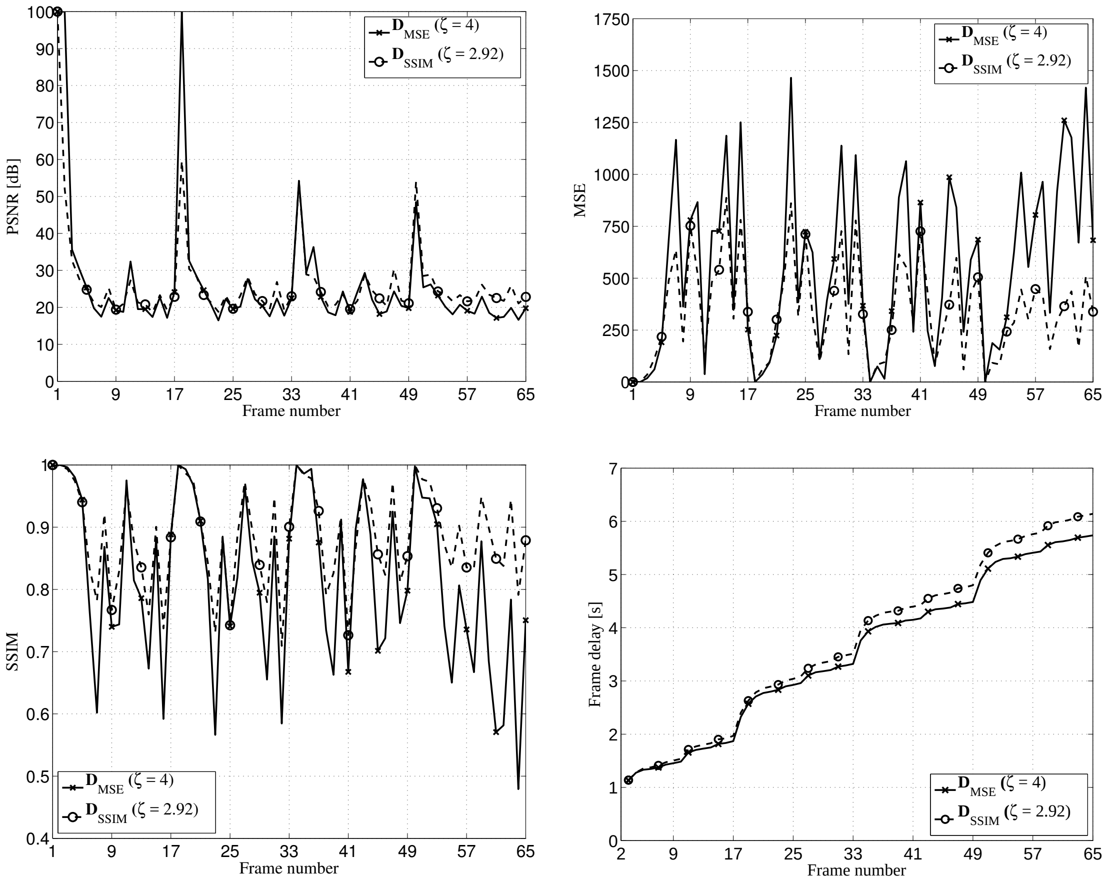

FIGURE 5.5: PSNR (upper left), MSE (upper right), SSIM (lower left) and cumulative delay (lower right) for $N = 5$ using $\zeta = 4$ for the MSE-based scenario and $\zeta = 2.92$ for the SSIM-based one.

one can infer from (5.10) (or (5.11)), this phenomenon is due to an increase of $D_{l,\text{MSE}}$ (or $D_{l,\text{SSIM}}$), which implies the use of higher m_k values. Summarizing, by increasing ζ, one increases the retry limits, reducing the drop probability and increasing the delay. The above considerations and the direct comparison of the delays in Fig. 5.3 for a given ζ may raise the doubt that the better performance of the SSIM-based adaptation with respect to the MSE-based one may be due just to a higher number of retransmissions and not to a more reliable modeling of the distortion. To investigate this issue, a further comparison between the two approaches is carried out by imposing that the total number of allowed retransmissions for the entire video stream must be identical, that is:

$$\sum_{k \in \mathcal{P}} m_{k,\text{MSE}} = \sum_{k \in \mathcal{P}} m_{k,\text{SSIM}}. \tag{5.15}$$

The satisfaction of this requirement implies that different ζ values can be selected in the two cases. Precisely, to match (5.15) in the presence of $N = 5$ contending nodes when the Bus sequence is adopted, one obtains $\zeta = 2.92$ for the SSIM-based scenario corresponding to the case $\zeta = 4$ for the MSE-based one. Fig. 5.5 shows the results of this comparison, confirming that, even when the constraint in (5.15) is applied, the SSIM-based approach results preferable. Besides, one may notice that, as expected, the cumulative frame delays are very similar, being identical the total number of allowed retransmissions. The limits of the use of the vector $\mathbf{D}_{\text{MSE}}$ lie on the fact that the MSE may provide the same results for frames that are affected by the same difference of pixel intensity, compared to the original one, but characterized by drastic differences in the perceived video quality [35]. In the same way, the MSE may provide very different results for frames with very similar perceived video quality. This can lead, for some π_k, to a very high $D_{k,\text{MSE}}$ value compared to that associated to other packets, even when the effects of their possible loss, in terms of perceived video quality, is very similar. The SSIM, instead, being more adherent to the perceived quality is able to more reliably model the distortion in agreement to the sensing of a human end user. As a further confirmation, Table 5.1 reports the MSE, PSNR, and SSIM (averaged also over the frames), and the maximum cumulative frame delay for all the simulated cases: $N = 4, 7, 10$ and $\zeta = 3, 5, 7$. Finally, according to the practical assumption that the SSIM lies between 0 and 1, none of the performed simulations has revealed negative values for the SSIM even for a single frame comparison.

Summarizing, the higher reliability of the SSIM with respect to the MSE as a quality assessment measure for images and videos seems to have a direct outcome on the retry limit adaptation policy. This may be, on one hand, a somewhat expected result, considering that the SSIM is a more sophisticated indicator as compared to the MSE. On the other hand, however, it is interesting to notice that the other quality assessment measure, namely the MSE, confirms, when used to evaluate the performance, that the

		Settings adopted by nodes in transmission					
		$\zeta = 2$		$\zeta = 4$		$\zeta = 6$	
		$\mathbf{D}_{\mathrm{MSE}}$	$\mathbf{D}_{\mathrm{SSIM}}$	$\mathbf{D}_{\mathrm{MSE}}$	$\mathbf{D}_{\mathrm{SSIM}}$	$\mathbf{D}_{\mathrm{MSE}}$	$\mathbf{D}_{\mathrm{SSIM}}$
N=3	Average MSE	1979.91	1267.10	583.92	288.60	316.19	64.14
	Average PSNR [dB]	18.18	20.09	27.69	29.15	34.85	45.40
	Average SSIM	0.53	0.67	0.80	0.91	0.88	0.97
	Maximum cumulative frame delay [s]	1.91	2.31	2.35	2.67	2.52	2.80
N=5	Average MSE	1615.53	858.94	545.57	174.98	211.84	42.51
	Average PSNR [dB]	18.98	21.87	26.59	32.52	36.62	48.23
	Average SSIM	0.58	0.76	0.82	0.94	0.92	0.98
	Maximum cumulative frame delay [s]	4.83	5.67	5.74	6.48	6.22	6.78
N=7	Average MSE	1209.32	479.97	336.11	72.90	145.44	18.25
	Average PSNR [dB]	20.52	24.91	32.65	42.28	39.41	52.60
	Average SSIM	0.67	0.85	0.88	0.97	0.94	0.99
	Maximum cumulative frame delay [s]	9.69	11.43	11.30	12.60	12.10	13.04
N=9	Average MSE	542.14	165.98	140.02	21.52	52.29	2.31
	Average PSNR [dB]	25.58	32.07	38.51	51.67	44.98	66.18
	Average SSIM	0.81	0.94	0.94	0.99	0.98	1.00
	Maximum cumulative frame delay [s]	16.73	19.05	19.01	20.58	20.02	20.90

TABLE 5.1: Average MSE, PSNR, SSIM, and maximum cumulative delay for $N = 3, 5, 7, 9$ and $\zeta = 2, 4, 6$ for the MSE-based and SSIM-based scenarios.

SSIM-based approach is more suitable. Furthermore, even in the presence of requirements on the total number of retransmissions, the use of the SSIM can be preferable. However, when the purpose is the implementation of the distortion estimation at the source node, also the complexity of the adopted quality assessment index must be taken into account. To investigate this further aspect, the CPU time required by the two compared approaches has been monitored during the simulations. In particular, the average CPU time required for the calculation of $\mathbf{D}_{\mathrm{MSE}}$ has been equal to 0.27 s, while that necessary to derive $\mathbf{D}_{\mathrm{SSIM}}$ has been equal to 10.49 s. Thus, the main drawback of using the SSIM-based approach is a computational cost that can be much higher than that required by the MSE-based one.

5.4 Conclusions

A study of the influence of the MSE and the SSIM indexes on the performance of an 802.11e network using an adaptive retransmission policy has been presented through this chapter. The numerical results have shown that the adoption of an SSIM-based approach is able to provide a better video quality at end user, even in the presence of constraints on the total number of allowed retransmissions, but accepting a higher computational burden. Thus one may adopt, in an adaptive retransmission policy, a SSIM based distortion instead of a MSE based one, aiming to better shape the effects on the total distortion on a video sequence due to the loss of a frame, but at the cost of a high computational cost required to evaluate the SSIM values. To overcome this issue, an algorithm for a fast evaluation of the SSIM based distortion is proposed in the following chapter.

Chapter 6

Fast distortion estimation for H.264/SVC encoded videos

The importance of on-demand and live-streaming leads to the need of novel standards and techniques to improve the quality of the video at the end-user. Several studies have investigated this issue, by proposing cross-layer mapping architectures for video transmissions [125], streaming methods with bandwidth estimation [126], and traffic prioritization [43]. As already mentioned in the previous chapters, an interesting solution for providing traffic prioritization can be handled at MAC layer, in order to provide higher protection to the more important information. In particular, the idea of adapting the number of retransmissions of a packet according to the distortion produced on the entire video by its loss has been explored in [42, 90, 91]. The adoption of video quality assessments can result very effective for classifying the information carried by a packet of an encoded video. The loss of different parts of a video can lead to different effects on the whole sequence, thus in [127,128], a layered protection technique and a packet prioritization for video streaming have been presented. Furthermore, in [129,104] a

The content of this chapter is based on F. Babich, M. Comisso, and R. Corrado, "Fast Distortion Estimation Based on Structural Similarity for H264/SVC Encoded Videos", *IEEE Vehicular Technology Conference (VTC)*, Glasgow, Scotland (UK), pp. 1 - 5, 11 - 14 May 2015.

Mean Squared Error (MSE)-based distortion estimation model has been proposed. Unfortunately, the MSE often fails in terms of the visual perception of the quality of the scenes [29,Ch.15,27]. Therefore, in [130–133] the authors have presented solutions for motion estimation, rate control and buffer management policies, based on the more reliable Structural SIMilarity (SSIM) index. In [142] the authors have compared the MSE and the SSIM in terms of their capability to estimate the video distortion for adaptive retransmission purposes, showing that an SSIM-based estimation can outperform an MSE-based one, but at the cost of a considerable computational burden.

This chapter aims to overcome this drawback, proposing an algorithm that reduces the computational cost and the time required to evaluate the distortion in terms of SSIM. The proposed solution allows one to obtain an estimation of the SSIM-based distortion considering only a partial decoding process and just three SSIM-based evaluations. The results show a relevant reduction of the calculation time, at the cost of a low approximation error with respect to the exact SSIM-based distortion, even when video sequences without homogeneous motion are involved. The SSIM estimation algorithm is finally numerically tested over an 802.11e distributed network implementing adaptive retransmission strategies.

The chapter is organized as follow. Section 6.1 introduces the SSIM index and the hierarchical coding structure of the H.264 Scalable Video Coding (SVC) standard. Section 6.2 presents the proposed SSIM-based distortion estimation algorithm. Section 6.3 discusses the numerical results. Finally, Section 6.4, summarizes the most relevant conclusions.

6.1 System model

6.1.1 Structural SIMilarity - SSIM

The concept underlying the original SSIM approach is that a human visual system is highly adapted to extract structural information from visual scenes [27]. As a quality assessment, the SSIM has proved highly effective in measuring the fidelity of the signals [119]. Further details concerning the characteristics of the SSIM and its mathematical derivation can be found in [27] and [35] and for more information we redirect to section 5.1.2.

6.1.2 H.264/SVC hierarchical coding structure

The target of the SVC standardization is to make possible the encoding of a high-quality video containing one or more subsets of a given bit stream that can themselves be decoded with a complexity and a reconstruction quality similar to those achieved using the H.264 AVC design [72]. The video sequence is subdivided into GOPs of size α. Each frame of the sequence, can be classified as I, P and B. In this chapter, the focus is placed

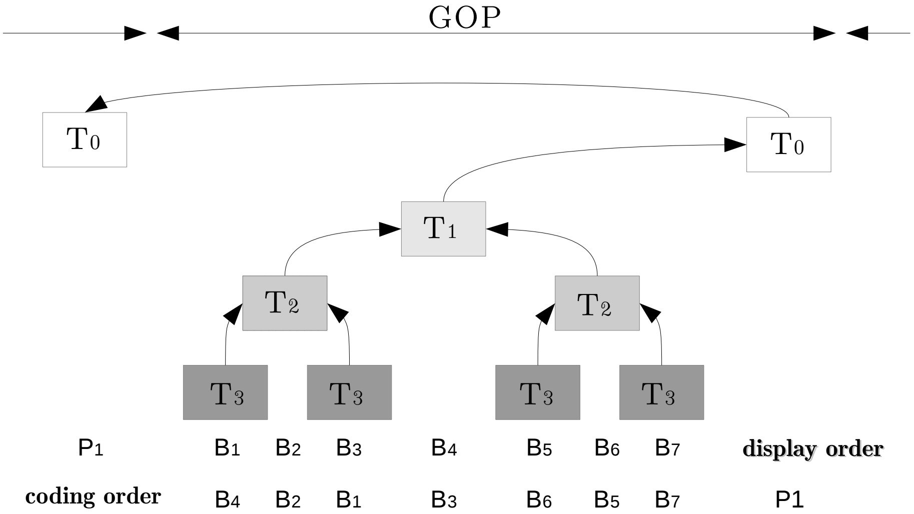

FIGURE 6.1: H.264/SVC hierarchical coding structure.

on the hierarchical structure that imposes a coding dependency between the B frames in order to allow the temporal scalability. Each temporal level can be identified by T_γ with $0 \leq \gamma \leq \Gamma$, where $\gamma = 0$ for a P frame and $\gamma = \Gamma$ for the highest temporal level of a B frame. For each temporal level T_γ, it is possible to obtain a valid bit stream removing all the Network Abstraction Layer Units (NALUs) of a temporal level $\gamma^{'} > \gamma$. Since the coding of a B frame needs a reference from a subsequent frame, this latter frame needs to be encoded before. This means that the coding order of the frames differs from the display order. As an example of this behavior, Fig. 6.1 reports a hierarchical structure for $\Gamma = 3$.

6.2 SSIM-based distortion estimation

6.2.1 Exact estimation

Let's consider a video sequence $\mathcal{V} = \{f_l : l = 1, ...L\}$, consisting of L frames, which is encoded by the hierarchical prediction structure of the H.264 standard. As previously discussed, each frame f_l has a different importance and, consequently, a different impact in terms of the total distortion on the decoded video sequence, in the case that it is lost. In general, the loss of a frame f_l determines a distortion $\tilde{D}_{l,l'}$ on the generic frame $f_{l'}$. Considering all the frames $f_{l'} \in \mathcal{V}$, the total distortion introduced on $\mathcal{V}$ by the loss of a frame f_l can be expressed as:

$$\tilde{D}_{l,SSIM} = \sum_{\substack{l'=1 \\ l' \neq l}}^{L} \tilde{D}_{l,l'}. \tag{6.1}$$

Performing this operation for all the L frames of the sequence, it is possible to obtain the set $\tilde{\mathcal{D}} = \{\tilde{D}_{l,SSIM} : l = 1, ..., L\}$ of all the total distortions. To evaluate each $\tilde{D}_{l,SSIM}$, three steps are required. First, it is necessary to remove from the encoded sequence all the NALUs belonging to the frame f_l. Second, a decoding process must be performed to obtain the decoded video sequence. Third, the distortion on each frame $f_{l'}$ must be evaluated to obtain, for each frame, the distortion $\tilde{D}_{l,l'}$, which has to be used in (6.1). While the time required for evaluating (6.1) and for extracting the NALUs belonging to f_l can be considered negligible, that required to perform the decoding process and the SSIM calculation cannot be neglected. Let's consider $\mathcal{V}$ and the video sequence $\mathcal{V}'=\{f_l':l=1,...,L,l\neq l'\}$, obtained after the decoding of the H.264 flow deprived by the NALUs belonging to a frame $f_{l'}$, with $1<l'\leq L$. Assuming to receive the I frame to avoid the failure of the entire decoding process, the time required to evaluate $\tilde{\mathcal{D}}$, can be evaluated as:

$$t_T=(L-1)\cdot t_{\text{DEC}}+(L-1)^2\cdot t_{\text{SSIM}}, \qquad (6.2)$$

where t_{DEC} is the time required for the decoding of the sequence, and t_{SSIM} is the time required to evaluate the SSIM between f_l and $f_l^{'}$ belonging to $\mathcal{V}$ and $\mathcal{V}'$, respectively.
A first reduction of the calculation time may be obtained exploiting the hierarchical structure of the encoder and limiting the impact of the distortion due to the loss of a frame $f_{l'}^{'}$ to the sole GOP containing $f_{l'}^{'}$. Therefore, for the exact estimation, a new reduced calculation time t_S may be derived as:

$$t_S=(L-1)\cdot[t_{\text{DEC}}+(\alpha-1)\cdot t_{\text{SSIM}}]=t_{d_S}+t_{s_S}, \qquad (6.3)$$

where $t_{d_S}=(L-1)\cdot t_{\text{DEC}}$ is the time required to perform all the decoding operations, and $t_{s_S}=(L-1)\cdot(\alpha-1)\cdot t_{\text{SSIM}}$ is the one required to evaluate all the SSIM values. Despite the noticeable reduction of the calculation time obtained in (6.3) with respect to (6.2), the computational cost remains prohibitive for video streaming applications. To deal with this issue, a low complexity approach is presented in the following section.

6.2.2 Proposed algorithm

As a preliminary step, the set $\mathcal{V} = \mathcal{I} \cup \mathcal{P} \cup \mathcal{B}$ is partitioned into three disjoint subsets, where $\mathcal{F}_I$ is the subset of the I frames, $\mathcal{F}_P$ is the subset of the P frames whose P elements are renumbered as $\mathcal{F}_P = \{f_p^{\mathrm{P}} : p = 1, ..., P\}$, and $\mathcal{F}_B$ is the subset of the B frames whose B elements are renumbered as $\mathcal{F}_B = \{f_b^{\mathrm{B}} : b = 1, ..., B\}$. Let's now define T_{γ_l} as the temporal level γ_l of the frame f_l. Analyzing the hierarchical structure, it is possible to define the number of frames that will be affected by distortion in the case that the frame f_l is lost. When the loss involves a frame f_l of temporal level $T_{\gamma_l} = 0$ (namely a P frame), the number of affected frames can be evaluated as:

$$M^{\mathrm{P}} = \alpha - 1, \tag{6.4}$$

for a GOP of size α. Instead, when the loss involves a frame of temporal level $T_{\gamma_l} > 0$ (namely a B frame), the number of affected frames, which will be always lower than M^{P}, can be derived as:

$$M^{\mathrm{B}}(T_{\gamma_l}) = 2^{T_\Gamma - T_{\gamma_l} + 1} - 2. \tag{6.5}$$

For mathematical purposes, and recalling that in practical scenarios the SSIM ranges between 0 and 1 [27, 35, 119, 120], we introduce the quantity $D_{l,l'}$ as the complementary of the SSIM-based distortion:

$$D_{l,l'} = 1 - \tilde{D}_{l,l'}. \tag{6.6}$$

Let's assume motion homogeneity in the video. This assumption allows one to consider the pair of distortions $D_{l,l-1}$ and $D_{l,l+1}$ equal for all the frames with the same temporal level T_{γ_l}. One may also reasonably assume that the loss of a B frame in the first GOP has negligible effects on the following GOP without the P frame. Thus, one can first remove $f_{b'}^{\mathrm{B}}$ and f_2^{P}, where $f_{b'}^{\mathrm{B}}$ is the B frame with temporal level equal to 1 in the first GOP and f_2^{P} is the P frame in the second GOP. Subsequently, one can measure the distortion values $D_{b',b'-1}^{\mathrm{B}}$, $D_{b',b'+1}^{\mathrm{B}}$ and $D_{2,r}^{\mathrm{P}}$, where f_r is the temporal neighbor frame of f_2^{P}

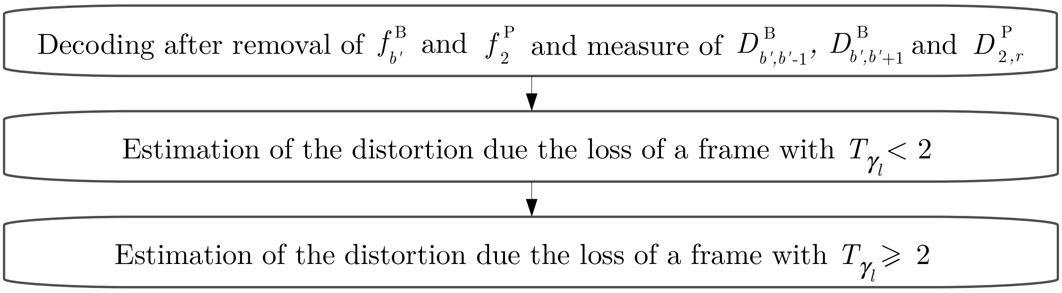
Decoding after removal of $f^{\mathrm{B}}_{b'}$ and f^{P}_{2} and measure of $D^{\mathrm{B}}_{b',b'-1}$, $D^{\mathrm{B}}_{b',b'+1}$ and $D^{\mathrm{P}}_{2,r}$
Estimation of the distortion due the loss of a frame with $T_{\gamma_l} < 2$
Estimation of the distortion due the loss of a frame with $T_{\gamma_l} \geqslant 2$

FIGURE 6.2: Proposed algorithm.

within the same GOP. The apexes B and P are used to indicate if the quantity is related to a B or a P frame. For estimating the complementary distortions due the loss of the two removed frames, we can rely on a practical observation, formulated by performing exhaustive tests on several H.264 video sequences. According to this observation, the complementary distortion $D_{l,l'}$ on the neighbor frames preceding the lost frame f_l, and, similarly, that on the neighbor frames following f_l, may be suitably approximated by a linear decreasing function. To this purpose, define $j = 1 + M^{\mathrm{B}}(T_{\gamma_l})/2$. If f_l is a B frame, $D_{l,l'}$ may be approximated by a first-order polynomial passing through the points $Z_1(l+1, D_{l,l+1})$ and $Z_2(l+j, 0)$, for $l < l' \leq l+j$, and through the points $Z_1'(l-1, D_{l,l-1})$ and $Z_2'(l-j, 0)$, for $l-j \leq l' < l$. If f_l is a P frame, $D_{l,l'}$ may be approximated by a first-order polynomial passing through the points $J_1(l-1, D_{l,l-1})$ and $J_2(l-\alpha, 0)$, for $l-\alpha < l' < l$, namely for each frame in the same GOP of f_l, which is temporally placed at the end of the GOP. Thus, recalling $D^{\mathrm{B}}_{b',b'+1}$, $D^{\mathrm{B}}_{b',b'-1}$ and $D^{\mathrm{P}}_{2,r}$, and using them as the ordinates of Z_1, Z_1' and J_1, respectively, one can estimate all the $D_{l,l'}$ values due the loss of a frame with $T_\gamma < 2$, that is all the P frames and all the B frames of temporal level $T_\gamma = 1$.

For the loss of a B frame with $T_\gamma \geq 2$, one may estimate all the corresponding complementary distortions by exploiting the previous linear approximation based on the points Z_1 and Z_1', once their ordinate values are properly estimated. To this purpose, one has now to apply the linear approximation considering the temporal level and the complementary distortion. In particular, one can notice that when f_l is a lost B frame and $f_{l'}$ is its temporal neighbor frame, the complementary distortion $D_{l,l'}$ may be approximated by a first-order polynomial passing through the points $W_1(T_\Gamma, 0)$ and $W_2(T_1, D_{b',b'-1})$, for $l' = l - 1$, and through the points $W_1(T_\Gamma, 0)$ and $W_2(T_1, D_{b',b'+1})$, for $l' = l + 1$. Through this approximation, it is possible to estimate the complementary distortion effects on all the frames of $\mathcal{V}$. Define as $q_{l,l'}$ the distance (in terms of number of frames) between a frame $f_{l'}$ and a lost frame f_l. The complementary distortion on f_l due to the loss of a P frame f_p^{P} can be approximated by:

$$D_{p,l}^{\mathrm{P}} \cong \frac{Q_{p,l}^{\mathrm{P}} \cdot D_{2,r}^{\mathrm{P}} \cdot (\alpha - q_{p,l})}{(\alpha - 1)}, \tag{6.7}$$

where $Q_{p,l}^{\mathrm{P}} = 1$ if f_l is a B frame in the same GOP of f_p, and $Q_{p,l}^{\mathrm{P}} = 0$ otherwise. Similarly, it is possible to approximate $D_{b,l}^{\mathrm{B}}$ for a B frame f_l when a B frame f_b is lost as:

$$D_{b,l}^{\mathrm{B}} \cong \frac{Q_{b,l}^{\mathrm{B}} \cdot D_*^{\mathrm{B}} \cdot \left[M^{\mathrm{B}}(T_{\gamma_b}) - 2q_{b,l} + 2\right] \cdot (T_\Gamma - T_{\gamma_b})}{M^{\mathrm{B}}(T_{\gamma_b}) \cdot (T_\Gamma - 1)}, \tag{6.8}$$

where $Q_{b,l}^{\mathrm{B}} = 1$ for $1 \le q_{b,l} \le M^{\mathrm{B}}(T_{\gamma_b})/2$, and $Q_{b,l}^{\mathrm{B}} = 0$ otherwise, while D_*^{B} represents $D_{b',b'-1}^{\mathrm{B}}$ or $D_{b',b'+1}^{\mathrm{B}}$, depending on the frame whereof we are evaluating the complementary distortion. In particular, if f_l is a frame that temporally precedes f_b (namely the one lost), then $D_*^{\mathrm{B}} = D_{b',b'-1}^{\mathrm{B}}$, otherwise $D_*^{\mathrm{B}} = D_{b',b'+1}^{\mathrm{B}}$. Performing the approximations (6.7) and (6.8) for all the P and all the B frames, respectively, it is possible to obtain all the $D_{l,l'}$ values that can be used in (6.6) to derive the $\tilde{D}_{l,l'}$ values, and, subsequently, in (6.1) to derive the set of the total approximated distortions $\tilde{\mathcal{D}}_A$. A summarizing diagram of the proposed algorithm is reported in Fig. 6.2. Fig. 6.3, instead, shows a comparison between the exact distortion $\tilde{D}_{l,SSIM}$ in (6.1) and the approximated one, with the purpose to confirm the suitability of the first-order approximations adopted to estimate the complementary distortions. The curves, which are obtained considering the Foreman sequence, confirm that, as expected, the loss of a P frame introduces a higher distortion as compared to the loss of a generic B one, since a P frame belongs to the lowest

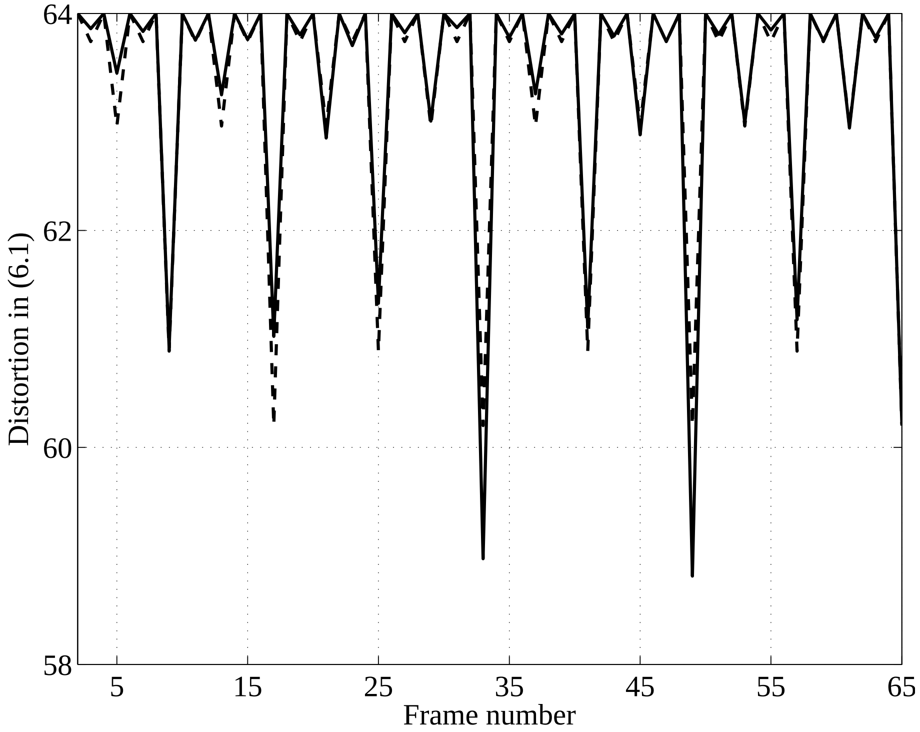

FIGURE 6.3: Exact (solid line) and approximated (dashed line) distortions for the Foreman test sequence.

temporal level T_0, but simultaneously reveal that the proposed algorithm reliably model this higher distortion. In terms of calculation time, the proposed algorithm dramatically reduces the time required to perform all the decoding operations, namely t_{d_A}, and that required to evaluate all the SSIM values, namely t_{s_A}, even when compared with the reduced approach that leads to (6.3). The time required by the algorithm, in fact, can be expressed as:

$$t_A = t'_{\mathrm{DEC}} + 3 \cdot t_{\mathrm{SSIM}} + t_{\mathrm{ALG}}, \tag{6.9}$$

where $t'_{\mathrm{DEC}} = t_{d_A}$ represents the calculation time required to perform the decoding process of the first two GOPs, t_{SSIM} is approximately equal to $t_{s_A}/3$, and t_{ALG}, which can be substantially neglected as compared to the other two times, is the time required to estimate the complementary distortion on the frames of the sequence according to (6.7) and (6.8).

6.3 Numerical results

This section presents the numerical results obtained estimating the SSIM-based distortion for different Common Intermediate Format (CIF) and 4CIF (4×CIF) videos (4CIF sequences are differentiated from the CIF sequences by a " * " on the sequence name). The proposed algorithm is compared to the exact estimation in terms of calculation time and considering the relative estimation error. The adopted test sequences, which are reported in Table 6.1, are encoded with the hierarchical coding prediction structure of the H.264 standard, considering different coding characteristics, in order to perform an exhaustive testing of the proposed algorithm. From Table 6.1, it is possible to notice that, with the adoption of the proposed algorithm, the approximated estimations show, with respect to the exact estimations, an average relative error that lies around 3% and that remains lower than 10% in all cases. The average error tends to slightly increase for videos where the motion is more accentuated, due to the initial assumption of motion homogeneity. However, even in these scenarios, such as the Football test sequence, in which there is a non-homogeneous motion, the algorithm introduces an error that lies around 10%. Moreover, Table 6.1 reveals how the adoption of this algorithm allows a

		EXACT ESTIMATION			PROPOSED ALGORITHM			
Test sequence	α	$t_{d_S}[s]$	$t_{s_S}[s]$	$t_S[s]$	$t_{d_A}[s]$	$t_{s_A}[s]$	$t_A[s]$	Relative Error $[\%]$
Bus	16	210.82	39.78	250.60	1.60	0.26	1.86	5.58±6.45
Container	16	196.89	43.28	240.17	1.52	0.28	1.80	1.82±1.38
Foreman	16	199.99	43.52	243.51	1.54	0.29	1.83	3.09±3.69
News	16	196.79	43.44	240.23	1.51	0.28	1.79	1.48±1.08
Highway	16	198.47	43.25	241.72	1.52	0.27	1.79	1.83±1.27
Akiyo	16	196.70	43.67	240.37	1.49	0.27	1.76	0.76±1.02
Football	16	212.24	41.70	253.94	1.62	0.28	1.90	9.82±10.59
Crew (*)	16	831.96	184.39	1016.35	6.14	0.70	6.84	3.70±4.49
Harbour (*)	16	822.20	181.83	1004.03	6.32	0.70	7.02	4.55±5.04
City (*)	16	817.09	180.64	997.73	6.16	0.72	6.88	4.52±3.75
Ice	8	199.53	40.27	239.80	0.75	0.26	1.01	2.91±2.76
Silent	8	200.60	40.13	240.73	0.74	0.29	1.03	2.20±1.93
Coastguard	16	148.58	39.27	187.85	1.12	0.28	1.40	3.79±3.14

TABLE 6.1: Computational time and relative error for the SSIM-based distortion using the exact estimation and the proposed algorithm.

drastic reduction of the calculation time t_A, as compared to the reduced approach t_S that allows the exact estimation. It is possible to notice, in fact, that, while t_S lies around 240 seconds for the CIF sequences, t_A always remains below 2 seconds. A similar behavior holds also for the 4CIF sequences.

6.3.1 Application

As a possible application of the proposed algorithm, consider an 802.11e distributed network with adaptive retransmissions. Since frame transmission and reception require spending energy by 802.11 wireless nodes, it may be desirable for each node to carefully control the overall number of transmissions necessary to send a video [134]. Hence, fixing a constraint on the retry limit during the channel contention may be a handy solution for energy saving applications. Since, in the H.264 coding, each frame has a different impact on the sequence when it is lost [72],[135],[43], the retry limit for each frame f_k may be selected according to the distortion $D_{l,l'}$. Let's consider, without loss of generality, each frame contained in a single packet. Adopting a Markovian approach for modeling the channel contention mechanism in an 802.11e random access network [110], it is possible to derive the drop probability $p_{\text{drop}}(m_l)$ for a frame f_l as:

$$p_{\text{drop}}(m_l) = p^{m_l+1}, \tag{6.10}$$

where m_l is the retry limit associated to f_l and p is the conditional collision probability [110]. Like in Chapter 5, the focus here is placed only on the VI packets, thus the index 2 referring to the VI queue $(q = 2)$ is omitted to simplify the notation. Introducing a parameter $En(\mathcal{V})$ to control the total number of retransmissions associated to a video sequence $\mathcal{V}$, one may limit the maximum energy consumption of a node for the transmission of $\mathcal{V}$. Through the manipulation of this parameter, it is possible to use the distortion value associated to each frame f_l, maintaining a constraint on the sum of all

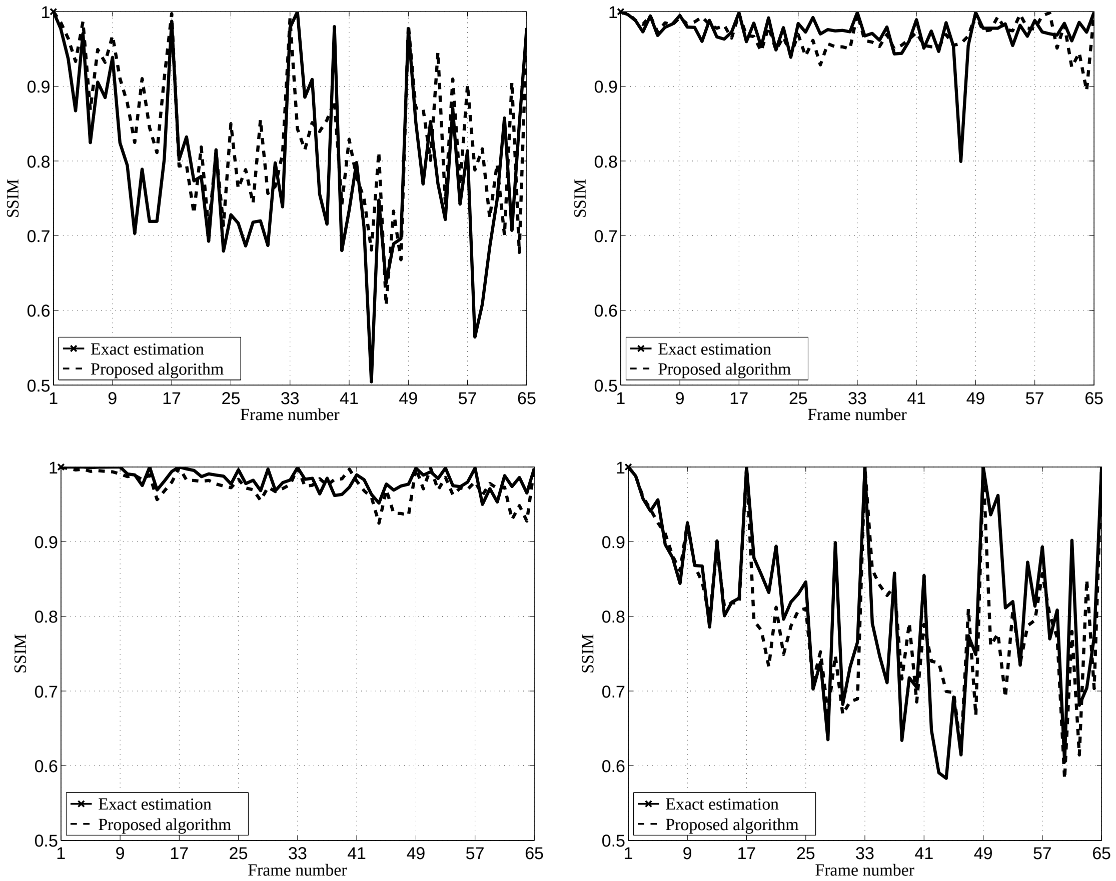

FIGURE 6.4: SSIM of the four received sequences in an 802.11e network with adaptive retransmission obtained using the exact distortion estimation and the proposed algorithm: (top-left) Foreman sequence, (top-right) Highway sequence, (bottom-left) News sequence and (bottom-right) Coastguard.

the retry limits [142]. In particular, by adopting the Markov approach in [110], the retry limits can be obtained by solving the set of equations:

$$\arg\min_{m_l\in\mathbb{N}} \left| p^{m_l+1} - 10^{-\frac{En(\mathcal{V})\cdot(1-\tilde{D}_{l,SSIM})}{\max(\mathcal{D}_*)}} \right|, \quad l = 1, ..., L \tag{6.11}$$

in the unknowns m_l by using $\mathcal{D}_* = \mathcal{D}$, when the exact estimation is adopted, and the set $\mathcal{D}_* = \mathcal{D}_A$ when the approximated one is used. Moreover, $\mathcal{D}$ and $\mathcal{D}_A$ are respectively the set of exact and approximated complementary distortions obtained from $\tilde{\mathcal{D}}$, namely the set of exact SSIM-based distortions and $\tilde{\mathcal{D}}_A$. Fig. 6.4 is obtained considering a scenario in which 4 different videos (Foreman, Highway, News, and Coastguard) are transmitted from 4 sources to 4 destinations by adopting the exact SSIM-based distortion and the proposed algorithm. The received SSIM at each destination is derived by comparing the actually received sequence with the corresponding one. It is possible to observe that the adoption of $\mathcal{D}_A$ instead of $\mathcal{D}$ introduces a negligible difference in terms of SSIM for all the 4 videos, thus justifying the drastic reduction of the computational time provided by the presented solution.

6.4 Conclusions

A novel algorithm for a fast evaluation of the SSIM-based distortion of a video sequence encoded using the H.264 standard with hierarchical structure has been presented. The results show that the adoption of the proposed algorithm drastically reduces the time required by the estimation process, introducing an acceptable error, even for videos with low homogeneous motion. The quality of the approximation has been confirmed by an application involving an 802.11e network implementing adaptive retransmissions. The reduced computational cost that characterizes the proposed algorithm allows one to obtain a satisfactory estimation of the SSIM-based distortion values that may be suitable also for scenarios characterized by constraints on the packet delay, thus resolving the issue of the high computational cost for the evaluation of the SSIM, arose in Ch.5.

Chapter 7

Joint voice/video retry limit adaptation in 802.11e networks

A recent study released by the European Commission has revealed that Wireless Fidelity (WiFi) represents the most widespread technology for streaming applications [136], and that this hegemony is expected to further grow with the diffusion of Gigabit-WiFi and WiFi-enabled smartphones and tablets. To manage the audio/video contents in a distributed network, the 802.11e EDCA has introduced four ACs [77]. The parameters that characterize these ACs, that is, the TXOP, the AIFS, the minimum and maximum contention windows, are specified by the EDCA according to the adopted PHY layer. Instead, the maximum number of retransmissions is not subject to mandatory specifications. Hence, it has been exploited by several studies to control the distortion in contention-based scenarios [42, 86, 88, 90–92]. These studies provide effective retry limit estimations, considering network layer priorities [42], M/M/1 queueing systems [91], virtual buffers [92], machine learning techniques [86], relative priority indexing

The content of this chapter is based on F. Babich, M. Comisso, R. Corrado and F. Merazka, “Joint Adaptation of the Maximum Retry Limit of Voice and Video Flows”, to be submitted in *IEEE Communications Letters*.

[88], content-aware strategies [90], and Markov analyses [139]. The common characteristic of these proposals is that the retry limit adaptation involves a single AC. However, the VO/VI ACs may be often simultaneously active and some scheduling schemes explicitly split the audio/video sequence in two parts that are separately sent to the two higher priority ACs [137]. Therefore, an approach able to jointly adapt the retry limits of both VO/VI packets may represent a desirable advance to improve the quality level of WiFi streaming.

This letter addresses this issue by presenting a fast adaptive algorithm that jointly estimates the retry limits of VO/VI packets according to the distortion introduced by their possible loss. The algorithm, which is characterized by a low computational cost and can operate in both saturated and non-saturated traffic conditions, is validated adopting an 802.11n PHY layer, and is compared with an existing method that is fairly extended to provide itself joint VO/VI retry limit estimation.

The chapter is organized as follow. Section 7.1 formulates the problem. Section 7.2 presents the algorithm. Section 7.3 discusses the results. Section 7.4 summarizes the conclusions.

7.1 System model

Consider an 802.11e distributed network with N node pairs, that is, N sources and the corresponding N destinations. The considered ACs are numbered according to $q = 1$ (VO) and $q = 2$ (VI), hence the higher the q value, the lower the priority.

7.1.1 Audio/video traffic

Each source sends an audio sequence $\mathcal{S}_1$ encoded by the G.729 standard to obtain a set $\{s_1^k : k = 1, ..., K_1\}$ of K_1 packets [58], and a video sequence $\mathcal{S}_2$ encoded by the

H.264 standard to obtain a set $\{s_2^k : k = 1, ..., K_2\}$ of K_2 packets [59]. The loss of a generic packet s_q^k produces a distortion on the corresponding sequence $\mathcal{S}_q^k$ decoded in absence of s_q^k. This distortion is estimated as $D_q^k = \mathcal{Q}(\mathcal{S}_q, \mathcal{S}_q^k)$ [86,88], where the quality assessment measure $\mathcal{Q}(\cdot, \cdot)$ is based on the Perceptual Evaluation of Speech Quality (PESQ) if $q = 1$ [138], and on the structural similarity (SSIM) if $q = 2$ [35]. Thus, each packet s_q^k is associated to a D_q^k value. The packet arrival is described by a Poisson process [42, 91, 92], which, in the limiting case of very high arrival rate, leads to the saturated traffic scenario [88, 90].

7.1.2 Network model

The sources adopt the EDCA basic access for channel contention, thus the average time $\bar{T}$ required by a success is identical to that wasted because of a collision. For the two ACs of interest, the maximum backoff stages are $m'_1 = m'_2 = 1$ and the AIFS values are equal [77], thus the VO/VI ACs differ just for the corresponding minimum contention window W_q. Hence, during the access procedure, the reactivation of a backoff counter previously freezed because of channel occupation requires that the channel is sensed idle for an identical time, regardless that the packet belongs to a VO or VI AC. Besides, as shown in [139], the impact of the lower priority BE/BK ACs on the VO/VI ones is limited. These two characteristics of the addressed scenario imply that the Markov model developed in [123] for the distributed coordination function (single AC case) may still be used to analyze the EDCA procedure for the VO/VI ACs [139], without the need of inserting additional states. To avoid cumbersome repetitions, the entire mathematical derivation of the adopted EDCA model is hence not reported, rather focusing on the basic equations that will be involved in the development of the proposed adaptation algorithm.

The core of the model is a nonlinear system that provides, for $q = 1, 2$, the conditional collision probability p_q, the transmission probability τ_q, the packet arrival probability during the processing of a previous packet $\lambda_{q,1}$, and the packet arrival probability when the source is idle $\lambda_{q,2}$. According to [123], this system may be expressed in the following form:

$$\begin{cases} \tau_q = \left[W_q + \frac{1}{2} + \left(\frac{1-\lambda_{q,1}}{\lambda_{q,2}} - \frac{W_q}{2} \right) \frac{1-p_q}{1-p_q^{m_q+1}} \right]^{-1} & q=1,2 \quad (7.1) \\ p_q = 1-(1-\tau_1)^{N-2+q}(1-\tau_2)^{N-1} & q=1,2 \quad (7.2) \\ \lambda_{q,1} = 1-e^{-\lambda E_{ns} E_s} & q=1,2 \quad (7.3) \\ \lambda_{q,2} = 1-e^{-\lambda E_s} & q=1,2 \quad (7.4) \end{cases}$$

where m_q is the (unknown) retry limit, λ is the average arrival rate, ν is the slot time,

$$E_{ns} = \left(W_q - \frac{1}{2} \right) \frac{1-p_q^{m_q+1}}{1-p_q} - \frac{W_q}{2} \tag{7.5}$$

is the average number of backoff counter decrements for m_q retransmissions, and:

$$E_s = \bar{T} - (\bar{T} - \nu)(1-\tau_1)^N (1-\tau_2)^N \tag{7.6}$$

is the average number of slots between two consecutive backoff counter decrements. In particular, by (7.1), one accounts for the transmission attempts, which are carried out when the backoff counter becomes equal to zero [139]. By (7.2), one models the collisions, which occur if, at the beginning of the same slot, transmissions are attempted by at least two sources (external collision), or by two ACs of the same source (internal collision). In this second case, the VO packet is transmitted and the VI packet is considered collided [77]. By (7.3), one expresses the probability that a packet arrives at the transmission queue of the source during the time $E_{ns}E_s$ spent for the access procedure of a previous packet [123]. Similarly, (7.4) expresses the arrival probability during E_s when the queue of the source is empty. Further mathematical details concerning the derivation of (7.1)-(7.6) may be found in [123].

7.1.3 Problem formulation

The availability of the collision and transmission probabilities enables the evaluation of the performance figures of the network. In particular, in retry limit adaptation problems, the interest focuses on the drop probability $p_q^{m_q+1}$. The basic idea is to guarantee a lower drop probability (higher retry limit) to a packet s_q^k that, whether lost, produces a higher distortion D_q^k, and a higher drop probability (lower retry limit) to a packet whose loss produces a lower distortion [86, 88, 90–92]. This requirement may be satisfied by selecting, for each VO/VI packet, a retry limit that minimizes the difference between the drop probability and the reciprocal of the distortion.

Mathematically, the objective is hence that of finding p_q for $q = 1, 2$ by (7.1)-(7.6) to then evaluate:

$$\hat{m}_q^k = \arg \min_{m_q \in \mathbb{N}} \left| p_q^{m_q+1} - \frac{1}{D_q^k} \right|, \tag{7.7}$$

for $k\!=\!1, ..., K_q$ and $q\!=\!1, 2$.

7.2 Algorithm

To deal with the formulated problem, (7.1)-(7.6) may be suitably solved by distinguishing between non-saturated and saturated traffic conditions.

7.2.1 Non-saturated traffic

In the non-saturated case, the arrival probabilities are lower than one. Therefore, one may adopt the approximation $e^x \cong 1+x$ in (7.3) and (7.4), thus obtaining:

$$\lambda_{q,1} \cong \lambda E_{ns} E_s, \tag{7.8a}$$

$$\lambda_{q,2} \cong \lambda E_s, \tag{7.8b}$$

for $q\!=\!1,2$. Substituting (7.8) and (7.5) in (7.1), and performing some algebra, one can write:

$$\tau_q \cong \left[1 + \frac{1}{\lambda E_s}\frac{1-p_q}{1-p_q^{m_q+1}}\right]^{-1}, \tag{7.9}$$

for $q = 1, 2$. According to the non-saturation hypothesis, one may reasonably assume $p_1, p_2 << 1$ and low m_q values, which imply $(1-p_q)/(1-p_q^{m_q+1}) \cong 1$ in (7.9) and hence $\tau_2 \cong \tau_1$. Now, applying these approximations and substituting (7.6) in (7.9), one obtains, after some calculations, the algebraic equation:

$$\pi_\beta(t) = t^\beta - \frac{\lambda\bar{T}+1}{\lambda(\bar{T}-\nu)}t + \frac{1}{\lambda(\bar{T}-\nu)} = 0, \tag{7.10}$$

of degree $\beta = 2N+1$ in the unknown $t = 1-\tau_1$. Since β is odd, $\pi_\beta(t)$ has at most three real roots. Besides, $\pi_\beta(t) \to \pm\infty$ for $t \to \pm\infty$, and, furthermore, $\pi_\beta(0) > 0$ and $\pi_\beta(1) < 0$, since $(\bar{T}) > \nu > 0$ and $\lambda > 0$. Therefore, $\pi_\beta(t)$ has exactly three real roots, with a unique one $\bar{t}$ in the interval $[0, 1]$. This latter root enables the calculation of the desired conditional collision probabilities in the non-saturated case. In fact, recalling (7.2) for $1 - \tau_2 \cong 1 - \tau_1 \cong \bar{t}$, yields:

$$p_q = 1 - \bar{t}^{2N-3+q}, \quad q\!=\!1,2. \tag{7.11}$$

7.2.2 Saturated traffic

In the saturated case, the arrival probabilities become very close to one and the m_q values get higher, thus $\lambda_{q,h} \cong 1$ for $q, h = 1, 2$, and $(1-p_q)/(1-p_q^{m_q+1}) \cong (1-p_q)$. Using these approximations in (7.1) and substituting (7.2) in the resulting expressions, one may derive the pair of equations:

$$t_q = 1 - \frac{2}{W_q\left(1+t_1^{N-2+q}\,t_2^{N-1}\right)+1}, \quad q\!=\!1,2, \tag{7.12}$$

in the unknowns $t_q = 1 - \tau_q$ for $q = 1, 2$. The equation corresponding to $q = 1$ in (7.12) may be solved for t_2 as:

$$t_2 = \frac{1}{t_1} \left[\frac{1+t_1+2W_1(t_1-1)}{W_1(t_1-1)} \right]^{\frac{1}{N-1}} . \tag{7.13}$$

Now, substituting (7.13) in (7.12) for $q = 2$, and remembering that $W_2 = 2W_1$ [77], one may obtain, after some manipulations, the algebraic equation:

$$\begin{aligned} \pi_{\beta'}(t_1) = (\alpha_{00}-1)(t_1-1) \left[2\alpha_{10}t_1^3 - (\alpha_{31}-2)t_1^2 + \alpha_{21}t_1\right]^{N-1} \\ - (\alpha_{10}t_1 - \alpha_{11})(2\alpha_{10}t_1^2 - \alpha_{31}t_1 + \alpha_{20})^{N-1} = 0, \end{aligned} \tag{7.14}$$

of degree $\beta' = 3N - 2$, where the terms $\alpha_{ij} = 2^i W_1 + (-1)^j$ for $i = 0, ..., 3$ and $j = 1, 2$ are introduced to obtain a more compact representation. Regardless of whether β' be odd or even, $\mathrm{d}\pi_{\beta'}(t_1)/\mathrm{d}t_1 < 0$ for $t_1 \in [0, 1]$ (calculations are boring and are omitted). Moreover, $\pi_{\beta'}(0) > 0$ and $\pi_{\beta'}(1) < 0$, since $W_1 > 1$. Hence, $\pi_{\beta'}(t_1)$ has a unique root $\bar{t}_1$ in the interval $[0, 1]$. Also in this case, this specific root allows the derivation of p_q for $q = 1, 2$. In fact, remembering that $\bar{\tau}_q = 1 - \bar{t}_q$ for $q = 1, 2$, and using (7.13) in (7.2), one may evaluate the conditional collision probabilities in the saturated case as:

$$p_q = 1 - 2\bar{t}_1^{q-1} + \frac{\bar{t}_1^{q-1}(1+\bar{t}_1)}{W_1(1-\bar{t}_1)}, \quad q = 1, 2. \tag{7.15}$$

7.2.3 Retry limit estimation

Once the p_1 and p_2 values are estimated for both traffic scenarios, the retry limit of the generic packet s_q^k may be calculated according to the distortion D_q^k by solving (7.7).

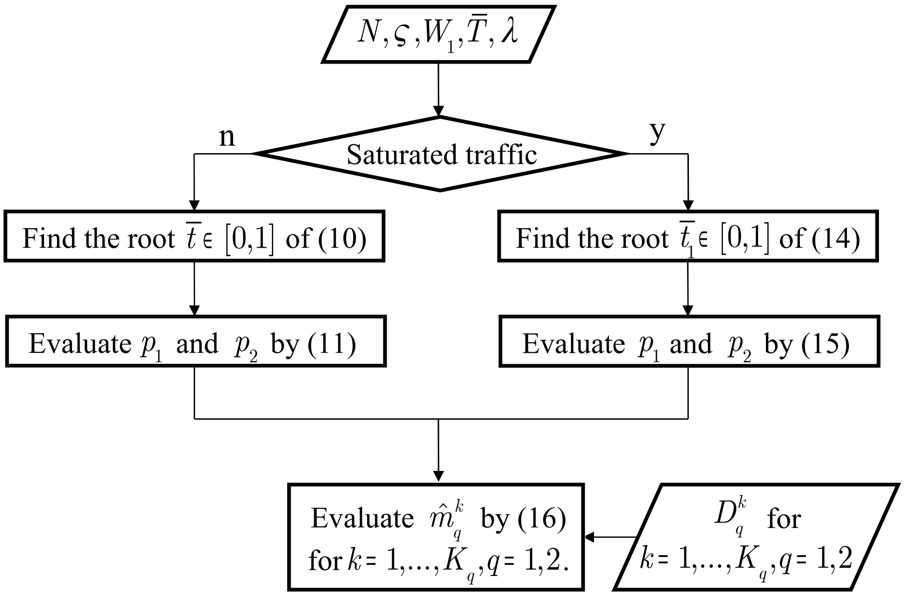
$N, \varsigma, W_1, \bar{T}, \lambda$
Saturated traffic
n
y
Find the root $\bar{t} \in [0,1]$ of (10)
Find the root $\bar{t}_1 \in [0,1]$ of (14)
Evaluate p_1 and p_2 by (11)
Evaluate p_1 and p_2 by (15)
Evaluate $\hat{m}_q^k$ by (16) for $k = 1,...,K_q, q = 1,2.$
D_q^k for $k = 1,...,K_q, q = 1,2$

FIGURE 7.1: Proposed algorithm.

This latter operation provides, for $k=1, ..., K_q$ and $q=1,2$, the closed-form expression:

$$\hat{m}_q^k = \left\lceil -\frac{\log(p_q D_q^k)}{\log p_q} \right\rceil, \tag{7.16}$$

where the ceiling function $\lceil\cdot\rceil$ has been adopted in place of the round one to guarantee a conservative overestimation of the distortion [139]. Now, all required elements are available.

Hence, given the quantities $N, \nu, W_1, (\bar{T}), \lambda$, and D_q^k for $k = 1, ..., K_q$ and $q = 1, 2$, which depend on the network traffic and on the selected PHY layer, the proposed retry limit adaptation algorithm develops as follows (Fig. 7.1). As a first step, find the root $\bar{t} \in [0,1]$ of (7.10) and then evaluate (7.11) (non-saturated case), or find the root $\bar{t}_1 \in [0,1]$ of (7.14) and then evaluate (7.15) (saturated case). As a second step, calculate $\hat{m}_q^k$ according to D_q^k by (7.16) for $k=1, ..., K_q$ and $q=1,2$.

Two are the main advantages of the proposed algorithm. Firstly, it is able to jointly estimate the retry limits of VO/VI packets in both saturated and non-saturated conditions. Secondly, the algorithm has a very low computational cost, since (7.10) and (7.14) may be quickly solved by efficient root-finding methods, and (7.11), (7.15), (7.16) are available in closed-form. This second advantage is emphasized by the fact that, for a given network scenario, a unique estimation of p_1 and p_2 is required for all VO/VI packets, because (7.10), (7.11), (7.14), (7.15) are independent of the specific packet.

7.3 Network simulations

In order to confirm the suitability of the approximations introduced for obtaining the collision probability, Figure 7.2 shows the comparison between the collision probability p_1, evaluated solving (7.1)-(7.6), that is performing an exact analysis, and the one evaluated solving (7.11), which represents the estimated p_1. In particular in Figure 7.2, p_1 is represented as a function of the average Poisson packet arrival. One may see that the exact and the approximated analysis provide results very close to each other,

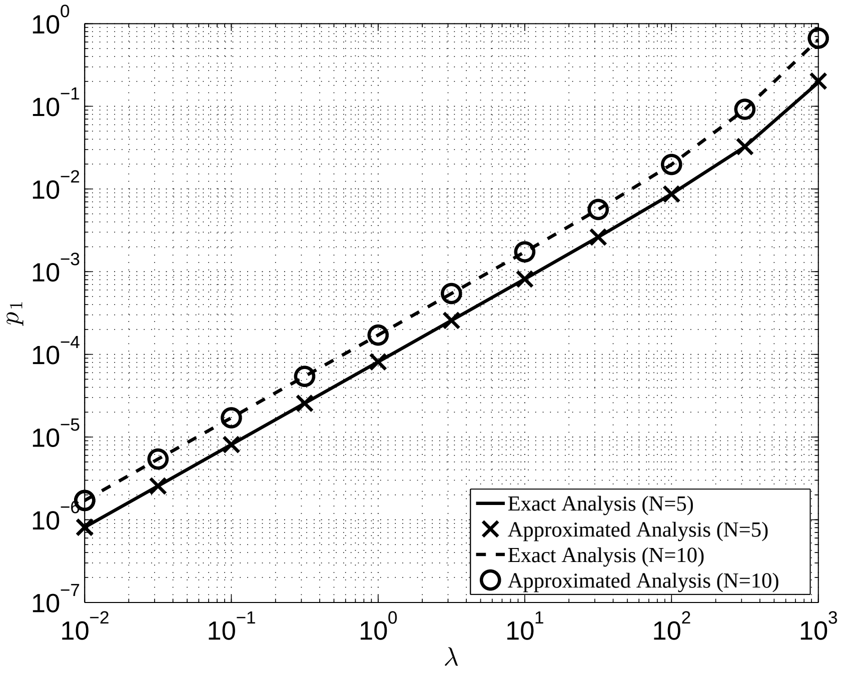

FIGURE 7.2: p_1, function of the Poisson arrival λ, obtained with an exact and an approximated analysis, for a scenario involving $N = 5, 10$ nodes with VO and VI queues active.

confirming the good quality of the approximations introduced to simplify the nonlinear system.

7.3.1 Results

This subsection focuses on the comparison between the performance in a distributed 802.11e/n-based network adopting, for the evaluation of the best retry limit for each packet in the VO and VI queues, the proposed method and an extended version of the algorithm presented in [88]. In particular, this second algorithm has been opportunely extended to operate on both audio and video ACs. The aim of this comparison is to evaluate the effectiveness of the proposed framework, both in the presence of saturated and non saturated traffic conditions. Each simulation has been carried out at packet-level implementing a Matlab state machine. The channel access contention process involves $N = 10$ nodes, each transmitting an audio and a video flow with data rate equal to

	2Q		**4Q**	
	Proposed Alg.	**Extended Alg. in [88]**	**Proposed Alg.**	**Extended Alg. in [88]**
PESQ	2.86	2.81	2.73	2.67
SSIM	0.93	0.91	0.91	0.82
CPU time $[s]$	0.60	0.60	0.60	0.60

TABLE 7.1: Averaged PESQ and SSIM in non saturated traffic conditions, with 2 active ACs (VO, VI) and with 4 active ACs, adopting the proposed algorithm and an extended version of [88]

	2Q		**4Q**	
	Proposed Alg.	**Extended Alg. in [88]**	**Proposed Alg.**	**Extended Alg. in [88]**
PESQ	1.70	1.04	1.70	1.04
SSIM	0.47	0.40	0.48	0.40
CPU time $[s]$	0.60	0.60	0.60	0.60

TABLE 7.2: Averaged PESQ and SSIM in saturated traffic conditions, with 2 active ACs (VO, VI) and with 4 active ACs, adopting the proposed algorithm and an extended version of [88]

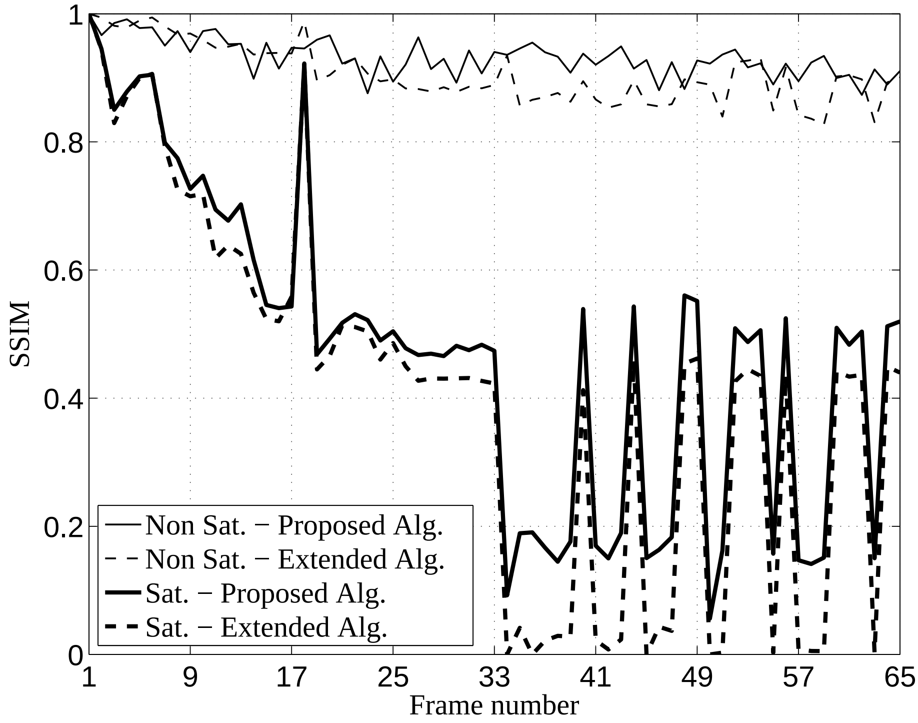

FIGURE 7.3: SSIM averaged over the frames for a scenario involving $N = 10$ nodes and the VO and VI queues active, in saturated and non saturated traffic conditions, adopting the proposed method and the extended version of [88]

120 Mbit/s. The adopted audio sequence is the 8 KHz audio sequence, The Barber of Seville, coded with the standard G.729. The video sequence is the Foreman sequence, coded with the standard H.264/SVC. The average Poisson arrival λ has been considered equal to 10^2. Since in [88] the authors don't provide the collision probability, the one estimated with the presented framework is adopted to evaluate the retry limit of each packet with the algorithm extended from [88]. The evaluation of the retry limit with the proposed method instead, has been carried out using (7.16). The results are shown in Tabs. 7.1 and 7.2 where the average PESQ and SSIM values for the tested scenarios are reported. From the two tables, one may notice that the presented solution allows one to obtain better results both in terms of PESQ for audio flows and SSIM for video flows. The results show that the improvements obtained adopting the proposed method become more evident with the increase of the collision probability. In particular, as λ becomes higher, the effectiveness of adopting the proposed algorithm become very relevant. Furthermore, the reduction in the CPU time required for the evaluation of the

collision probabilities for the VO and VI ACs, compared to the resolution of the $4Q$ equations system in (7.1)-(7.6), is very significant. In fact, the theoretical calculation required always more then 60 second for each tested scenario, making the theoretical approach not suitable for transmissions where the delay represents a stringent constraint. In Figure 7.3 we can see a comparison between the SSIM of the frames of the received video sequence, averaged over the number of simulations and nodes. These values are obtained adopting $N = 10$ with VO and VI queues active in non saturated and saturated traffic conditions. Figure 7.3 confirms how much has been observed from the results in Tabs. 7.1 and 7.2. As the traffic load increases due to a higher λ, the improvements obtained by using the proposed method are more relevant, and become very relevant in saturated traffic conditions.

7.4 Conclusions

A fast method for jointly evaluating the best retry limit for VO and VI packets, in an 802.11e distributed network has been presented. The method relies on a Markov model of the EDCA procedure of the 802.11e standard, developed both for saturated and non saturated traffic conditions. The estimated collision probabilities have been derived to set the retry limit of the packets of the VO and VI queues in agreement with the distortion associated to each packet. The results have been compared with the ones obtained adopting a properly extended algorithm. The comparison has revealed the improvements achievable by using the developed method, which become very significant when the traffic load increases. Furthermore, the proposed method is characterized by a very low computational cost, thus making the present solution suitable for scenarios characterized by devices with limited computational resources.

Conclusions

The objective of this thesis has focused on the audio and video transmission over wireless networks adopting the family of the IEEE 802.11x standards. In particular, this thesis has discussed four issues: the adaptive retransmission of video packets, the comparison of video quality indexes with retry limit adaptation purposes, the fast estimation of the video SSIM based distortion and the joint adaptation of the maximum number of retransmissions of voice and video flows.

Initially, a fast and simple retry limit adaptation method for video streaming applications over 802.11e distributed networks in the presence of distortion and delay requirements has been presented. The method relies on an properly approximated model of the network, whose reliability has been validated by numerical simulations, capable to describe its evolution, which have allowed to considerably reduce the computational cost of the conceived solution. The presented results have shown that the algorithm is able to accurately account for the impact of the higher priority VO AC on the video transmission, while simultaneously maintaining the frame delay below the video expiration time. The satisfactory performance has been reached maintaining a really low processing time for the retry limit estimation process. Furthermore, aiming to exploit the possibility of improving the video distortion evaluation for adaptive retransmission purposes, a study of the influence on the MSE and the SSIM indexes on the performance of an 802.11e network using an adaptive retransmission policy, has been presented. The numerical results have shown that the adoption of an SSIM-based approach is able to provide a better video quality at the end user, even in the presence of constraints on the total number of allowed retransmissions, but requiring a higher computational burden. Thus, to overcome the drawback introduced by the SSIM, a novel algorithm for a fast evaluation of the SSIM-based distortion of a video sequence encoded using the H.264 standard with hierarchical structure has been presented. The comparison between the exact SSIM values, and the ones obtained by the proposed algorithm, has shown that the adoption of the proposed solution drastically reduces the time required by the estimation process, introducing an acceptable error, even for videos with low homogeneous motion. The reduced computational cost that characterizes the proposed algorithm allows one to obtain a satisfactory estimation of the SSIM-based distortion values resolving the drawback of the delay introduced for the calculation of the SSIM. Finally, with the aim to extend the scenario to the transmission of audio contents, and considering also the complementary non saturated scenarios, a fast method for jointly evaluating the best retry limit for each VO and VI packet, has been presented. This solution relies on a Markovian model of the EDCA channel access procedure of the 802.11e standard. The study confirmed the improvements introduced by using the proposed method, in comparison with an extension of the solution present in scientific literature. These improvements become considerable in accordance with the increase of the traffic load, since they require a drastically lower computational cost to improve the network performance when audio/video streaming applications are involved.

Bibliography

[1] Netflix. `https://www.netflix.com`.

[2] Hulu. `https://www.hulu.com`.

[3] Youtube. `https://www.youtube.com`.

[4] M. Gast. *802.11ac: A survival guide*. O'Reilly Media, 2013.

[5] itunes. `https://www.apple.com/itunes`.

[6] S. Parekh, A. Sasan, R. Jain and M. Tofighbakhsh. *Quality of service architectures for wireless networks: Performance Metrics and Management*. IGI Global, 2010.

[7] M. van der Schaar and P. A. Chou. *Multimedia over IP and wireless networks: compression, networking, and systems*. Elsevier, 2007.

[8] D. Kim, P. Kroon, A. W. Rix, J. G. Beerends and O. Ghitza. Objective assessment of speech and audio quality—technology and applications. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14, No. 6, pp. 1890 – 1901, Nov. 2006.

[9] *ITU-T P.800 (08/1996): Methods for subjective determination of transmission quality*.

[10] M. A. Clements, S. R. Quackenbush and T. P. Barnwell. *Objective measures of speech quality*. Prentice Hall, 1988.

[11] *ITU-T BS.1116: Methods for the subjective assessment of small impairments in audio systems including multichannel sound sSystems.*

[12] *ITU-T SG12: A subjective/objective test protocol for determining the conversational quality of a voice link.*

[13] *ITU-T Reccomandation P.800.1: Mean Opinion Score (MOS) terminology.*

[14] E. Jones, D. Campbell and M. Glavin. Audio quality assessment techniques - a review, and recent developments. *Elsevier Signal Processing*, Vol. 89, No. 8, pp. 1489 – 1500, Mar. 2009.

[15] J. G. Beerends and J. A. Stemerdink. A perceptual speech-quality measure based on a psychoacoustic sound representation. *Audio Engineering Society Journal (JAES)*, Vol. 42, No. 3, pp. 115 – 123, Mar. 1994.

[16] *ITU-T Reccomandation P.861: Objective quality measurement of telephone-band (300-3400 Hz) speech codecs.*

[17] S. Voran. Objective estimation of perceived speech quality-part I: Development of the measuring normalizing block technique. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 7, No. 4, pp. 371 – 382, Jul. 1999.

[18] *ITU-T Reccomandation BS.1387: Method for objective measurements of perceived audio quality.*

[19] R. Reynolds, A.W. Rix and M.P. Hollier. Perceptual measurement of end-to-end speech quality over audio and packet-based networks. In *106th Audio Engineering Society Convention*, pp. 1 – 4.

[20] A.W. Rix and M.P. Hollier. The perceptual analysis measurement system for robust end-to-end speech quality assessment. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1515 – 1518, 2000.

[21] A.W. Rix, J. G. Beerends, A.P. Hekstra and M.P. Hollier. Perceptual evaluation of speech quality (PESQ) the new itu standard for end-to-end speech quality assessment part II: Psychoacoustic model. *Audio Engineering Society Journal (JAES)*, Vol. 50, No. 10, pp. 765 – 778, Oct. 2002.

[22] B. Furht and O. Marques. *Handbook of Video Databases: Design and applications*. CRC Press, 2003.

[23] *ITU-T Reccomandation BT.500-10: Methodology for the subjective assessment of the quality of Television pictures*.

[24] Z. Wang, L. Cormack, H.R. Sheikh and A.C. Bovik. Blind quality assessment for JPEG2000 compressed images. In *36th Asilomar Conference on Signals, Systems and Computers*, Vol. 2, pp. 1735 – 1739, Nov. 2002.

[25] M. Wada, S. Matsumoto, O. Sugimoto and R. Kawada. An objective measurement scheme for perceived picture quality degradation caused by mpeg encoding without reference pictures. In *Conference on Visual Communications and Image Processing*, 2001.

[26] M. H. Pinson, S. D. Voran, S. Wolf, A. A. Webster and C. T. Jones. An objective video quality assessment system based on human perception. *SPIE Human Vision, Visual Processing, and Digital Display* , pp. 15 – 26, 1993.

[27] Z. Wang and A.C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processig Magazine*, Vol. 26, No. 1, pp 98 – 117, Jan. 2009.

[28] A.M. Eskicioglu and P.S. Fisher. Image quality measures and their performance. *IEEE Transaction on Communications*, Vol. 43, No. 12, pp. 2959 – 2965, Dec. 1995.

[29] A.B. Watson. *Digital Images and Human Vision*. MIT Press, 1993.

[30] Z. Wang and A.C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, Vol. 9, No. 3, pp. 81 – 84, Mar. 2002.

[31] E. Peli. *Vision models for target detection and recognition*. The Schepens Eye Research Institute, Harvard Medical School, May. 1995.

[32] A.B. Watson. Dctune: A technique for visual optimization of DCT quantization matrices for individual images. *Society for Information Display Digest of Technical Papers XXIV*, pp. 946 – 949, 1993.

[33] S. A. Karunasekera and N. G. Kingsbur. A distortion measure for blocking artifacts in images based on human visual sensitivity. *IEEE Transaction on Image Processing*, Vol. 4, No. 6, pp. 713 – 724, Jun. 1995.

[34] C.H. Chou and Y.C. Li. A perceptually tuned subband image coder based on the measure of just-noticeable-distortion profile. *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 5, No. 6, pp. 467 – 476, Dec. 1995.

[35] H.R. Sheikh, Z. Wang, A.C. Bovik and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transaction on Image Processing*, Vol. 13, No. 4, pp. 600 – 612, Apr. 2004.

[36] G. Drettakis, N. Tsingos, E. Gallo and Projet Reves. Perceptual audio rendering of complex virtual environments. In *ACM Special Interest Group on GRAPHics and Interactive Techniques (SIGGRAPH 2004)*, pp. 249 – 258, 2003.

[37] J. Hahn, H. Fouad and J. P. Ballas. Perceptually based scheduling algorithms for real-time synthesis of complex sonic environments. In *4th International Conference on Auditory Display*, pp. 77 – 81.

[38] M. Wand and W. Strasser. Multi-resolution sound rendering. In *ACM Special Interest Group on GRAPHics and Interactive Techniques (SIGGRAPH 2004).*

[39] E. Hellerud, P. Svensson, and J.E. Voldhaug. Evaluation of packet loss distortion in audio signals. In *120 th Audio Engineering Society Convention*, May 2004.

[40] P. Frossard and O. Verscheure. Joint source/fec rate selection for quality-optimal mpeg-2 video delivery. *IEEE Transaction on Image Processing*, Vol. 10, No. 12, pp. 1815 – 1825, Dec. 2001.

[41] A. T. Campbell and G. Coulson. Qos adaptive transports: delivering scalable media to the desktop. *IEEE Network*, Vol. 11, No. 2, pp. 18 – 27, Mar. 1997.

[42] Q. Li and M. van der Schaar. Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation. *IEEE Transaction on Multimedia*, Vol. 6, No. 2, pp. 278 – 290, Apr. 2004.

[43] E. Masala, A. Fiandrotti, D. Gallucci and E. Magli. Traffic prioritization of H.264/SVC video over 802.11e ad hoc wireless networks. In *17th International Conference on Computer Communications and NetworksComputer Communications and Networks*, pp. 1 – 5, Aug. 2008.

[44] G. Bozdagi, E. Gurses and N. Akar. Selective frame discarding for video streaming in TCP/IP networks. In *Packet Video Workshop*, Apr. 2003.

[45] J. M. Ho, S. H. Chang, R. I. Chang and Y. J. Oyang. A priority selected cache algorithm for video relay in streaming applications. *IEEE Transaction on Broadcasting*, Vol. 53, pp. 79 – 91, Mar. 2007.

[46] N. Feamster and H. Balakrishnan. Packet loss recovery for streaming video. In *12th International Packet Video Workshop*, 2002.

[47] W. Kellerer, W. Tu, E. Steinbach, P. Svensson and J.E. Voldhaug. Rate distortion optimized video frame dropping on active networks nodes. In *14th International Packet Video Workshop (PV2004)*, Dec. 2004.

[48] X. Niu, Y. Zhang Z. Li, J. Chakareski and W. Gu. Modeling and analysis of distortion caused by markov-model burst packet losses in video transmission. *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 19, No. 7, pp. 917 – 931, Jul. 2009.

[49] E. Filippi, P. Bucciol, E. Masala and J. C. De Martin. Cross-layer perceptual ARQ for video communications over 802.11e wireless networks. *Advances in Multimedia*, Jan. 2007.

[50] M. Baldi, J. C. De Martin, E. Masala and A. Vesco. Quality-oriented video transmission with pipeline forwarding. *IEEE Transactions on Broadcasting*, Vol. 54, No. 3, pp. 542 – 556, Sep. 2008.

[51] M. D'Orlando, F. Babich and F Vatta. Video quality estimation in wireless IP networks: Algorithms and applications. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, Vol. 4, No. 1, pp. 972 – 985, Jan. 2008.

[52] R. Zhang, S.L. Regunathan and K. Rose. Video coding with optimal inter/intra-node switching for packet loss resilience. *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 6, pp. 966 – 976, Jun. 2000.

[53] K. Schmidt and K. Rose. First-order distortion estimation for efficient video streaming at moderate to high packet loss rates. *Packet Video*, Vol. 18, No. 6, pp. 318 – 325, 2007.

[54] *ISO/IEC 8802-11:1999(E), ANSI/IEEE Standard 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.*

[55] *IEEE Standard 802.11b-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-speed physical layer extension in the 2.4 GHz Band.*

[56] Cisco systems. `https://www.cisco.com`.

[57] *ISO/IEC IS 10918-1, Reccomandation ITU-T T.81: JPEG Requirements and guidelines.*

[58] *ITU-T Recommendation G.729 : Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP).*

[59] *ITU-T Recommendation H.264: Advanced video coding for generic audiovisual services.*

[60] M. Bosi and R. E. Goldberg. *Introduction to digital audio coding and standards*. Springer, 2003.

[61] *ITU-T Recommendation G.711 : Pulse code modulation (PCM) of voice frequencies.*

[62] J. Johnston, Y. Mahieux, K. Brandenburg, J. Herre and E.F. Schroeder. ASPEC. Adaptive spectral perceptual entropy coding of high quality music signals. In *Audio Engineering Society (Convention)*, 1991.

[63] G. Stoll, Y.V. Dehery and L. v.d. Kerkhof. Musicam source coding for digital sound. In *17th International Television Symposium*, pp. 612 – 617, Jun. 1991.

[64] P. Noll. MPEG digital audio coding. *IEEE Signal Processing Magazine*, Vol. 14, No.5, pp. 59 – 81, Sep. 1997.

[65] D. Pan. A tutorial on MPEG/audio compression. *IEEE Multimedia*, Vol.2, No. 2, pp. 60 – 74, 1995.

[66] Vorbis. `https://xiph.org/Vorbis`.

[67] Ogg. `https://xiph.org/Ogg`.

[68] K. Brandenburg. Mp3 and aac explained. In *17th International Conference: High-Quality Audio Coding*, pp. 612 – 617, Aug. 1999.

[69] Windows media audio (wma). https://msdn.microsoft.com/en-us/library/windows/desktop/gg153556(v-vs.85).aspx.

[70] I.E. Richardson. *The H.264 advanced video compression standard*. Wiley, 2010.

[71] *ISO/IEC 14496-10:2009, Advanced Video Coding for generic audio-visual services*.

[72] D. Marpe, H. Schwarz 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.

[73] x.265. https://x265.org/hevc-h265.

[74] V.S.Bagad and I.A. Dhotre. *Data Communication and Networking*. Pune, 2010.

[75] M. Gerlach. Assessing and improving privacy in vanets. In *4th Workshop on Embedded Security in Cars (ESCAR 2006)*, Nov. 2006.

[76] *IEEE Standard 802.11g-2003, Supplement to Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Further higher-speed physical layer extension in the 5 GHz band*.

[77] *IEEE Standard for high rate WPANs MAC/PHY specifications. Amendement 8: Medium Access Control (MAC) quality of service enhancements*.

[78] W. Stallings. *Data communication and networking*. Prentice Hall, 2011.

[79] *IEEE Standard 802.11a-1999, Supplement to Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-speed physical lLayer extension in the 5 GHz band*.

[80] *IEEE Standard 802.11n-2009:CI-502, Specific requirements part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications amendment: enhancement for higher throughput*.

[81] P.H. Pathak, Y. Zeng and P. Mohapatra. A first look at 802.11ac in action: Energy efficiency and interference characterization. In *2014 IFIP Networking Conference*, pp. 1 – 9, Jun. 2014.

[82] R. Zhang, L. Cai, J. Pan and X. Shen. Resource management for video streaming in ad hoc networks. *Elsevier Ad Hoc Networks*, Vol. 9, No. 4, pp. 623 – 634, Jun. 2011.

[83] Q. Ni, L. Romdhani and T. Turletti. A survey of QoS enhancements for IEEE 802.11 wireless LAN. *Wiley Wireless Communications and Mobile Computing*, Vol. 4, No. 5, pp. 547 – 566, Aug. 2004.

[84] S. Kumar, V.S. Raghavan and J. Deng. Medium access control protocols for ad hoc wireless networks: A survey. *Elsevier Ad Hoc Networks*, Vol. 4, No.3, pp. 326 – 358, May 2006.

[85] *IEEE Standard 802.11e-2005, Part 11: Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications amendment 8: Medium Access Control(MAC) quality of service enhancements.*

[86] M. van der Schaar, D.S. Turaga and R. Wong. Classification-based system for cross-layer optimized wireless video transmission. *IEEE Transactions on Multimedia*, Vol. 8, No. 5, pp. 1082 – 1095, Oct. 2006.

[87] M.H. Lu, P. Steenkiste and T. Chen. A time-based adaptive retry strategy for video streaming in 802.11 WLANs. *Wiley Wireless Communications and Mobile Computing*, Vol. 7, No. 2, pp. 187 – 203, Feb. 2007.

[88] Y. Zhang, Z. Ni, C.H. Foh and J. Cai. Retry limit based ULP for scalable video transmission over IEEE 802.11e WLANs. *IEEE Communications Letters*, Vol. 11, No. 6, pp. 498 – 500, Jun. 2007.

[89] J. L. Hsu and M. van der Schaar. Cross layer design and analysis of multiuser wireless video streaming over 802.11e EDCA. *IEEE Signal Processing Letters*, Vol. 16, No. 4, pp.0 268 – 271, Apr. 2009.

[90] C. M. Chen, C. W. Lin, and Y. C. Chen. Cross-layer packet retry limit adaptation for video transport over wireless LANs. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 20, No. 11, pp. 1448 – 1461, Nov. 2010.

[91] H. Bobarshad, M. van der Schaar, and M.R. Shikh-Bahaei. A low-vomplexity analytical modeling for cross-layer adaptive error protection in video over WLAN. *IEEE Transactions on Multimedia*, Vol. 12, No. 5, pp. 427 – 438, Aug. 2010.

[92] H. Bobarshad, M. van der Schaar, A.H. Aghvami, R.S. Dilmaghani and M.R. Shikh-Bahaei. Analytical modeling for delay-sensitive video over WLAN. *IEEE Transactions on Multimedia*, Vol. 14, No. 2, pp. 401 – 414, Apr. 2012.

[93] C. Greco, M. Cagnazzo, and B. Pesquet-Popescu. Low-latency video streaming with congestion control in mobile ad-hoc networks. *IEEE Transactions on Multimedia*, Vol. 14, No. 4, pp. 1337 – 1350, Aug. 2012.

[94] C.F. Kuo, N.W. Tseng and A.C. Pang. A fragment-based retransmission scheme with quality-of-service considerations for wireless networks. *Wiley Wireless Communications and Mobile Computing*, Vol. 13, No. 16, pp. 1450 – 1463, Nov. 2013.

[95] J. Jimenez, R. Estepa, F.R. Rubio and F. Gomez-Estern. Energy efficiency and quality of service optimization for constant bit rate real-time applications in 802.11 networks. *Wiley Wireless Communications and Mobile Computing*, Vol. 14, No. 6, pp. 583 – 595, Apr. 2014.

[96] P. Ameigeiras, J.J. Ramos-Munoz, J. Navarro-Ortiz and J.M. Lopez-Soler. Analysis and modelling of YouTube traffic. *Wiley Transactions on Emerging Telecommunications Technologies*, Vol. 23, No. 4, pp. 360 – 377, Jun. 2012.

[97] *Recommendation H.264: Advanced video coding for generic audiovisual services, annex G: scalable video coding*.

[98] T. Stutz and A. Uhl. A Survey of H.264 AVC/SVC encryption. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No. 3, pp. 325 – 339, Mar. 2012.

[99] D. Kandris, M. Tsagkaropoulos, I. Politis, A. Tzes and S. Kotsopoulos. Energy efficient and perceived QoS aware video routing over wireless multimedia sensor networks. *Elsevier Ad Hoc Networks*, Vol. 9, No. 4, pp. 591 – 607, Jun. 2011.

[100] X. Zhu and B. Girod. A unified framework for distributed video rate allocation over wireless networks. *Elsevier Ad Hoc Networks*, Vol. 9, No. 4, pp. 608 – 622, Jun. 2011.

[101] 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.

[102] S. H. Chang, R. I. Chang, J. M. Ho and Y. J. Oyang. A priority selected cache algorithm for video relay in streaming applications. *IEEE Transactions on Broadcasting*, Vol. 53, No. 1, pp. 79 – 91, Mar. 2007.

[103] Y. Wang, Z. Wu and J.M. Boyce. Modeling of transmission-loss-induced distortion in decoded video. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 6, pp. 716 – 732, Jun. 2006.

[104] F. Babich, M. Comisso, M. D'Orlando and F. Vatta. Distortion Estimation Algorithms (DEAs) for wireless video streaming. In *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1 – 5, Nov. 2006.

[105] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 3, pp. 535 – 547, Mar. 2000.

[106] P. Chatzimisios, A.C. Boucouvalas and V. Vitsas. Influence of channel BER on IEEE 802.11 DCF. *IET Electronics Letters*, Vol. 39, No. 23, pp. 1687 – 1688, Nov. 2003.

[107] G. R. Cantieni, Q. Ni, C. Barakat and T. Turletti. Performance analysis under finite load and improvements for multirate 802.11. *Elsevier Computer Communications*, Vol. 28, No. 10, pp. 1095 – 1109, Jun. 2005.

[108] D. Malone, K. Duffy and D. Leith. Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions. *IEEE/ACM Transactions on Networking*, Vol. 15, No. 1, pp. 159 – 172, Feb. 2007.

[109] B. Alawieh, C. Assi and H. Mouftah. Power-aware ad hoc networks with directional antennas: Models and analysis. *Elsevier Ad Hoc Networks*, Vol. 7, No. 3, pp. 486 – 499, May 2009.

[110] F. Babich, M. Comisso, M. D'Orlando and A. Dorni. Deployment of a reliable 802.11e experimental setup for throughput measurements. *Wiley Wireless Communications and Mobile Computing*, Vol. 12, No. 10, pp. 910 – 923, Jul. 2012.

[111] K. Kosek-Szott. A comprehensive analysis of IEEE 802.11 DCF heterogeneous traffic sources. *Elsevier Ad Hoc Networks*, Vol. 16, pp. 165 – 181, May 2014.

[112] J.W. Tantra, C.H. Foh and A.B. Mnaouer. Throughput and delay analysis of the IEEE 802.11e EDCA saturation. In *IEEE International Conference on Communications (ICC)*, Vol. 5, pp. 3450 – 3454, May 2005.

[113] N. Cranley and M. Davis. Video frame differentiation for streamed multimedia over heavily loaded IEEE 802.11e WLAN using TXOP. In *IEEE International*

Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1 – 5, Sep. 2007.

[114] A. Politis, I. Mavridis, A. Manitsaris and C. Hilas. X-EDCA: A cross-layer MAC-centric mechanism for efficient multimedia transmission in congested IEEE 802.11e infrastructure networks. In *IEEE International Conference on Wireless Communications and Mobile Computing (IWCMC)*, pp. 1724 – 1730, Jul. 2011.

[115] J.H. Hong, G.J. Min, K. Choi, J.K. Choi and J. Lee. Comparison of video streaming quality measurement methodologies. In *10th International Conference on Advanced Communication Technology (ICACT 2008)*, Vol. 2, pp. 993 – 996, Feb. 2008.

[116] K. Belahs, R. Pauliks, K. Tretjaks and R. Pauliks. A survey on some measurement methods for subjective video quality assessment. In *2013 World Congress on Computer and Information Technology (WCCIT)*, pp. 1 – 6, Jun. 2013.

[117] H.R. Sheikh, Zhou Wang, A.C. Bovik and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. URL `http://www.cns.nyu.edu/~lcv/ssim/`. Available from http://dopu.cs.auc.dk.

[118] E.P. Simoncelli and B. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, Vol. 24, pp. 1193 – 1216, May 2001.

[119] M. Yukawa, H. Watanabe, P. Ndajah, H. Kikuchi and S. Muramatsu. SSIM image quality metric for denoised images. In *Proceedings of the 3rd WSEAS International conference on Visualization, Imaging and Simulation*, pp. 53 – 57, 2010.

[120] Y.H. Tan, S. Rahardja, H.L. Tan, Z. Li and C. Yeo. A perceptually relevant MSE-based image quality metric. *IEEE Transactions on Image Processing*, Vol. 22, No. 11, pp. 4447 – 4459, Nov. 2013.

[121] J. Cornelis, M. Stoufs, A. Munteanu and P. Schelkens. Scalable joint source-channel coding for the scalable extension of H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 18, No. 12, pp. 1657 – 1670, Dec. 2008.

[122] F. Babich and M. Comisso. On the use of a nonuniform backoff in 802.11 networks. *IET Electornic Letters*, Vol. 46, No. 21, pp. 1468 – 1470, Oct. 2010.

[123] F. Babich, and M. Comisso. Throughput and Delay Analysis of 802.11-Based Wireless Networks Using Smart and Directional Antennas. *IEEE Transactions on Communications*, Vol. 57, No. 5, pp. 1413 – 1423, May 2009.

[124] I. Bouazizi. Size-distortion optimization for application-specific packet dropping: The case of video traffic. In *IEEE International Symposium on Computers and Communication (ISCC)*, Vol. 2, pp. 899 – 904, Jun. 2003.

[125] Q. Zhang, W. Zhu, C.C.J. Kuo, W. Kumwilaisak, Y.T. Hou and Y.Q. Zhang. A cross-layer quality-of-service mapping architecture for video delivery in wireless networks. *IEEE Journal on Selected Areas in Communications*, Vol. 21, No. 10, pp. 1685 – 1698, Dec. 2003.

[126] M.U. Demircin and P. van Beek. Bandwidth estimation and robust video streaming over 802.11e wireless LANS. In *IEEE International Conference on Multimedia and Expo (ICME 2005)*, pp. 1250 – 1253, Jul. 2005.

[127] H. Sangjin, K. Lee, Hendry, M. Kim and K. Park. A partial protection scheme based on layer dependency of scalable video coding. In *5th Internationa Conference on Visual Information Engineering (VIE 2008)*, pp. 777 – 782, Jul. 2008.

[128] G. Kormentzas, T. Pliakas and S. Tsekeridou. Joint scalable video coding and packet prioritization for video streaming over IP/802.11e heterogeneous networks. In *Proceedings of the 3rd international conference on Mobile multimedia communications (MobiMedia '07)*, No. 31, 2007.

[129] P. Nasiopoulos, H. Mansour and V. Krishnamurthy. Modeling of loss-distortion in hierarchical prediction codecs. In *IEEE International Symposium on Signal Processing and Information Technology*, pp. 536 – 540, Aug. 2006.

[130] Lai-Man Po, Chun-Ling Yang, Rong-Kun Leung and Zhi-Yi Mai. An SSIM-optimal H.264/AVC inter frame encoder. In *IEEE Internationa Conference on Intelligent Computing and Intelligent Systems (ICIS 2009)*, No. 31, pp. 291 – 295, Nov. 2009.

[131] W. Zhu, P. Wan, W. Hu, W. Dai, O.C. Au and J. Zhou. SSIM-based rate-distortion optimization in H.264. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7343 – 7347, May 2014.

[132] Y.H. Huang, T.S. Ou and H.H. Chen. SSIM-based perceptual rate control for video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 21, No. 5, pp. 682 – 691, May 2011.

[133] L.M. Po, C.L. Yang, R.K. Leung and Z.Y. Mai. Joint source-channel adaptation for perceptually optimized scalable video transmission. In *Global Telecommunications Conference (GLOBECOM 2011)*, No. 31, pp. 1 – 5, Dec. 2011.

[134] M. Filo, M. Kucharzak, J. Kibilda and R. Piesiewicz. The 802.11g relaying MAC that saves energy. In *2nd Baltic Congress on Future Internet Communications (BCFIC 2012)*, pp. 40 – 45, Apr. 2012.

[135] D.S. Turaga, M. van der Schaar and R. Wong. Classification-based system for cross-layer optimized wireless video transmission. *IEEE Transactions on Multimedia*, Vol. 8, No. 5, pp. 1082 – 1095, Oct. 2006.

[136] J.S. Marcus and J. Burns. Study on impact of traffic off-loading and related technological trends on the demand for wireless broadband spectrum. Technical Report MSU-CSE-00-2, EUC DG Content, 2013.

[137] N. Ghani, M. Peng, A.V. Vasilakos, Z. Wan, N. Xiong and L. Zhou. Adaptive scheduling for wireless video transmission in high-speed networks. In *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2011.

[138] *ITU-T Recommendation P.862. PESQ: Objective method for E2E speech quality assessment of narrowband telephone networks and speech codecs*, Feb. 2001.

Publications and conferences

Journal papers:

[139] F. Babich, M. Comisso, and R. Corrado, “Fast Retry Limit Adaptation for Video Distortion/Delay Control in IEEE 802.11e Distributed Network”, *Elsevier Ad Hoc Networks*, Vol. 36, No. 1, pp. 229 - 243, Jan. 2016.

[140] F. Babich, M. Comisso, R. Corrado and F. Merazka, “Joint Adaptation of the Maximum Retry Limit of Voice and Video Flows”, to be submitted in *IEEE Communications Letters*.

Conference proceedings:

[141] F. Babich, M. Comisso, and R. Corrado, “Adaptive Retry Limit Setting for Video Delay/Distortion Control in IEEE 802.11e Ad-Hoc Networks”, In *Italian Networking Workshop*, Cortina d’Ampezzo, Italy, pp. 1 - 5, 15 - 17 Jan. 2014.

[142] F. Babich, M. Comisso, and R. Corrado, “On the Impact of the Video Quality Assessment in 802.11e Ad-Hoc Networks Using Adaptive Retransmissions”, In *IEEE IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Piran, Slovenia, pp. 47 - 54, 2 - 4 Jun. 2014.

[143] F. Babich, M. Comisso, and R. Corrado, “Fast Distortion Estimation Based on Structural Similarity for H264/SVC Encoded Videos”, In *IEEE Vehicular Technology Conference (VTC)*, Glasgow, Scotland (UK), pp. 1 - 5, 11 - 14 May 2015.