UNIVERSITÀ DEGLI STUDI DI TRIESTE

DIPARTIMENTO DI ELETTROTECNICA, ELETTRONICA ED INFORMATICA

XIX Ciclo del
Dottorato di Ricerca in
Ingegneria dell'Informazione
(Settore scientifico-disciplinare: ING-INF/03)

TESI DI DOTTORATO

Multimedia over Wireless IP Networks: Distortion Estimation and Applications

Dottorando
MARCO D'ORLANDO

Coordinatore
Chiar.mo Prof. Alberto Bartoli
(Università degli Studi di Trieste)

FIRMA: .....

Tutore
Chiar.mo Prof. Fulvio Babich
(Università degli Studi di Trieste)

FIRMA: .....

Relatore
Chiar.mo Prof. Lucio Manià
(Università degli Studi di Trieste)

FIRMA: .....

Correlatore
Chiar.ma Dott.ssa Francesca Vatta
(Università degli Studi di Trieste)

FIRMA: .....

Abstract

This thesis deals with multimedia communication over unreliable and resource constrained IP-based packet-switched networks. The focus is on estimating, evaluating and enhancing the quality of streaming media services with particular regard to video services. The original contributions of this study involve mainly the development of three video distortion estimation techniques and the successive definition of some application scenarios used to demonstrate the benefits obtained applying such algorithms. The material presented in this dissertation is the result of the studies performed within the Telecommunication Group of the Department of Electronic Engineering at the University of Trieste during the course of Doctorate in Information Engineering.

In recent years multimedia communication over wired and wireless packet based networks is exploding. Applications such as *BitTorrent*, music file sharing, multimedia podcasting are the main source of all traffic on the Internet. Internet radio for example is now evolving into peer to peer television such as *CoolStreaming*. Moreover, web sites such as *YouTube* have made publishing videos on demand available to anyone owning a home video camera. Another challenge in the multimedia evolution is inside the house where videos are distributed over local WiFi networks to many end devices around the house. More in general we are assisting an all media over IP revolution, with radio, television, telephony and stored media all being delivered over IP wired and wireless networks. All the presented applications require an extreme high bandwidth and often a low delay especially for interactive applications. Unfortunately the Internet and the wireless networks provide only limited support for multimedia applications. Variations in network conditions can have considerable consequences for real-time multimedia applications and can lead to unsatisfactory user experience. In fact, multimedia applications are usually delay sensitive, bandwidth intense and loss tolerant applications. In order to overcame this limitations, efficient adaptation mechanism must be derived to bridge the application requirements with the transport medium characteristics. Several approaches have been proposed for the robust transmission of multimedia packets; they range from source coding solutions to the addition of redundancy

with forward error correction and retransmissions. Additionally, other techniques are based on developing efficient QoS architectures at the network layer or at the data link layer where routers or specialized devices apply different forwarding behaviors to packets depending on the value of some field in the packet header. Using such network architecture, video packets are assigned to classes, in order to obtain a different treatment by the network; in particular, packets assigned to the most privileged class will be lost with a very small probability, while packets belonging to the lowest priority class will experience the traditional best–effort service. But the key problem in this solution is how to assign optimally video packets to the network classes. One way to perform the assignment is to proceed on a packet-by-packet basis, to exploit the highly non-uniform distortion impact of compressed video. Working on the distortion impact of each individual video packet has been shown in recent years to deliver better performance than relying on the average error sensitivity of each bitstream element. The distortion impact of a video packet can be expressed as the distortion that would be introduced at the receiver by its loss, taking into account the effects of both error concealment and error propagation due to temporal prediction.

The estimation algorithms proposed in this dissertation are able to reproduce accurately the distortion envelope deriving from multiple losses on the network and the computational complexity required is negligible in respect to those proposed in literature. Several tests are run to validate the distortion estimation algorithms and to measure the influence of the main encoder-decoder settings. Different application scenarios are described and compared to demonstrate the benefits obtained using the developed algorithms. The packet distortion impact is inserted in each video packet and transmitted over the network where specialized agents manage the video packets using the distortion information. In particular, the internal structure of the agents is modified to allow video packets prioritization using primarily the distortion impact estimated by the transmitter. The results obtained will show that, in each scenario, a significant improvement may be obtained with respect to traditional transmission policies.

The thesis is organized in two parts. The first provides the background material and represents the basics of the following arguments, while the other is dedicated to the original results obtained during the research activity.
Referring to the first part in the first chapter it summarized an introduction to the principles and challenges for the multimedia transmission over packet networks. The most recent advances in video compression technologies are detailed in the second chapter, focusing in particular on aspects that involve the resilience to packet loss impairments. The third chapter deals with the main techniques adopted to protect the multimedia flow for mitigating the packet loss corruption

due to channel failures. The fourth chapter introduces the more recent advances in network adaptive media transport detailing the techniques that prioritize the video packet flow. The fifth chapter makes a literature review of the existing distortion estimation techniques focusing mainly on their limitation aspects.
The second part of the thesis describes the original results obtained in the modelling of the video distortion deriving from the transmission over an error prone network. In particular, the sixth chapter presents three new distortion estimation algorithms able to estimate the video quality and shows the results of some validation tests performed to measure the accuracy of the employed algorithms. The seventh chapter proposes different application scenarios where the developed algorithms may be used to enhance quickly the video quality at the end user side. Finally, the eight chapter summarizes the thesis contributions and remarks the most important conclusions. It also derives some directions for future improvements.

The intent of the entire work presented hereafter is to develop some video distortion estimation algorithms able to predict the user quality deriving from the loss on the network as well as providing the results of some useful applications able to enhance the user experience during a video streaming session.

List of acronyms

ADA	Advances Distortion Algorithm.
ARD	Accelerated Retroactive Decoding.
ARQ	Automatic Repeat Request.
ASO	Arbitrary Slice Ordering.
ASP	Advanced Simple Profile.
AEC	Adaptive temporal and spatial Error Concealment.
BER	Bit Error Rate.
CABAC	Context Adaptive Binary Arithmetic Coding.
CCA	Cross-Correlation Approximation.
CRC	Cyclic Redundancy Check.
CoDiO	Congestion Distortion Optimized streaming.
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance.
DAD	Direct Acyclic Dependency.
DAG	Direct Acyclic Graph.
DCT	Discrete Cosine Transform.
DEA	Distortion Estimation Algorithm.
DM	Distortion Matrix.
DSL	Digital Subscriber Line.
DTS	Decoding Time Stamp.
EC	Error Concealment.
EDA	Exponential Distortion Algorithm.
FEC	Forward Error Correction.
FMO	Flexible Macroblock Ordering.
FTP	File Transfer Protocol.
GDR	Gradual Decoding Refresh.
GoB	Group of Blocks.
GOP	Group Of Pictures.
HDTV	High Digital TeleVision.
JSVM	Joint Scalable Video Model.
JVT	Joint Video Team (of ITU-T VCEG and ISO/IECMPEG).
IDR	Instantaneous Decoding Refresh.

IETF	Internet Engineering Task Force Internet Protocol.
MAP	Markov Random Field.
MB	Macroblock (16 by 16 pixels).
MCP	Motion Compensated Prediction.
MDS	Maximum Distance Separable.
MMS	Multimedia Messaging Service.
MPEG-4	Motion Pictures Experts Group.
MSE	Mean Square Error.
MTU	Maximum Transmission Unit.
MV	Motion Vector.
NAL	Network Abstraction Layer.
NALU	Network Abstraction Layer Units.
P2P	Peer to Peer.
PBX	Private Branch eXchange.
PER	Packet Error Rate.
PFC	Previous Frame Concealment.
PLR	Packet Loss Rate.
PSNR	Peak to Signal Noise Ratio.
PTS	Presentation Time Stamp.
QoS	Quality of Service.
RaDiO	Rate Distortion Optimized streaming.
RM	Re-synchronization Marker.
ROI	Region Of Interest.
ROPE	Recursive Optimal per Pixel Estimate.
RPS	Reference Picture Selection.
RTCP	Real Time Control Protocol.
RTP	Real Time Protocol.
RVLC	Reversible Variable Length Coding.
SDA	Step Distortion Algorithm.
SP	Switching Predictive.
UEP	Unequal Error Protection.
VLC	Variable Length Coding.
VoD	Video on Demand.

Contents

Part I

Background

Chapter 1

Multimedia over Networks: a new revolution

Today multimedia communication over wired and wireless packet based networks is exploding. Applications such as BitTorrent [1], originally used for video downloads, now take up the lion's share of all traffic on the Internet. Music file sharing for example has moved into the mainstream with significant legal downloads of music and video to small devices such as cellular phones, smart phones, iPods and other portable media players. Multimedia podcasting to client computers and portable devices is a phenomenon exploding in its own right. Internet radio, pioneered in the late 1990s, is now evolving in peer to peer television such as CoolStreaming [2]. Audio and video on demand over the Internet, also available since the late 1990s on the Web sites of well-funded organizations such as `www.CNN.com`, is now the core of new music and video businesses from Napster [3] to the iTunes [4] service. Moreover, web sites such as YouTube [5] allowed publishing videos on demand available to anyone owning home video camera, which these days is nearly everyone owning a mobile phone. Indeed, most mobile phones today can actively download and upload photos and videos, sometimes in real time. Finally, Internet telephony is another multimedia service emerged, with popular applications such as Skype [6], VoIPStunt [7] and many other companies offering voice and video conference over the Internet. In general, Voice over IP (VoIP) is revolutionizing the voice telecommunications industry, as circuit-switched equipment from Private Branch eXchange (PBX) to long haul equipment is being replaced by soft IP switches such as Asterisk [8]. Enhanced television is also being delivered into the living room over IP networks by traditional telephone providers through Digital Subscriber Line (DSL) technologies as demonstrated in Italy by FastWeb [9].

Another challenge in the multimedia revolution takes place inside the house: the electronics manufacturers, the computer industry and its partners, are distribut-

ing audio and video over local WiFi networks to monitors and speakers around the house. Now that the analog-to-digital revolution is complete, we are assisting an all media over IP revolution, with radio, television, telephony and stored media all being delivered over IP wired and wireless networks.

The presented applications include an extreme high number of new multimedia related services but unfortunately the Internet and the wireless networks only provide limited support for multimedia applications. The Internet and wireless networks have unpredictable and variable conditions influenced by many factors. If averaged over time, this variability may not significantly impact delay insensitive applications such as file transfer. However, variations in network conditions can have considerable consequences for real-time multimedia applications and can lead to unsatisfactory user experience. In fact, multimedia are usually delay sensitive, bandwidth intense, and loss tolerant. These properties can change the fundamental principles of communication design for these applications. The concepts, theories and solutions that have traditionally been taught in information theory, communication, and signal processing may not be directly applicable to highly time-varying channel conditions and delay sensitive multimedia applications. As a consequence, in recent years, the area of multimedia communication and networking has emerged not only as a very active and challenging research topic, but also as an area that requires the definition of new fundamental concepts and algorithms that differ from those taught in conventional signal processing and communication theory.

1.1 Features of today Internet: mobile, home networks

The emergence of communication infrastructures such as the Internet and wireless networks enabled the proliferation of the above mentioned multimedia applications. These applications range from simple music download to a portable device, watching TV through the Internet on a laptop, or viewing movie trailers posted on the web through a wireless link. Some of these applications are new to the Internet revolution, while others may seem more traditional, such as sending VoIP to an apparently conventional telephone, sending television over IP to an apparently conventional set top box, or sending music over WiFi to an apparently conventional stereo amplifier. All these applications have different characteristics and requirements that will be discussed in the following paragraphs.

1.2 Streaming vs Downloading

Conventional downloading applications (e.g., file transfer such as File Transfer Protocol (FTP)) involve downloading a file before it is viewed or consumed by a user. Examples of such multimedia downloading applications are downloading an MP3 song to a portable device, downloading a video file to a computer through BitTorrent, or downloading a podcast from a web site. Downloading is usually a very robust way to deliver media to an end user. However, downloading has two potentially disadvantages for multimedia applications. First, a large buffer is required whenever a large media file (e.g., an entire MPEG-4 movie film) is being downloaded. Second, the amount of time required for the download can be relatively large, requiring the user to wait minutes or even hours before being able to view the content. Thus, while downloading is simple and robust, it provides only limited flexibility to users. An alternative to the downloading is the streaming. Streaming applications are able to split the media bitstream into separate chunks (usually referred as packets), which can be transmitted independently in the network, so that the receiver is able to decode and play back the parts of the bit stream that are already received. The transmitter continues to send multimedia data packets while the receiver decodes and simultaneously plays back other already received parts of the bit stream. This simple issue enables low delay between the instant when data is sent by the transmitter to the moment it is viewed by the user. Low delay is a fundamental property for interactive applications such as video conferences, but it is also important both for video on demand, where the user may desire to change channels quickly, and for live broadcast where the delay must be finite and relative low. The last advantage of streaming is its relatively low storage requirements and increased flexibility for the user compared to downloading. However, streaming applications, unlike downloading applications, have deadlines and other timing requirements to ensure continuous real-time media playout. This leads to new challenges for communication system designers who have to plan new strategies to support multimedia streaming applications.

1.3 Unicast, multicast and broadcast

Multimedia communication can be classified into one of three different categories: unicast, multicast and broadcast depending on the relationship between the number of senders and receivers. Unicast transmission connects one sender to one receiver. Examples of such applications include downloading, streaming media on demand and point-to-point telephony. A main advantage of unicast is that a feedback channel can be established between the receiver and the transmitter. When there is such a feedback channel, the receiver can return information to the

sender about the channel conditions, the end-user requirements, the end-device characteristics and so on, which can be used accordingly, by the transmitter, to adapt compression, error protection and other transmission aspects. Multicast transmission connects the sender to multiple receivers that decide to participate in the multicast session, over IP multicast or application level multicast (such as Peer to Peer (P2P)). Multicast is more efficient than multiple unicasts in terms of network resource utilization and server complexity. However, the main disadvantage of multicast, compared to unicast, is that the sender cannot target its transmission toward a specific receiver. Finally, broadcast transmission connects a sender to all receivers that it can reach through the network. An example is broadcast over a wireless link or a shared Ethernet link. As in multicast, the communication channel may be different for every receiver.

Chapter 2

Advances Compression Techniques: Video Coding and Decoding Strategies

2.1 Introduction

Video is becoming more and more popular for a large variety of applications and networks. Internet and wireless video, has became part of our life. However, despite many advances in terms of bandwidth and capacity enhancements in different networks, data transmission rate will always be limited due to physical limitation aspects, especially for high quality high rate applications. For this reason recent advance compression techniques are turning out to be very important. Furthermore real-time delivery of multimedia data is required in several applications such as conversational, streaming, broadcast or Video on Demand (VoD) services. Under the real-time constraints required, the Quality of Service (QoS) available in the current networks is in general inadeguate to guarantee error free delivery content to all the receivers. Therefore, in addition to the capability of easy integration into existing and future networks, video codecs must provide advances tools to cope with various transmission impairments. In particular video decoder must tolerate delay and packet losses. It is worth noticing that in every communication environment, standardized solutions are appreciated at terminals to ensure compatibility. That is the reason why video coding standards such as MPEG-4 and H.264/AVC became the most popular and attractive solution for many network environments and application scenarios.
These standards, like numerous previous standards, use a hybrid coding approach, namely Motion Compensated Prediction (MCP) that is combined with transform coding of the residual components. This chapter is centered on MCP-coded video

and the discussion mainly concentrates on tools and features integrated in the latest video coding standard H.264/AVC [10–11] and its test model software JM [12]. Instead of focusing on compression specific tools this chapter will focus on specific tools for improved error resilience within standard-compliant MCP-coded video. The encoding and decoding process based on MCP-coded video is discussed for example in [13] and in the Special Issue [14].

2.1.1 Applications of Video Streaming

As discussed in 1, digitally coded video is used in a wide variety of applications and in different environments. Video communications can be differentiated in unicast streaming, multicast and broadcast services of on line generated or pre-encoded content, video telephony and conferencing services as well as download and play services. These applications can operate in completely different bit-rate ranges. For example, High Digital TeleVision (HDTV) applications require data rates of about 20 Mbit/s, whereas simple download and play services such as Multimedia Messaging Services (MMS) on mobile devices might be satisfied with 20 kbit/s that is three orders of magnitude less. However, video applications have certain characteristics, which are of high importance for system design. For example, they can be distinguished by the maximum tolerable end to end delay and the possibility of online encoding or transcoding. This is in contrast with the transmission of pre-encoded content where the video content is stored in a server and is the end user that requires the media trough the Internet. In particular, the real-time services like broadcasting, unicast streaming, and conversational services have many different challenges because reliable delivery of all data can not be guaranteed in the existing networks. This is due to the fact that the feedback link in the actual systems is not always available or due to constraints on the maximum tolerable delay. Among these applications, conversational applications with end-to-end delay constraints of less than 200 to 250 ms are most challenging for the system design.

2.2 Video Communication System

The main components of an entire video communication system will be described accurately in the following sections with particular regard to the end user effects caused by the network impairments.

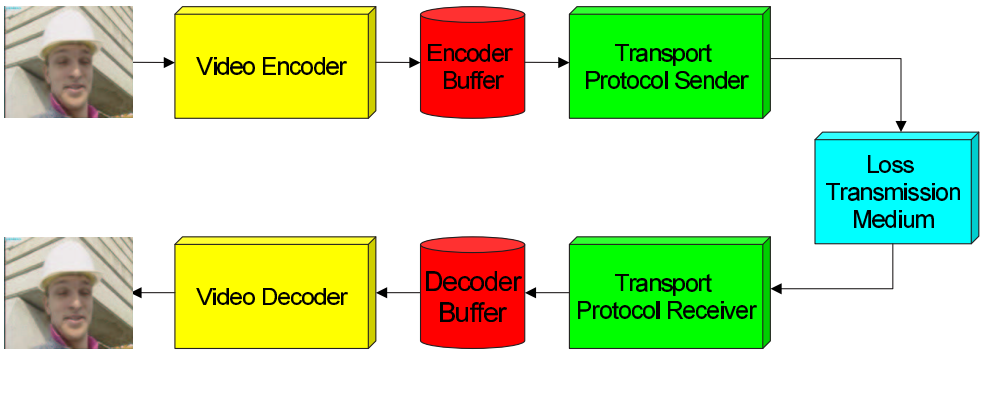

Figure 2.1: Lossy video transmission system.

2.2.1 End to End Video System

Fig. 2.1 provides an abstraction of a video transmission system that include the main components from the transmitter end to the receiver end. In order to keep this work focused, capturing, display devices, user interfaces, and security issues have been excluded; other computational complexity issues have also been ignored. Components that enhance system performance, e.g., a feedback channel, will be introduced as well, later in this chapter.

In contrast with still images, video frames include timing information, which has to be maintained to assure perfect reconstruction at the receiver's display. Furthermore, due to significant amount of spatio-temporal and psychovisual redundancy in natural video sequences, video encoders are able to reduce the amount of transmitted data significantly. However, excessive lossy compression results in noticeable, annoying, or even intolerable artifacts in the decoded video pictures. For this reason, a trade-off between *rate* and *distortion* is always necessary. Furthermore, real-time transmission of video adds additional challenges. In particular, according to Fig. 2.1, the video encoder generates data units containing the compressed video stream, which is stored in the encoder buffer before the transmission. The transmission system may delay, lose, or damage individual data units. Furthermore, each processing and transmission step adds some delay, which can be fixed, deterministic, or random. The encoder buffer and the decoder buffer are used to compensate the bit rates fluctuations produced by the encoder as well as channel delay variations to keep the end-to-end delay constant and to maintain the right timeline at the decoder. Nevertheless, in general the initial playout delay cannot be too excessive and strongly depends on the application constraints.

In contrast with analog audio, for example, compressed digital video cannot be accessed at any random point due to variable-length entropy coding as well the

syntax and semantics of the encoded video stream. In general, coded video data can be viewed as a sequence of data units, referred to as Access Units (AU) in MPEG-4 or Network Abstraction Layer Units (NALU) in H.264. The data units are self-contained on a syntactic level and can be labeled as data unit specific information; for example their relative importance for video reconstruction quality (slice P type, slice I type, slice B type). On the other hand due to spatial and temporal prediction the independent compression of data units cannot be guaranteed without significantly losing compression efficiency. A concept of Directed Acyclic Dependency (DAD) graphs on data units has been introduced in [15], which formalizes all these issues. The data units themselves are either directly forwarded to a packet network or encapsulated into a bit or byte stream format containing unique synchronization codes and then injected into a circuit-switched network.

2.2.2 Transmission over Error Prone Networks: Impairments

The process of introduction of errors and its effects are considerably different in wired or in wireless networks. For wireless networks, fading and interference cause *burst errors* in form of multiple lost packets. Moreover congestion can result in lost packets in an wired IP network. Nowadays, even for wireless networks, systems include tools able to detect the presence of errors in a packets on physical layer and the losses are reported to higher layers. These techniques usually use Cyclic Redundancy Check (CRC) mechanisms. By consequence the video decoder will not receive the entire bitstream. Intermediate protocol layers such as User Datagram Protocol (UDP) [16] might decide to completely drop erroneous packets so deleting all the encapsulated data units.

Furthermore, video data packets are treated as lost if they are delayed more than a tolerable threshold defined by the user video application. In both cases the end effect is the loss of the entire data units so the decoder needs to deal with such losses. Detailed description of the processes of losses in IP wireless based networks will be given in another section.

2.2.3 Data Losses in MCP coded Video

Fig. 2.2 presents a typical simplified version of an end to end video system when video, compressed using MCP, is transmitted over error prone channels. In this context t represents the time, s_t is a single video frame encapsulated in a P_t network packet, C_t indicate if P_t is correctly received or discarded, while $\overline{s_t}(C)$ represents the decoded frame as a function of the channel error pattern. Suppose that all Macroblocks (MBs) of one frame s_t are contained in a single packet P_t, for example in a Network Abstraction Layer (NAL) unit in the case of H.264/AVC.

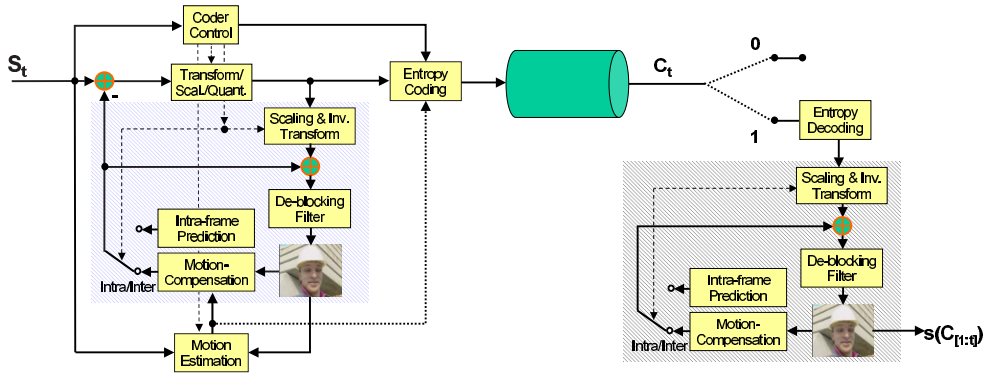

Figure 2.2: Simplified lossy video transmission system.

Furthermore, assume that this packet is transmitted over a channel that forwards correct packets to the decoder, and identify with $C_t = 1$ the correct reception of that packet while $C_t = 0$ means that the packet is corrupted and so discarded at the receiver.

In case of successful transmission the packet is forwarded to the normal decoder operation such as *Entropy Decoding* and *Motion compensated Prediction*. The prediction information and transform coefficients are reloaded from the coded bitstream to reconstruct the current frame $\overline{s}_{t-1}$. After that the frame is forwarded to the display buffer, and also to the reference frame buffer to be used in the MCP process to reconstruct the following inter-coded frames (i.e. the frame $\overline{s}_t$ frame). In the case that the packet P_t is lost, i.e. at the reference time t, $C_t = 0$, the so called Error Concealment (EC) is necessary to be enabled. In the simplest form, the decoder just skips the decoding operation and the display buffer is not updated and so the displayed frame is still $\overline{s}_{t-1}$. The viewer will immediately recognize the loss of motion since continuous display update is not maintained.

However, in addition to display buffer, the reference frame buffer is also not updated as a result of this data loss. Even in case of successful reception of packet P_{t+1}, the inter-coded frame $\overline{s}_{t+1}$, reconstructed at the decoder, will in general not be identical to the reconstructed frame s_{t+1} at the encoder side. The reason is: as the encoder and the decoder refer to a different reference signal in the MCP process resulting in a reconstruction mismatch. Therefore, there will be a mismatch in reference signal when decoding $\overline{s}_{t+2}$. For this reason it is obvious that the loss of a single packet P_t affects the quality of all the inter-coded frames: $\overline{s}_{t+1}$, $\overline{s}_{t+2}$, $\overline{s}_{t+3}$,....
This phenomenon is present in any predictive coding scheme and is called *error propagation*. If predictive coding is applied in the spatial and temporal domains

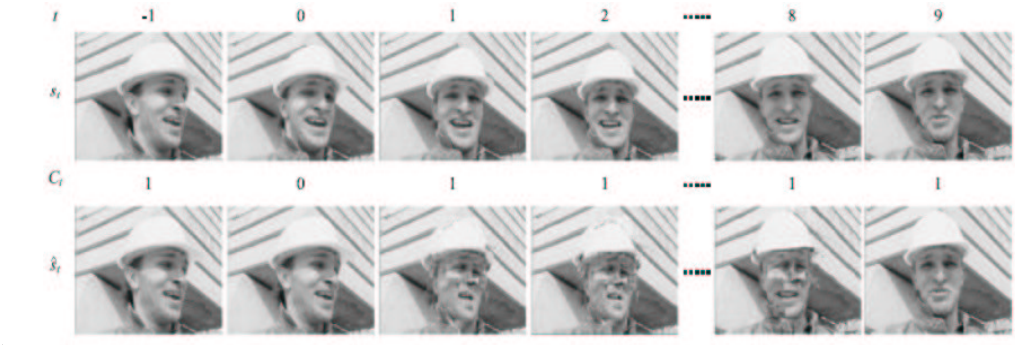

Figure 2.3: Example of *Error Propagation* effect in a hybrid video coding system due to a picture loss for the *Foreman* video test sequence.

of a sequence of frames, it is referred to as *spatio-temporal error propagation*.

Therefore, for MCP-coded video, the reconstructed frame at the receiver, $\overline{s_t}$, not only depends on the actual channel behavior C_t, but on the previous channel behavior $C_{[1:t]}$ and this dependency is evidenced with: $\overline{s}_t(C_{[1:t]})$. An example for error propagation is shown in Fig. 2.3. The top row presents the sequence with perfect reconstruction so without channel losses; in the bottom row instead only packet P_t at time $t = 1$ is lost. Although the remaining packets are again correctly received the error propagates and is still visible in decoded frame $\overline{s}_t = 8$. At time $t = 9$, the encoder transmits an intra-coded image, and since no temporal prediction is used for coding this frame, temporal error propagation is terminated at this time. It should be noted, however, that even with inter-coded images, the effect of a loss is reduced with every correct reception. This is because inter-coded frames might consist of intra-coded regions that do not use temporal prediction. An encoder might decide to do so when it finds that temporal prediction is inefficient for coding a certain image region. Following the intra image at $t = 9$, the decoder will be able to perfectly reconstruct the encoded images till another data packet is lost for $t > 9$.

Therefore, a video coding system operating in environments where data units might get lost, should provide one or more of the following features:

1. A mean that allows to avoid completely transmission errors;

2. Features that minimize the visual effects of errors in a reconstructed and displayed frame;

3. Features to limit spatial as well as spatio-temporal error propagation in hybrid video coding.

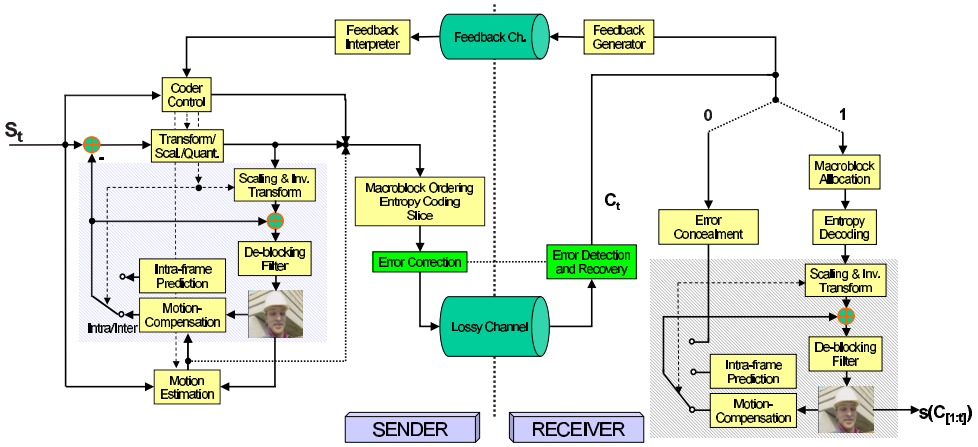

Figure 2.4: Lossy video transmission system.

The remainder of this chapter is restrict to forward predictive MCP video coding, though most of the concepts generalize to any kind of dependencies. A formal description of packetized video with slice structured coding, error concealment, as well as the extension of operational encoder control for error-prone video transmission, are discussed in Section 2.3.

2.3 Error Resilient Video Transmission

2.3.1 System Overview

The operation of an MCP video coding system in a transmission environment is depicted in Fig. 2.4. It extends the simplified schema illustrated in Fig. 2.2 by the addition of typical features used when transmitting video over error-prone channels. However, in general, for specific applications not all features are used, but only a suitable subset is extracted. Frequently, the generated video data belonging to a single frame is not encoded as a single data unit, but MBs are grouped in data units and the entropy coding is such that individual data units are syntactically accessible and independent. The generated video data might be processed in a transmission protocol stack and some kind of error control is typically applied, before the video data is transmitted over the lossy channel. Error control features include Forward Error Correction (FEC), Backward Error Correction (BEC), and any prioritization methods, as well as any combinations of those. At the receiver, it is essential that erroneous and missing video data are detected and localized. Commonly video decoders are able to treat only correctly received video data

units, or at least with an error indication, that certain video data has been lost. Video data units such as NAL units in H.264 are self-contained and therefore the decoder can assign the decoded MBs to the appropriate locations in the decoded frames. For those positions where no data has been received, error concealment needs to be applied. Advanced video coding systems also allow reporting the loss of video data units from the receiver to the video encoder. Depending on the application, the delay, and the accurateness of the information, an online encoder can exploit this information in the encoding process. Likewise, streaming servers can use this information in their decisions. Several of the concepts briefly mentioned in this high-level description of an error-resilient video transmission system will be elaborated and investigated more in detail in remaining sections.

2.3.2 Design Principles

Video coding features such as MB assignments, error control methods, or exploitation of feedback information can be used exclusively for error robustness purposes, depending on the application. It is necessary to understand that most error-resilience tools decrease compression efficiency because they add further information in the bitstream. Therefore, the main goal in transmitting video goes along with the spirit of Shannon's famous separation principle [17]:
"Combine compression efficiency with link layer features that completely avoid losses such that the two aspects, compression and transport, can be completely separated".
Nevertheless, in several applications and environments, for example in low delay situations, error-free transport may be impossible, in these cases the following system design principles are essential:

1. *Loss correction below codec layer*: Minimizing the amount of losses in the channel without sacrificing the video bit rate;

2. *Error detection*: If errors are unavoidable then erroneous video data will be detected and localized;

3. *Prioritization methods*: If losses are unavoidable then at least minimize losses for very important data are minimized;

4. *Error recovery and concealment*: In case of losses, the visual impact of losses on the actually distorted frame is minimized;

5. *Encoder-decoder mismatch avoidance*: In case of losses, encoder and decoder mismatch needs to limit or completely avoid the effects of the error propagation.

The following sections will focus especially on the latter three design principles. However, for completeness, we include a brief overview on the first two aspects leading to treat all these advanced issues.

2.3.3 Error Control Methods

In wireless systems, below the application layer, error control such as FEC and retransmission protocols are the primary tools for providing QoS. However, the trade-offs among reliability, delay, and bit rate have to be considered. Nevertheless, to compensate the shortcomings of non-QoS (best effort) networks such as the Internet or some mobile systems, error control features are introduced at the application layer. For example, broadcast services apply application-layer FEC schemes while point-to-point services use selective application layer retransmission schemes. For delay-uncritical applications instead, the Transmission Control Protocol (TCP) [18–19] can provide QoS. The topics of channel protection techniques and FEC will be covered in detail later. We will not deal with these features in this chapter, but we concentrate on video-related signal processing techniques that enhance reliability and improve QoS.

2.3.4 Video Compression Tools related to Error Resilience

Video coding standards such as H.263 [20], MPEG-4 [21], as well as H.264 only specify the decoder operation in case of reception of an error free bitstream as well as the syntax and semantics of the video bitstream. By consequence, the deployment of video coding standards still provides a significant amount of freedom for decoders that have to process erroneous bitstreams. Depending on the compression standard used, different compression tools, that offer some mechanisms for error resilient transmission and for robust decoding, are actually available.

Video compression tools have evolved significantly over time in terms of the error resilience they propose. Early video compression standards, as H.261 [22], had very limited error resilience capabilities. Later standards like MPEG-1 and MPEG-2 changed little in this regard since they were developed primarily for compression and storage applications, like Compact Disk (CD) or Digital Video Device (DVD) storage device, that does not require resilience capabilities [23]. With the introduction of H.263, the application starts to change dramatically. The resilience tools of the first version of H.263 [24] had only marginal improvements over MPEG-1, however later versions of H.263 (referred as H.263+ and H.263++ respectively) introduced several new tools that were developed specifically for the purpose of error resilience. These tools resulted in a popular acceptance of this codec; it replaced H.261 in most video communication applications giving possible the transmission of video over the nets. In parallel to this work, the new

emerging standard MPEG-4 Advanced Simple Profile (ASP) opted for an entirely different approach. Some sophisticated resilience tools like Reversible Variable Length Coding (RVLC) and Re-synchronization Markers (RM) were introduced [25]. However, despite their strong concept, all these tools did not gain a wide acceptance. One of the reasons why is that these tools try to solve the issues of lower ISO/OSI layers in the application layer, which is not a widely accepted approach. For example, RVLCs can be used at the decoder to reduce the impact of errors in a corrupted data packet. However, as discussed in Section 2.2.2, errors on physical layer can be detected and lower layers might discard these packets instead of forwarding them to the application.

The introduction of H.264/AVC has changed radically this behavior. This standard, in fact, is equipped with a wide range of error resilience tools. Some of these tools are modified and enhanced forms of these are introduced in H.263++. The following subsections give a brief overview of these tools as they are formulated in H.264/AVC, and the basic idea behind the introduction. Considering the rapid evolution of these tools, it is also important to know the origin of these tools in previous standards. Some specific H.264 error resilience features such as error-resilient entropy coding schemes and Arbitrary Slice Ordering (ASO) will not be discussed due to the complexity of the issues involved. A detailed description of these features is provided in in [14]. Some of the resilience tools have a dual purpose of increased compression efficiency along with error resilience, which seems to be initially contradictory although it is not. In the last part of this chapter, some of these tools will be considered in action in different applications measuring the utilization impact on system performance.

2.3.4.1 Slice Coding

For typical digital video transmission over networks it is not suitable to transmit all the compressed data belonging to a complete coded frame in a single data packet for many reasons. Primarily, variations are expected in sizes of certain such data packets because of a varying amount of redundancy in different frames of the sequence. In this case the lower layers have to fragment the packet to make it suitable for transmission. In case of a loss of a single fragment, the decoder might be unable to decode an entire frame with one only synchronization point available for an entire coded frame.

To overcome this problem, *slices* provide spatially distinct re-synchronization points within the video data for a single frame (Fig. 2.5). A number of MBs are grouped together, introducing a slice header which contains syntactic re-synchronization bits. The concept of slices (referred to as Group of Blocks (GOB) in H.261 and H.263) exists in different forms and in different standards, its usage was limited to encapsulate individual rows of MBs in H.263 and MPEG-2. In this case, slices

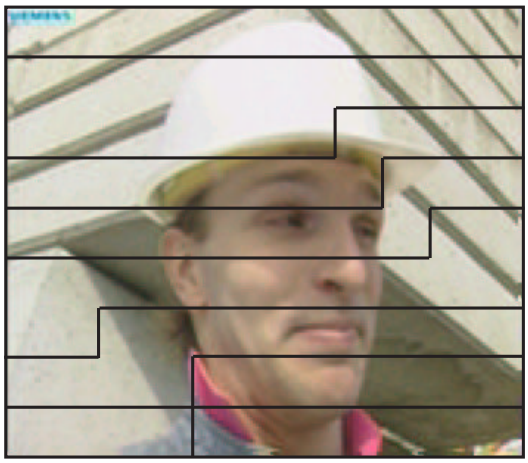

Figure 2.5: A picture divided into several slices. Slices are enhanced by the line boundaries.

will result in a variable data unit size because of the varying amount of redundancy in different regions of a frame. Slices were also introduced in H.264/AVC. The encoder selects the location of the synchronization points at any MB boundary. Intra prediction and motion vector prediction are not allowed over slice boundaries. In this way slices are independently decodable. Moreover, an arbitrary number of MBs can be assigned to a single slice using some configuration files, which results in different operation modes. For example, the encoder chooses to allocate either a fixed number of MBs or a fixed number of bits to a single slice. The later operation mode, with a pre-defined data slice size, is especially useful from a network perspective, since the slice size can be better matched to the packet size supported by the network layer. In this case, a loss of a data unit on network layer will result in a loss of a discrete number of slices, and a considerable portion of a picture might remain unaffected by the loss.

Hence in H.264/AVC, slices are the basic output of the video encoder and form an independently accessible entity. Provision of access to those units is either provided by the use of an unique synchronization marker or by the appropriate encapsulation in underlying transport protocols. The illustration of the slice mode operation is clearly described in Fig. 2.5.

2.3.4.2 Flexible Macro Block Ordering

In previous video compression standards like MPEG- 1, MPEG-2 and H.263 etc., MBs are processed and transmitted in raster scan order, that is starting from the

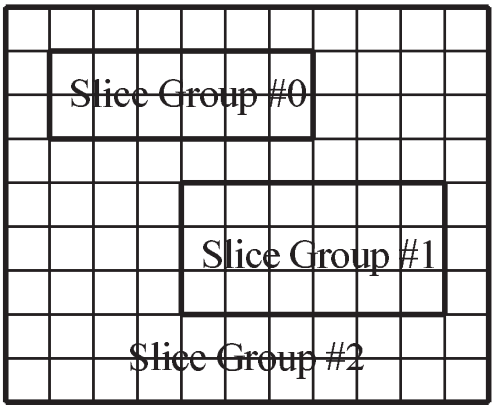

Figure 2.6: MBs of a picture allocated to 3 slice groups.

top-left angle of the image to the bottom right. However, if a data unit is lost, this results in the loss of a connected area in a single frame.

In order to allow a more flexible transmission order of MBs in a frame in H.264/AVC, Flexible Macroblock Ordering (FMO) [14] feature allows mapping MBs to so-called *slice groups*. A slice group may contain several slices. For example, in Fig. 2.6, each region (a slice group) might be subdivided into several slices. Hence, a *slice group* can be viewed as an entity similar to a picture consisting of slices in the case when FMO is not used. Therefore, MBs may be transmitted out of raster-scan order in an efficient way. This can be beneficial in several cases as detailed in the following:

1. Several concealment techniques at the decoder rely on the availability of correctly received neighbor MBs to conceal a lost MB. Hence, a loss of collocated image areas results in poor concealment performance. Using FMO, spatially collocated image areas can be interleaved in different slices. This simple mechanism will result in a high probability that neighboring MB data is available for concealing the lost MBs.

2. There might exist a Region Of Interest (ROI) within the images of a video sequence, e.g., the face of the caller in a video conference system. Such regions can be mapped into a separate slice group than the background to protect better video packets from losses in the network layer.

2.3.4.3 Scalability

Scalable coding usually refers to a source coder that simultaneously provides different encoded version of the same data source at different quality levels by ex-

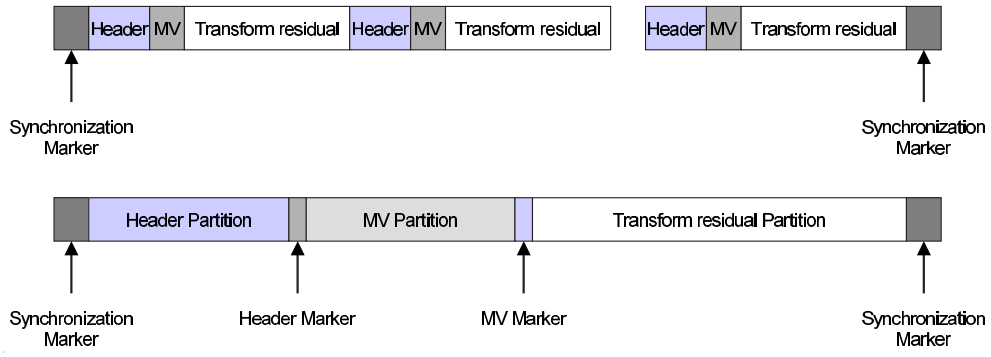

Figure 2.7: The layout of a compressed video data without using data partitioning (above) and with data partitioning (below) in H.263++. A packet starts with a synchronization marker, while for data partitioning mode two additional synchronization points are available, i.e., the header marker and the MV marker.

tracting a lower quality reconstruction from a single bitstream. Scalable coding can be realized using *embedded bitstreams*, i.e., the bit stream of a lower resolution is fitted into the bit stream of higher resolution layers. In general for video sources the quality can be changed in three main directions, namely the *spatial resolution*, the *temporal resolution* or frame rate, and the quantization distortion or *SNR scalability*. Scalable video coding is realized in standards in many different variants. Commonly, scalability is synonymously used with a specific type of scalability named *successive refinement*. This specific case addresses the point of view that information is added such that the initial reproduction is refined. A new standard compatible with H.264/AVC has already started that perform encoding specification in order to gain benefits from scalability, the new model is called Joint Scalable Video Model (JSVM) [26] and provides primarily support for temporal scalability, spacial, and SNR scalability.

2.3.4.4 Data Partitioning

The concept of Data Partitioning (DP) begins with the fact that loss of some syntax elements of bitstream result in a larger degradation of quality due to lack of redundancy during Variable Length Coding (VLC) or CABAC decoding. For example, the loss of MB mode information or Motion Vector (MV) information will for the most of cases result in a larger distortion compared to loss of a high frequency transform coefficients. This is intuitive since, for example, MB mode information is required for decoding all the remaining dependent MBs.

In the case of MB data loss, data partitioning results in the so-called graceful degradation of video quality. Graceful degradation targets reduction of perceived

video quality that is approximatively proportionate to the amount of data lost. In this case, the emphasis is on a good final reproduction quality.

The concept of separating the syntax elements in the order of their importance starts with the MPEG-4 and H.263++. For these standards, video coded data was categorized into header information, motion information, and texture information (also called transformed residual coefficients), listed in the order of their importance. Fig. 2.7 shows the layout of a compressed video data with and without using data partitioning in H.263++. A packet starts with a synchronization marker while, for data partitioning mode, two additional synchronization points are available, i.e., the header marker and the MV marker. For example, combining this concept with that of RVLC and RM, it could be possible to recover most of header and MV information even for the case of high loss within the transform coefficients partition.

In H.264/AVC with DP mode enabled each slice can be segmented into header and motion information, intra information, and inter texture information spanning the syntax elements to individual data units. Typically, the importance of the individual segments of the partition appears in the order of the list. In contrast to MPEG-4, H.264/AVC distinguishes between *inter-* and *intra-texture* information because of the more important role of the latter in error mitigation. The partitions of different importance can be protected with for example Unequal Error Protection (UEP) techniques. The more important data need to be highly protected and vice-versa. It is worth noticing that due to this reordering only on syntax level, coding efficiency is not disturbed, but the loss of individual segments still results in error propagation with similar but typically less severe effects as those shown in Fig. 2.3. Additional detailed investigations of synergies of data partitioning and UEP can be found in [27–29].

2.3.4.5 Redundant Slices

An H.264/AVC encoder may transmit a redundant version of slice sacrificing compression efficiency. Duplicated redundant slice at the receiver can be simply discarded by the decoder during the decoding process. However, when the original slice is lost, this redundant data can be used to reconstruct the lost regions. For example, in a system with high data loss probability, an H.264/AVC encoder can exploit this feature to send redundant information about a ROI. Hence, the decoder will be able of displaying the lost ROI. It is worthwhile noticing that this will still result in an encoder-decoder mismatch of reference pictures, since the encoder being unaware of the loss uses the original slice as a reference, but this effect will be less severe compared to the case when this tool is not used.

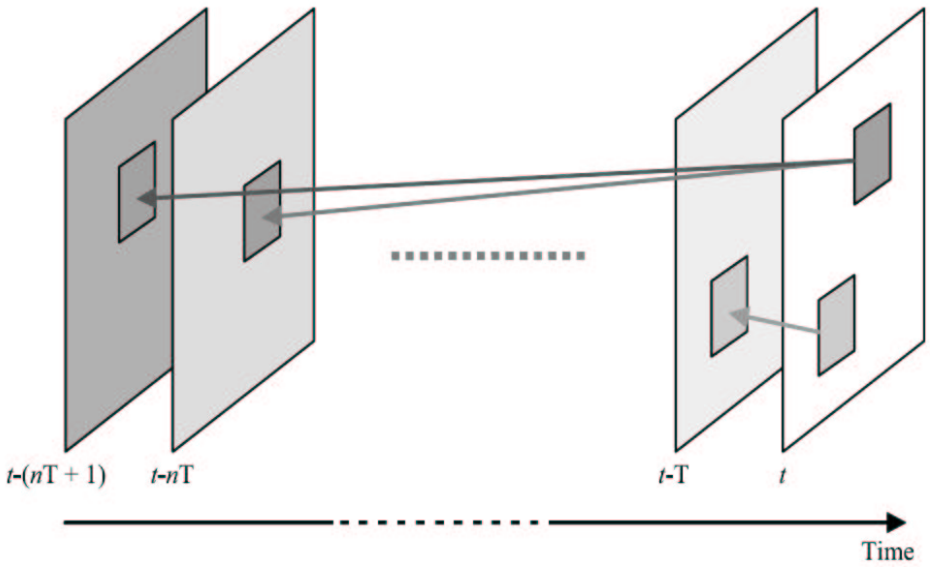

Figure 2.8: Example of an H.264/AVC inter-predicted frame at a given time t, with different MBs referencing different frames. The frame interval in this sketch is T.

2.3.4.6 Flexible Reference Frame Concept

Standards such as H.263 version 1 and MPEG-2 only allow to use a single reference frame for predicting a P frame and at most two frames for predicting a B frame. However, most input sequence manifest a significant dependencies between near pictures in the same Group Of Pictures (GOP). Hence, using more frames than just the recent frame has the advantage of both increasing compression efficiency and improving error resilience. This concept will be beneficial mostly for transmission over error-prone channels. In prior codecs, if the encoder only uses one reference picture and this picture is lost at the decoder side the only available option, to limit error propagation was request intra coded information. In fact, in this case, the encoder and the decoder are not synchronized and the error propagation happen while an I frame is received. However, intra coded data has significantly large size compared to temporally predicted data, which results in further delays and losses on the network. H.263+ and MPEG-4 have proposed tools like the Reference Picture Selection (RPS) that allows encoder selection of a reference picture on a slice or a GOP bases. This feature has posed several computational complexity in the encoder that need to enhance significantly the

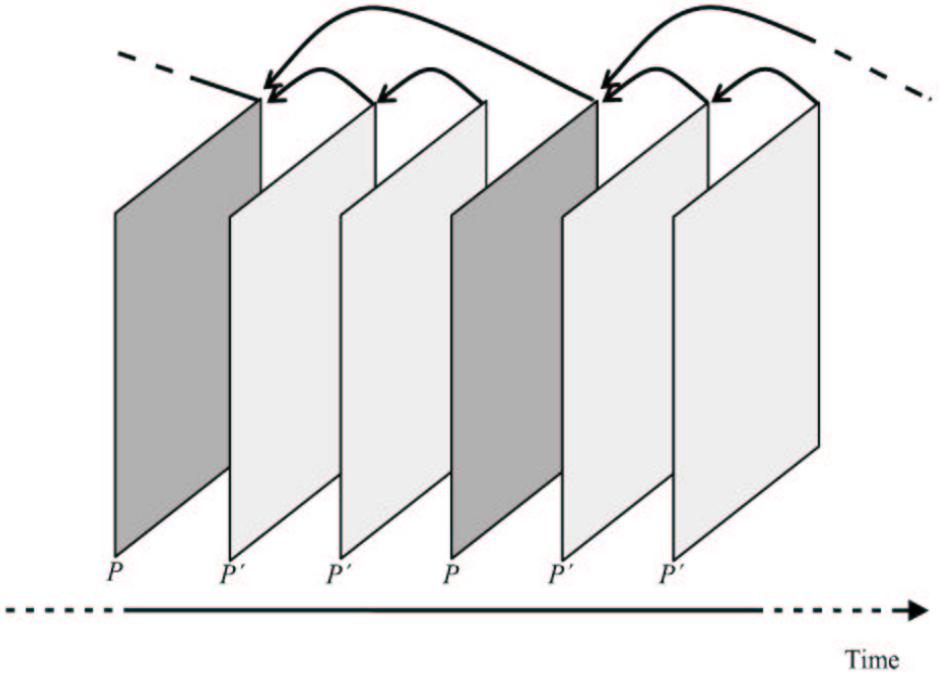

Figure 2.9: H.264/AVC inter-prediction with sub-sequences. Arcs show the reference frame used for prediction.

encoding time of P slice because of the high complexity in searching the motion vectors that minimize the MB Mean Square Error (MSE). On the contrary, at the decoder side temporal prediction is still possible from other correctly received frames. This issue results in an improved error resilience by avoiding using corrupted picture areas as reference. In H.264/AVC this concept has been generalized to allow reference frames to be selected in a flexible way on MB basis (Fig. 2.8). It is also possible using two weighted reference signals for MB inter prediction. The encoder leaves encoded frames in a short-term and long-term memory buffers for future use. The frames stored in the buffer can be used for compression efficiency, for bit-rate control and even for error resilience. It is worth noticing that flexible reference frames can also be used to enable *sub-sequences* in the compressed stream to effectively enable temporal scalability. The basic idea is to use a subsequence of "anchor frames" at lower frame rate than the overall sequence frame rate, shown as P frames in Fig. 2.9. Other frames are inserted in between these frames to achieve the overall target frame rate, shown as P' frames in Fig. 2.9. Here, as an example, every third frame is a P frame. These P' frames can use the low frame rate P frames as reference. This is shown by the chain of prediction

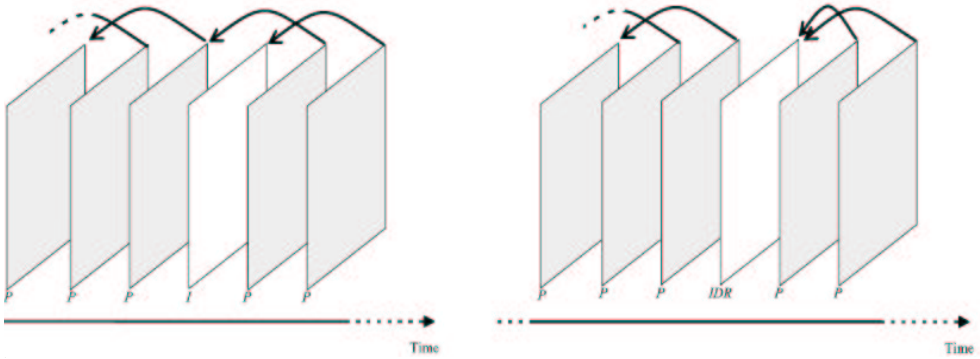

Figure 2.10: Inter prediction with open GOP intra "I" (left) and IDR (right). Temporal prediction (shown by arcs) is not allowed from the frames coded before an IDR frame.

arcs in the same figure. If such a P' frame is lost, the error propagates only till the next P is received. Hence, P frames are more important and they should be protected more than P' frames. Prioritization techniques at lower layers can apply this fact. This concept is similar to using B frames in prior standards, except that a one-directional prediction chain avoids any buffering overhead as with the bi-directionally predicted B pictures.

2.3.4.7 Intra Information Coding

Even though temporal redundancy might exist in a frame, it is still necessary to have the capability of switching off temporal prediction in hybrid video coding. This feature enables random access and also provides an efficient error robust mechanism. Any video coding standard allows encoding image regions in intra mode, i.e., without reference to a previously coded reference frame. In a straightforward way completely intra coded frames might be inserted. These frames will be referred to as *intra frames*. In H.264/AVC, flexible reference frame concept allow the usage of several reference frames. Hence in H.264/AVC, intra frames are further distinguished as Instantaneous Decoder Refresh (IDR) frames and *open GOP* intra frames whereby the latter do not provide the random access property as possibly frames "before" the intra frame are used as reference for "later" predictively coded frames (Fig. 2.10).

In addition, intra information can be introduced for parts of a predictively coded image. Again, most video coding standards allow encoding of single MBs for regions that cannot be predicted efficiently or due to any other case the encoder

decides for non predictive mode. H.264/AVC intra coded MBs gain significant compression by using spatial prediction from neighboring blocks. To limit error propagation, in H.264/AVC this intra mode can be modified such that intra prediction from inter-coded MBs is disallowed. In addition, encoders can also guarantee that MB intra updates result in Gradual Decoding Refresh (GDR), i.e., the entirely correct output pictures after a certain period of time.

2.3.4.8 Switching Pictures

H.264/AVC includes a feature that allows to apply predictive coding even in case of different reference signals. This unique feature is enabled by introducing Switching Predictive (SP) pictures for which the MCP process is performed in the transform domain rather than in the spatial domain and the reference frame is quantized usually with a liner quantizer than that used for the original frame before it is forwarded to the reference frame buffer. These so called Primary SP (PSP) frames, which are introduced to the encoded bit stream, are generally slightly less efficient than regular P-frames but significantly more efficient than regular I-frames. The major benefit results from the fact that this quantized reference signal can be generated without mismatch using any other prediction signal. In case of this prediction signal is generated by predictive coding, the frames are referred to as Secondary SP (SSP) frames, which are usually significantly less efficient than P-frames, as an exact reconstruction is necessary. To generate this reference signal without any predictive signal, the so called Switching Intra (SI) frames can be used. SI pictures are only slightly less inefficient than common intra-coded pictures and can also be used for adaptive error resilience purposes. Further details on this unique feature within H.264/AVC are included in [30].

2.4 Re-synchronization and Error Concealment

2.4.1 Formalization of H.264 Packetized Video

Using the slices and slice groups as detailed in the above sections, video coding standards and in particular H.264/AVC provides a flexible and efficient syntax to map the N_{MB} MBs of each frame s_t of the image sequence to individual data units. The encoding of s_t results in one or more data units P_i with sequence number i. The video transmission system considered is shown in Fig. 2.11 assumes that each data unit P_i is transmitted over a channel that either delivers the data unit P_i correctly, indicated by $C_i = 1$, or loses the data unit, i.e., $C_i = 0$. A data unit is also assumed to be lost if it is received after its Decoding Time Stamp (DTS) expired. More sophisticated concepts also regards multiple decoding deadlines

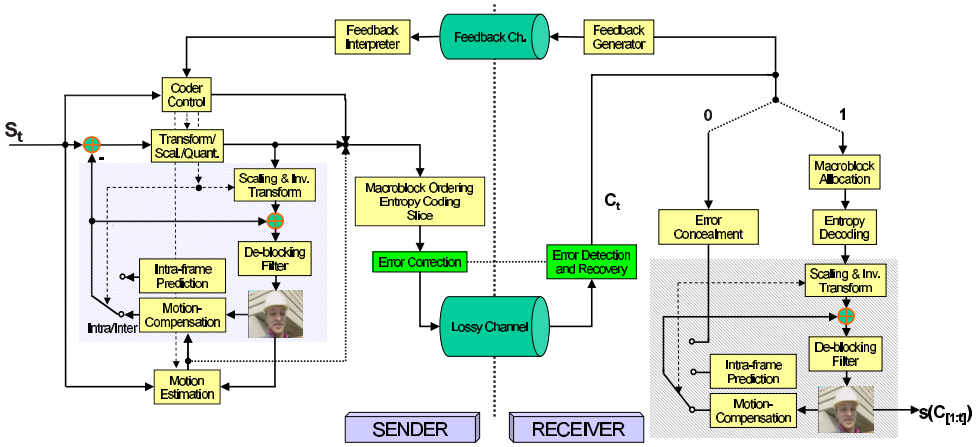

Figure 2.11: Lossy video transmission system.

named Accelerated Retroactive Decoding (ARD) [31] in which late data units are processed by the decoder to update at least the reference buffer, resulting in a reduction of long-term error propagation.

At the receiver, due to the coding restriction of slices the decoder is able to reconstruct the information of each data unit and its encapsulated slice received correctly. The decoded MBs are then distributed according to the mapping M in the frame. For all MBs positions, for which no data has been received, appropriate error concealment has to be invoked before the frame is forwarded both to the reference buffer and the display buffer. The decoded source $\hat{s}_t$ obviously depends on the channel behavior for all the data units P_i corresponding the current frame s_t, but due to the predictive coding and error propagation it also depends on the channel behavior of all previous data units, C_t , $C_{[1:i_t]}$. This dependency is expressed as $\hat{s}_t(C_t)$.

Due to the bidirectional nature of conversational applications, a low-delay, low bit rate, error-free feedback channel from the receiver to the transmitter, as indicated in Fig. 2.11 can be assumed at least for some applications. This feedback link allows sending back some channel messages. These messages make the transmitter aware of the channel conditions so that it may react to these conditions.

2.4.2 Video Packetization Modes in H.264

At the encoder the application of slice structured coding and FMO allows limiting the amount of lost data in case of transmission errors. Especially using FMO, the mapping of MBs to data units basically provides arbitrary flexibility. However,

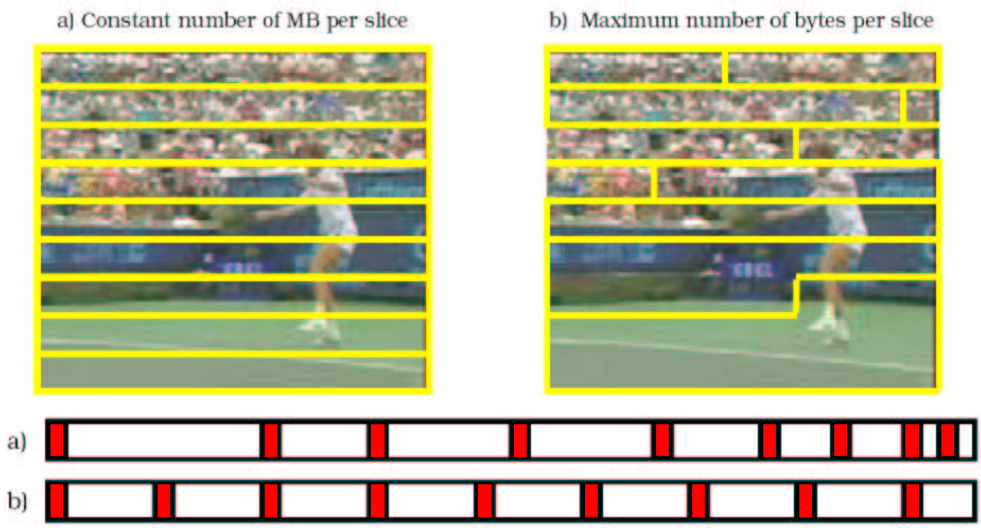

Figure 2.12: Different packetization modes: a) Constant number of MBs per slice with variable number of bytes per slices; b) Maximum number of bytes per slice with variable number of MBs per slice.

there is a few typical mapping modes, which are discussed in the following.

Without the use of FMO, the encoder can choose between two slice coding options, one with a constant number of MBs, $N_{MB/DU}$, within one slice resulting in an arbitrary size, and one with the slice size limited to some maximum number of bytes S_{max}, resulting in an arbitrary number of MBs per slice. Whereas, with the first mode the slice types are similar to that present in H.263 and MPEG-2, the last mode is especially useful to introduce some QoS as commonly the slice size and the resulting packet size determines the data unit loss rate, for example, in wireless systems. Examples of the two different packetization modes and the resulting locations of the slice boundaries in the bit stream are shown in Fig. 2.12. With the use of FMO, the flexibility of the packetization modes is significantly enhanced as shown in the examples in Fig. 2.13. Features such as slice interleaving, dispersed MB allocation using checkerboard-like patterns, enable grouping one or several slice groups. Slice interleaving and dispersed MB allocation are especially powerful in conjunction with appropriate error concealment, i.e., when the samples of a missing slice are surrounded by many samples of correctly decoded slices. Although, this is discussed in the following section. For dispersed MB allocation typically and most efficiently checkerboard patterns are used, if no specific area of the video is treated with higher priority.

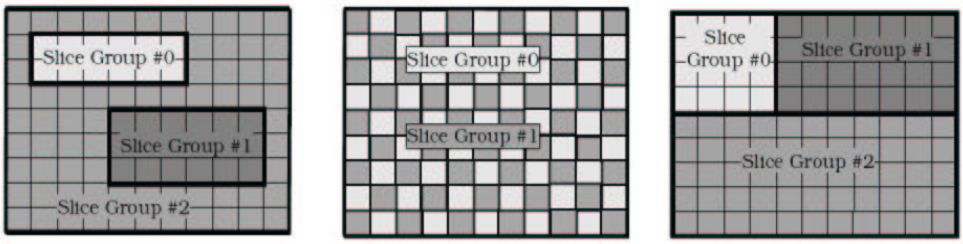

Figure 2.13: Specific MB allocation maps: foreground slice groups with one left-over background slice group, checkerboard pattern with two slice groups, and sub-pictures within a picture.

Video data units may also be packetized on a lower transport layer, e.g., within Real Time Protocol (RTP) [32], by the use of aggregation of packets, by which several data units are collected and fitted into a single transport packet, or by using fragmentation units, i.e., a single data unit is distributed over several transport packets.

2.4.3 Error Concealment

With the detection of a lost data unit at the receiver, the decoder conceals the lost image areas. Error concealment is a *non normative* feature in any actual video decoder, and a large number of techniques have been proposed in literature spanning in a wide range of performance and complexity. The basic idea is that the decoder should generate a representation for the lost area that matches perceptually as close as possible to the lost information without knowing the lost information. All these techniques are based on *best effort*, with no guarantee of an optimal solution. Since the concealed version of the decoded image will still differ from its corresponding version at the encoder side, error propagation will still occur in the following decoded images until the reference frames receives re-synchronize encoder and decoder.

From this point of view most popular techniques are based on a few common assumptions as follow:

- Continuity of image content in spatial domain: natural scene content typically consists of smooth texture;
- Temporal continuity: smooth object motion is more common compared to abrupt scene changes and those collocated regions in image tend to have similar motion displacement.

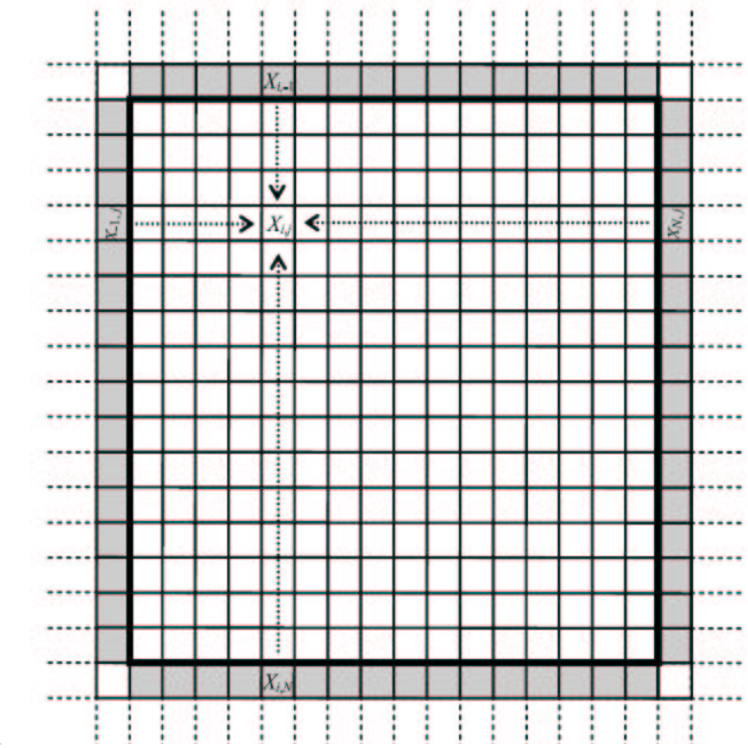

Figure 2.14: Pixels used for spatial error concealment (shaded pixels) of a lost MB, $M = N = 16$.

Such techniques manage the received information of surrounding area in the spatial and temporal domain to conceal the lost regions. In the following the focus is on the techniques that conceal each lost MB individually and do not modify the correctly received data.

To simplify the discussion in this section, "data loss" refers to the case that *all* the related information of one or several MBs is lost, e.g., MB mode, transformed residual coefficients and MVs (for the case of inter-coded MBs). This assumption is quite practical as typically a corrupted packet will be detected and discarded before the video decoder initiates the decoding process (i.e. at other ISO levels).

There is an exhaustive amount of literature proposing different error concealment techniques. However, only a few schemes are commonly used in practical applications due to their limited computational complexity. Emphasis will be given on error concealment techniques with some practical relevance providing reference to other important error concealment methods available in literature. In general, error concealment needs to trade off between performance and complexity.

2.4.3.1 Spatial Error Concealment

Spatial error concealment techniques are based on the assumption of continuity of natural scene content in space domain (i.e. in the same frame). This method generally uses pixel values of surrounding available MBs in the same frame as shown in Fig. 2.14. Availability refers to MBs that either have been received correctly or have already been concealed. In the example it is considered the case of loss of a 16 x 16 MB. The most common way of determining the pixel values in a lost MB is by using a weighted sum of the closest boundary pixels of available MBs, with the weights being inversely related to the distance between the pixel to be concealed and the boundary pixel. For example at a pixel position i, j in Fig. 2.14, an estimate $\hat{X}_{i,j}$ of the lost pixel $X_{i,j}$ is:

$$\hat{X}_{i,j} = \alpha\{\beta X_{i,-1} + (1-\beta)X_{i,16}\} + (1-\alpha)\{\gamma X_{-1,j} + (1-\gamma)X_{16,j}\}, \quad (2.1)$$

In this equation, α, β and γ are weighing factors that establish the relative impact of pixel values of vertical versus horizontal, upper versus lower and left versus right neighbors, respectively. The top-left pixel of the lost MB is considered as origin. As discussed previously, the weighing factors are set according to the inverse of the distances from the pixel being estimated. This technique as proposed in [33] is widely used in practice because of its simplicity and very low complexity. Since this technique is based on the assumption of continuity in spatial domain, discontinuity is avoided in concealed regions of the image. Obviously, this techniques will result in erroneous reconstruction of lost region, since natural scene content is not perfectly continuous and lost details will not be recovered. Typically, spatial error concealment technique is never used on its own in applications rather, it is combined with other techniques as discussed in the following sections. It is worthwhile to note that since this technique heavily relies on the availability of horizontal and vertical neighbor pixels, decoders applying this technique can benefit from the application of FMO, e.g., by the use of a checkerboard-like pattern.

More sophisticated methods with higher complexity have been proposed in literature. These methods target to recover some of the lost texture. The most important techniques are listed in the following.

- In [34] a spatial error concealment technique is based on an a priori assumption of continuity of *geometric structure* across the lost region. The available neighboring pixels are used to extract the local geometric structure, which is characterized by a bimodal distribution. Missing pixels are reconstructed by the extracted geometric information.

- Projection onto convex sets in the frequency domain is proposed in [35]. In this method each constraint about the unknown area is formulated as a

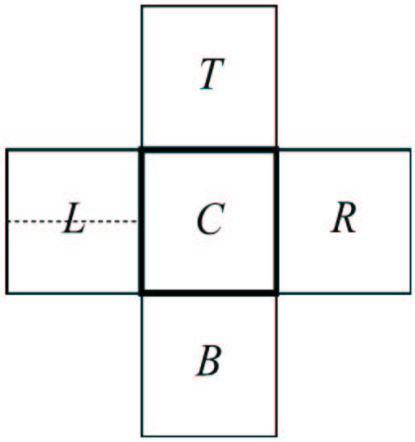

Figure 2.15: Neighboring available MBs (T , R, B, and L) used for temporal error concealment of a lost MB C. MB L is encoded in 16 x 8 inter mode, and the average of its two MVs is used as a candidate.

convex set, and a possible solution is iteratively projected onto each convex set to obtain a refined solution.

2.4.3.2 Temporal Error Concealment

Temporal error concealment relies on the continuity of a video sequence in the time domain. This technique uses the temporally neighboring areas to conceal the lost regions.

In the simplest form of this technique, known as the Previous Frame Concealment (PFC), the spatially corresponding data of the lost MB in the previous frame is copied to the current frame. If the scene has little motion, PFC performance quite well. However, as soon as the region to be concealed is displaced from the corresponding region in the previous frame, this technique will result in significant artifacts in the displayed image. However, due to its simplicity this technique is widely used, especially in decoders with limited processing power.

A refinement of PFC attempts to reconstruct the image by making an estimate of the lost MV. For example, with the assumption of a uniform motion field in the collocated image areas, motion vectors of the neighboring blocks are good candidates to be used as displacement vectors to conceal the lost region. Good candidate MVs for this technique are the MVs of available horizontal and vertical inter-coded neighbor MBs. If a neighboring MB is encoded in an inter mode other than the inter 16 x 16 mode, one approach is to use the average of the MVs of all the blocks on the boundary of the lost MB. In general, more than one option for

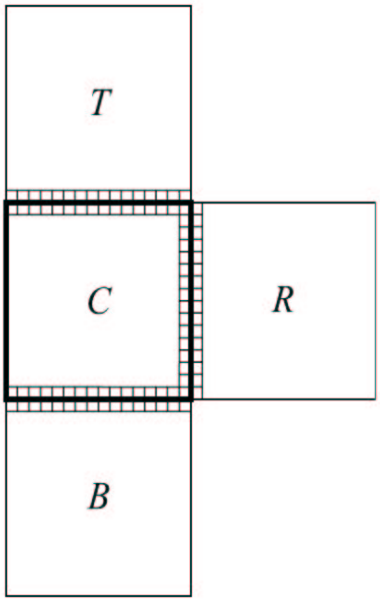

Figure 2.16: Boundary pixels of MB C used for the boundary matching criteria.

the application of displacement vectors exist, e.g., using the horizontal neighbor and the vertical neighbor. A boundary-matching-based technique can be applied, for instance, to select one of the many candidates (Fig. 2.16). In this case, from the set of all candidate MVs $\mathbf{S}$, the MV $\hat{\upsilon}$ for temporal error concealment is chosen according to:

$$\begin{aligned} \varepsilon_T(\upsilon_i) &= \textstyle\sum_{m=1}^{15}(X_{x+m,y}(\upsilon_i) - X_{x+m,y-1}(\upsilon_i))^2 \\ \varepsilon_R(\upsilon_i) &= \textstyle\sum_{m=1}^{15}(X_{x+15,y+n}(\upsilon_i) - X_{x+16,y+n}(\upsilon_i))^2 \\ \varepsilon_B(\upsilon_i) &= \textstyle\sum_{m=1}^{15}(X_{x+m,y+15}(\upsilon_i) - X_{x+m,y+16}(\upsilon_i))^2 \\ \hat{\varepsilon} &= \arg\min_{\upsilon_i \epsilon \mathbf{S}}(\varepsilon_T(\upsilon_i) + \varepsilon_R(\upsilon_i) + \varepsilon_B(\upsilon_i)) \end{aligned} \quad , \tag{2.2}$$

Here, for each motion vector $\upsilon_i \epsilon \mathbf{S}$, errors $\varepsilon_T(\upsilon_i)$, $\varepsilon_R(\upsilon_i)$, and $\varepsilon_B(\upsilon_i)$ are calculated for top, right, and bottom edges respectively. The first term of error functions is the pixel recovered from the reference frame using the selected motion vector υ_i , while the second element is an available boundary pixel of a neighboring MB. The upper left pixel of the lost MB has a pixel offset x, y. Finally, the vector that results in minimum overall error is selected, since this vector gives a block that possibly fits best in the lost area. Obviously it is possible that none of the candi-

date vectors are suitable and in such a case temporal error concealment results in a fairly noticeable discontinuity artifacts in the concealed regions.

Several variants and refinements of temporal error concealment technique were proposed, usually presenting some better performance at the expense of sometimes significantly higher complexity. The literature proposed numerous contributions regarding the temporal EC, the following is a list concerining these combinations:

- In [36], *overlapped block motion compensation* is proposed. In this case an average of three 16 x 16 pixel regions is used to conceal the missing MB. One of these regions is the 16 x 16 pixel data used to conceal the lost MB by the process described above, the second and third regions are retrieved from the previous frame by using the motion vectors of horizontal and vertical neighbor MBs respectively. These three regions are averaged to get the final 16 x 16 data used for concealment. Averaging in this way can reduce artifacts in the concealed regions.

- In [37], it is proposed to use median motion vector of the neighboring blocks for temporal concealment. However, the benefits of this techniques have been relativized in, e.g., [38].

- In [38], Sum of Absolute Differences (SAD) is used instead of Sum of Squared Differences (SSD) for boundary-matching technique. This results in reduced computational complexity.

- A more simple variant is used in practice [39] where the authors propose only to apply the motion vector of top MB, if available, otherwise it is used zero MV.

- In [40], a *multi-hypothesis* error concealment is proposed. This technique take advantage of the multiple reference frames available in an H.264/AVC decoder for temporal error concealment. The erroneous block is compensated by a weighted average of correctly received blocks in more than one previous frame. The weighting coefficient used for different blocks can be determined adaptively.

- In [41], the idea presented in [40] is extended. In the paper, temporal error concealment is used exclusively. However, two variants of temporal error concealment are available, the low-complexity concealment technique and the multi-hypothesis temporal error concealment. The decision on which technique is used is based on the temporal activity (SAD) in the neighboring regions of the damaged block. For low scene activity, the low-complexity

technique is used, while multi-hypothesis temporal error concealment is used for higher scene activity.

Also, the adaptive combination of spatial concealment with temporal error concealment is of some practical interest and will therefore be discussed in detail in the following subsection.

2.4.3.3 Hybrid concealment

Neither the application of spatial concealment nor temporal concealment alone can provide satisfactory performance: if only spatial concealment is used, concealed regions usually are significantly blurred. Similarly, if only temporal error concealment is applied significant discontinuities in the concealed regions can occur, especially if the surrounding area can not provide any or not sufficiently good motion vectors. Hence, for better results, hybrid temporal-spatial technique might be applied. In this technique, MB mode information of reliable and concealed neighbors can be used to decide whether spatial error concealment or temporal error concealment is more suitable. For intra-coded images only spatial concealment is used. For inter-coded images, temporal error concealment is used only if, for example, in the surrounding area more than half of the available neighbor MBs (shown in Fig. 2.15) are inter-coded, otherwise it's used the spatial error concealment. This ensures that a sufficient number of candidate MVs are available to estimate the lost motion information. This error concealment technique is referred as Adaptive temporal and spatial Error Concealment (AEC). Other techniques have been proposed to decide between temporal and spatial concealment mode as detailed in the following:

- A simple approach in [38] proposes use of spatial concealment for intra-coded images and temporal error concealment for all inter-coded images invariably.
- In [42], it is suggested that if the residual data in a correctly received neighboring inter-predicted MB is smaller than a threshold, temporal error concealment should be used.
- In [43], it is suggested to use the last N frames to construct a differential equation used to predict the lost pixels values.

2.4.3.4 Miscellaneous Techniques

In addition to the signal-domain MB-based approaches, other techniques have been proposed in the literature, for example:

- Model based or object concealment techniques, as proposed in [44–45], do not take up simple a priori assumptions of continuity as given earlier. These techniques are based on the specific texture properties of video objects, and as such are a suitable option for multiobject video codec, that is, MPEG-4. An object-specific context-based model is built and this model governs the assumptions used for concealment of that object.

- Frequency-domain concealment techniques [46–47] work by reconstructing the lost coefficients by using the available coefficients of the neighboring MBs as well as coefficients of the same MB not affected by the loss. These initial proposals are specifically for DCT transform block of 8 x 8 coefficients. For example, in [46], based on the assumption of continuity of the transform coefficients, lost coefficients are reconstructed as a linear combination of the available coefficients. However, noticeable artifacts are introduced by this technique. As a more realistic consideration, in [47] the constraint of continuity holds only at the boundaries of the lost MB in spatial domain.

- In an extension to the spatial and temporal continuity assumptions, it is proposed in [48] that the frames of video content are modelled as a Markov Random Field (MRF). The lost data is suggested to be recovered basing on this model. In DelpFastEC [49] the authors proposed a less complex but suboptimal alternative to implement this model for error concealment. For example, for temporal error concealment, only the boundary pixels of the lost MB are predicted based on a MAP estimate, instead of predicting the entire MB. These predicted pixels are used to estimate the best predicted motion vector to be used for temporal error concealment. In [50], the MAP estimate is used to refine an initial estimate obtained from temporal error concealment.

2.4.3.5 Visual Error Concealment Effects

A few selected results from the presented important error concealment techniques are presented in Fig. 2.17. From left to right, it's shown a sample concealed frame when using PFC, spatial, temporal, and AEC. PFC simply replaces the missing information by the information at the same location in the temporally preceding frame. Hence, it shows artifacts in the global motion part of the background as well as in the foreground.

Spatial error concealment based on weighted pixel averaging smoothes the erroneously decoded image and removes strange block artifacts, but also many details. Temporal error concealment relying on motion vector reconstruction with boundary-matching based techniques keeps details, but results in strange artifacts

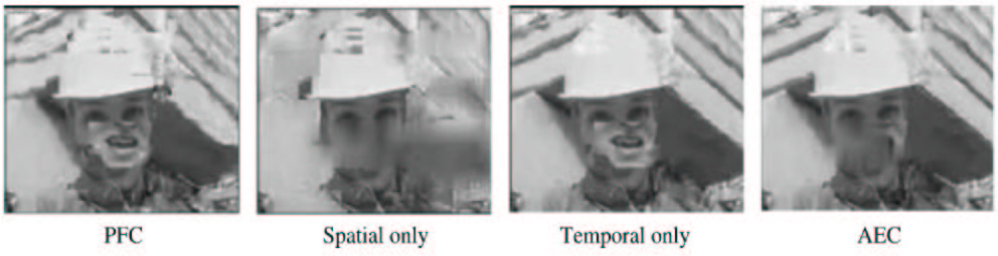

Figure 2.17: Performance of different error concealment strategies: PFC, spatial concealment only, temporal error concealment only and AEC.

in uncovered areas. Finally, AEC (the combination of temporal and spatial error concealment) keeps many details but also avoids strange block artifacts and is therefore very appropriate with feasible complexity. It is worth noticing that AEC is the most general form of EC which reduces to PFC in the case that all MBs of a picture are transmitted in a single packet. For a further detailed and investigations on the recent advances in error concealment techniques, the reader is referred to [51] and the references therein.

2.4.3.6 Selected Performance Results for Wireless Test Conditions

To get an insight in error-resilient video coding for cellular and in particular for 3G mobile communication scenarios, some few selected results are proposed. The simulated scenario is a packet-switched conversational application and is specified in detail by the 3GPP in [52]. This application is characterized by its tight low-delay and low-complexity requirements, since the processing has to be done in real time on hand-held devices. As a result, the maximum allowed buffering at the encoder is limited to 250 ms and only the first frame is encoded as intra, to limit any delays caused by buffering overheads. A simple random intra MB refresh technique is used, with only 5% MBs of every frame coded in intra mode. The most recent frame is used for motion compensation to limit the complexity. With these limitations, the impact of slice size on error resilience of the application is observed. In particular two main channel configurations are compared: one with moderate Radio Link Control (RLC) Packet Data Unit (PDU) loss rate of 0.5% and the other with a higher loss rate of 1.5%. The physical link in this test supports transmission of 128 Kbps, with a radio frame size of 320 bytes. In this configuration one frame splits in several radio data units. The Quarter Common Intermediate Format (QCIF) test sequence *Foreman* at 15 frames per second is used. The encoder is configured to match the maximum throughput of channel while taking into account packetization overheads. The criterion used here as a

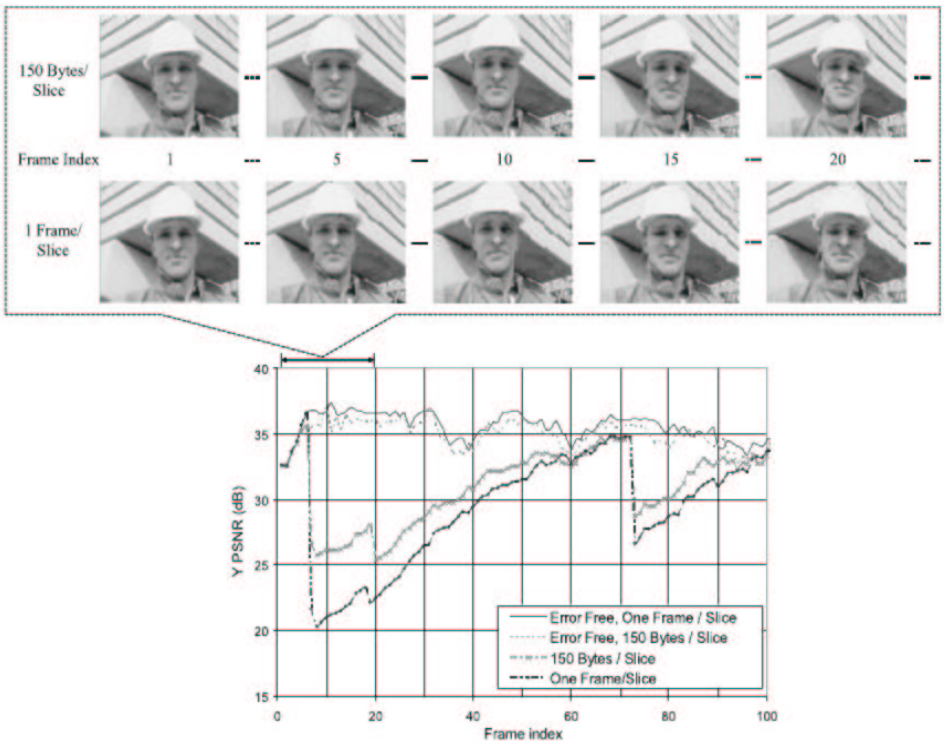

Figure 2.18: (Bottom) Plot of Y PSNR with two different slice modes. Results for an error-free case are given as a reference. (Top) A few selected frames for the two slice modes for comparison.

metric of perceived video quality is PSNR of the luma (Y) signal component.

Fig. 2.18 compares the Y PSNR of the decoded video at a loss rate of 1.5% for two cases: transmitting an entire frame in a slice versus a fixed slice size of 150 bytes. At the given bit rate, a compressed frame has an average size of roughly 1000 bytes. The error-free performance for both cases is also plotted as a reference. Obviously, using a smaller slice size of 150 bytes results in typically lower PSNR in an error-free case because of two reasons: increased packetization overhead and prediction limitations on slice boundaries. However, this configuration outperforms in the case of lossy channel throughout the observed period. A few selected frames are also presented for comparison. The effects of losses already start to appear in the fifth frame. While transmitting one frame per slice results in loss of an entire frame for a lost data unit, the loss affects only a small area of the image for fixed slice size. The spatiotemporal error propagation is much smaller in this case.

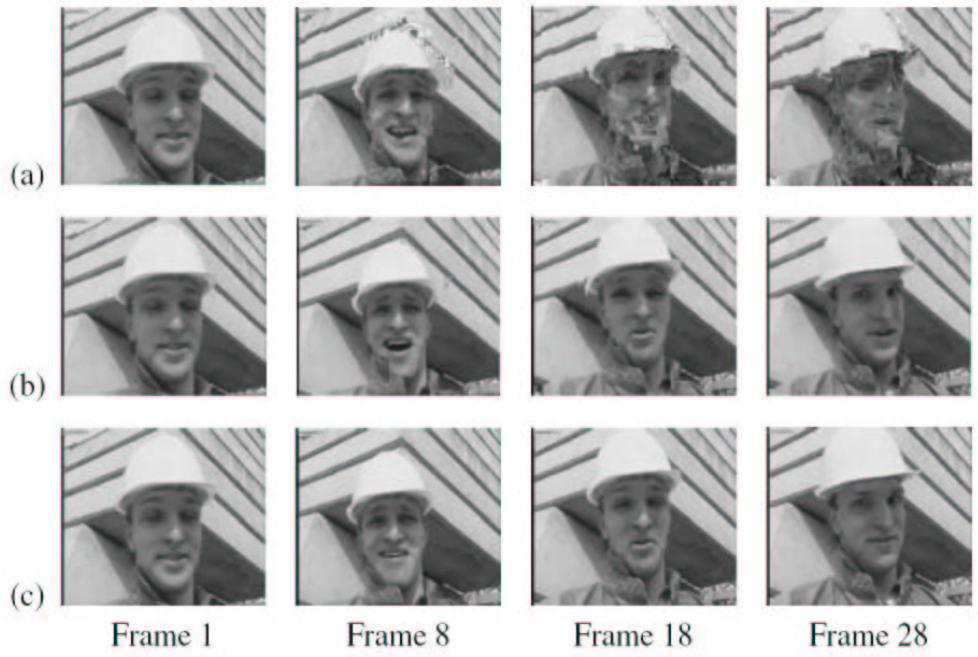

Figure 2.19: Selected frames of decoded video sequence *Foreman* for a packet lossy channel with same bitrate and error constraints (a) no error robustness, (b) adaptive intra updates and (c) interactive error control.

2.5 Error Mitigation Techniques

2.5.1 Motivations

As already discussed, error propagation is the major problem when transmitting MCP-coded video over lossy channels. Therefore, if the encoder knows that the channel has damaged certain data units or knows that the decoder has experienced the loss of certain data units, it should change its encoding strategy, sacrificing the compression efficiency. To illustrate this behavior, selected frames for different encoding strategies when transmitting over channels with the same bit rate and error rate are shown in Fig. 2.19. The first line, indicated as (a), shows the case where no specific error resilience tools are applied. The error propagation affect all the last frames. For the sequence in the second line, indicated as (b), the same bit rate and error statistics are applied, but the encoder chooses to select intra-coded MBs in a suitable way. It can be observed that the error propagation is less evident but some residual artifacts are still visible. In addition, the error-free video has lower quality as its compression efficiency is reduced due to the increased amount of intra coding that requires additional bit rate but the constraints are the

same. Error propagation can be completely avoided only by using interactive error control, as shown in the third row, indicated with (c) in Fig. 2.19. However, also in this case, compression efficiency is sacrificed, especially if necessary feedback of the decoder state is delayed. Additional details on the appropriate selection of MB modes in error-prone environments, especially taking into account the tradeoff between quantization distortion and reduced error propagation, are discussed in the following.

2.5.2 Operational Encoder Control

The tools for increased error resilience in *hybrid* video coding, in particular those to limit error propagation, do not significantly differ from the ones used for compression efficiency. Features like multi frame prediction or intra coding of individual MBs are not primarily error resilience tools, they are mainly used to increase coding efficiency in error-free environments. The encoder implementation is responsible for appropriate selection of one of the many different encoding parameters, the so-called *operational coder control*. Therefore, the encoder must take into account constraints imposed by the application in terms of bit rate, encoding and transmission delay, complexity, and buffer size etc. When a standard decoder is used, such as H.264/AVC compliant decoder, the encoding parameters should be selected by the encoder such that good rate-distortion performance is achieved. Since the encoder is limited by the syntax of the standard, this problem is called *syntax-constrained rate-distortion optimization* [53].

In case of H.264/AVC, for example, the encoder must appropriately select parameters such as motion vectors, MB modes, quantization parameters, reference frames, or spatial and temporal resolution, as shown in Fig. 2.20. This also means that bad decisions at the encoder can lead to poor results in coding efficiency or error resilience or both. For compression efficiency, operational encoder control based on Lagrangian multiplier techniques have been proposed. The distortion d_{b,m_b} usually (at least in the H.264/AVC test model) reflects the SSD between the original MB s_b and reconstructed version of the MB $\widetilde{s}_{b,m}$ if coded with option m, i.e:

$$d_{b,m_b} = \sum_i |s_{b,i} - \widetilde{s}_{b,m,i}|^2 \tag{2.3}$$

and the rate $r_{b,m}$ is defined by the number of bits necessary to code MB b with option m. Finally, the coding mode is selected for MB b as:

$$\forall_b \qquad m_b^* = \arg\min_{m \epsilon \Theta} \left(d_{b,m_b} + \lambda_\Theta r_{b,m}\right) \tag{2.4}$$

whereby Θ defines the set of selectable options, e.g., MB modes. For the Lagrangian parameter λ_Θ it is proposed in [54] and [55] that if the SSD is applied

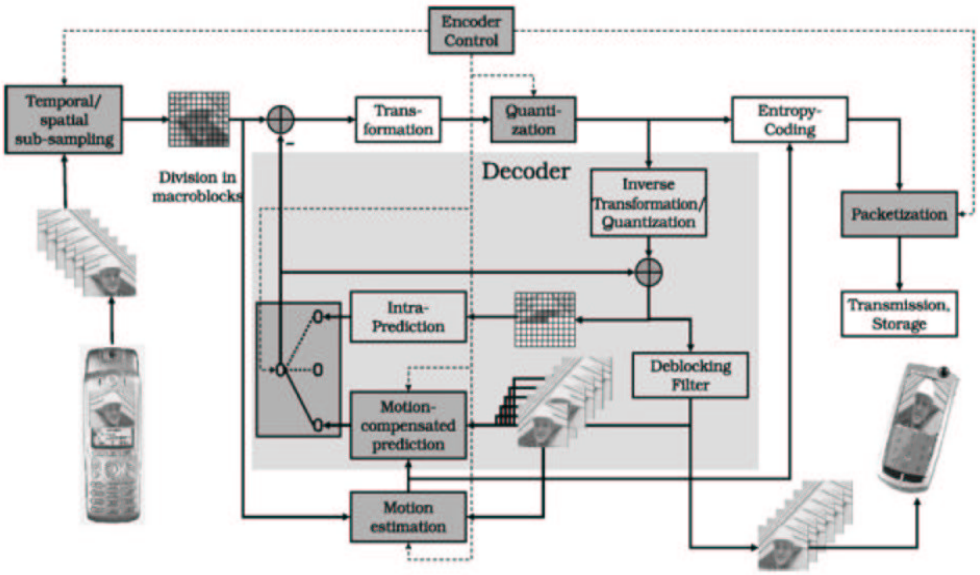

Figure 2.20: H.264/AVC video encoder with selectable encoding parameters highlighted.

as distortion measure then λ_Θ should be directly proportional to the square of the step size Δ of a uniform quantizer applied. The procedure in (2.4) can be applied to select motion vectors, reference frames, and MB modes. However, it is obviously contradictory if the same decision procedure is applied to obtain good selections for compression efficiency and error resilience.

2.5.3 Intra Updates

In the presence of errors the introduction of more frequent non predictively coded image parts is of major importance. In previous works that concern this subject, e.g., [56–58], it has been proposed to introduce intra coded MBs, regularly, randomly or preferably in a certain pseudo random update pattern. In addition, sequence characteristics and bit rate in influence the appropriate percentage of intra updates. Recognizing this it has been proposed in [59–60] to modify the selection of the coding modes according to (2.4) to take into account the influence of the lossy channel. When encoding MB b with a certain coding mode m_b, it is suggested to replace the encoding distortion $d_{b,m}$ by the decoder distortion:

$$\widetilde{d}_{b,m}(C_t) \triangleq ||s_{b,t} - \hat{s}_{b,t}(C_t, m)||^2 \tag{2.5}$$

which obviously depends on the reconstructed pixel values $\hat{s}_{C_t,m}$ and therefore also on the channel behavior C_t and the selected coding mode m. In general, the channel behavior is not deterministic and the channel realization C, observed by the decoder is unknown to the encoder. Thus it is not possible to directly determine the decoder distortion (2.5) at the encoder. However, it is possible to assume that the encoder has at least some knowledge of the statistics of the random channel behavior, denoted as C_t. In RTP [32] environment, the Real Time Control Protocol (RTCP) [61] for example can use a feedback channel to send receiver reports on the experienced loss and delays statistics, which allows the encoder to incorporate the statistics into the encoding process. Assuming that the statistics on the loss process are perfectly known to the encoder, i.e., $B(C_t) = C_t$, then, the encoder is able to compute the expected distortion:

$$\bar{d}_{b,m} = E_{\hat{C}_t}\{\tilde{d}_{b,m}(\hat{C}_t)\} = E_{\hat{C}_t}\{||s_{b,t} - \hat{s}_{b,t}(\hat{C}_t, m)||^2\} \tag{2.6}$$

A similar procedure can be applied to decisions on reference frames and motion vectors. The selection of motion vectors based on the expected distortion has for example been proposed in [62]. The estimation of the squared expected pixel distortion in packet loss environment has been addressed in several papers available in literature. For example, in [60], [63], and [64], several methods to estimate the distortion introduced due transmission errors and the resulting error propagation have been proposed. In all these proposals the quantization noise and the distortion introduced by the transmission errors are combined linearly. Since the encoder needs to keep track of an estimated pixel distortion, additional complexity and memory is required in the encoder. The most important method, the so called Recursive Optimal per Pixel Estimate (ROPE) algorithm [59], provides an accurate estimation for baseline H.263 and MPEG-4 simple profile algorithms, using simple temporal error concealment, by keeping track of the first and second moment of the decoded pixel value $\widetilde{s}_{C_t}$, namely $E\{\widetilde{s}(C_t)\}$ and $E\{\widetilde{s}^2(C_t)\}$ respectively. A powerful more complex method has been proposed in [65] where the authors proposes a Monte Carlo like method. An estimate of the decoder distortion $\bar{d}_{b,m}$ in (2.6) is obtained as:

$$\bar{d}_{b,m}^{(N_C)} \triangleq \frac{1}{N_C}\sum_{n=1}^{N_C}\tilde{d}_{b,m}(C_{n,t}) = \frac{1}{N_C}\sum_{n=1}^{N_C}||s_{b,t} - \hat{s}_{b,t}(C_{n,t}, m)||^p \tag{2.7}$$

with $C_{n,t}$, $n = 1...N_C$, representing N_C independent realizations of the random channel $\widehat{C}_t$, and estimate of the loss probability at the receiver represented as p. An interpretation of (2.6) leads to a simple solution to estimate the expected pixel distortion $\bar{d}_{b,m}$. For more details we refer to [65]. To obtain an estimate of the loss probability p at the receiver, the feedback channel can be used in practical systems.

2.5.4 Interactive Error Control

The availability of a feedback channel, especially for conversational applications, led to different standardization and research activities in recent years to include this feedback in the video encoding process. Assuming that, in contrast to the previous scenario where only the statistics of the channel process $\hat{C}$ are known to the encoder, in the case of timely feedback it is even possible to assume that a δ-frame delayed version $C_{t-\delta}$ of the loss process experienced at the receiver is known at the encoder. This characteristic can be conveyed from the decoder to the encoder by sending acknowledgment for correctly received data units, negative acknowledgment messages for missing slices, or both types of messages. In less time-critical applications, such as streaming or downloading, the encoder could obviously decide to retransmit lost data units in case it has stored a backup of the data unit at the transmitter. However, in low-delay applications the retransmitted data units, especially in end-to-end connections would in general arrive too late to be useful at the decoder. In case of online encoding, the observed and possibly delayed receiver channel realization, $C_{t-\delta}$, can still be useful to the encoder, although the erroneous frame has already been decoded and concealed at the decoder. The basic goal of these approaches is to reduce, limit, or even completely avoid error propagation by integrating the decoder state information into the encoding process.

The exploitation of the observed channel at the encoder has been introduced in [66] and [67] under the acronym *Error Tracking* for standards such as MPEG-2, H.261 or H.263 version 1, but has been limited by the reduced syntax capabilities of these video standards. When receiving the information that a certain data unit - typically including the coded representation of several or all MBs of a certain frame $s_{t-\delta}$ - has not been received correctly at the decoder, the encoder attempts to track the error to obtain an estimate of the decoded frame $\hat{s}_{t-1}$ serving as reference for the frame to be encoded, s_t. Appropriate actions after having tracked the error are discussed in [66–69]. However, all these concepts have in common that error propagation in frame $\hat{s}_t$ is only removed if frames $\hat{s}_{t-\delta+1}, ..., \hat{s}_{t-1}$ have been received at the decoder without any error.

Assume that at the encoder each generated data unit P_i is assigned a decoder state $C_{enc,i}\epsilon\{ACK, NAK, OAK\}$, whereby $C_{enc,i} = ACK$ reflects that data unit P_i is known to be correctly received at the decoder, $C_{enc,i} = NAK$ reflects that data unit P_i is known to be missing at the decoder, and $C_{enc,i} = OAK$ reflects that for data unit P_i the acknowledgment message is still outstanding and it is not known whether this data unit will be received correctly. With feedback messages conveying the observed channel state at the receiver, that is, $B(C_t) = C_t$, and a back channel that delays the back channel messages by δ frames, it can be assumed in the remainder that for the encoding of s_t , the encoder is aware of the following

information:

$$C_{enc,i} = \begin{cases} \text{ACK} & \text{if} \quad \tau_{PTS,i} \leq \tau_{s,t-\delta} \quad \text{and} \quad C_i = 1 \\ \text{NAK} & \text{if} \quad \tau_{PTS,i} \leq \tau_{s,t-\delta} \quad \text{and} \quad C_i = 0 \\ \text{OAK} & \text{if} \quad \tau_{PTS,i} \geq \tau_{s,t-\delta} \end{cases}, \tag{2.8}$$

where $\tau_{PTS,i}$ is the Presentation Time Stamp (PTS) of P_i and $\tau_{s,t-\delta}$ is the sampling time of $s_{t-\delta}$. This information about the decoder state $C_{enc,i}$ can be integrated in a modified rat-distortion optimized operational encoder control similar to what has been discussed in Subsection 2.5.2. In this case the MB mode m_b^* is selected from a modified set of options,$\hat{\Theta}$, with a modified distortion $\hat{d}_{b,m}$ for each selected option m as:

$$\forall_b \qquad m_b^* = \arg\min_{m \epsilon \hat{\Theta}} (\hat{d}_{b,m} + \lambda_{\hat{\Theta}} r_{b,m}), \tag{2.9}$$

In the following it can be distinguished four different operation modes, which differ only for the set of coding options available to the encoder in the encoding process,$\hat{\Theta}$, as well as the applied distortion metric, $\hat{d}_{b,m}$. The encoderś reaction to delayed positive acknowledgment (ACK) and negative acknowledgment (NAK) messages is shown in Fig. 2.21, assuming that frame d is lost and the feedback delay is $\delta = 2$ frames for three different feedback modes.

2.5.4.1 Feedback Mode 1: Acknowledged Reference Area Only

Fig. 2.21 (a) shows this operation mode: only the decoded representation of data units P_i that have been positively acknowledged at the encoder, that is, $C_{enc,i} = ACK$, are allowed to be referenced in the encoding process. In the context of operational encoder control, this is formalized by applying the encoding distortion in (), that is, $\hat{d}_{b,m} = d_{b,m}$, as well as the set of encoding options that is restricted to acknowledged areas only, that is, $\hat{\Theta} = \Theta_{ACK,t}$. Note that the restricted option set $\Theta_{ACK,t}$ depends on the frame to be encoded and is applied to the motion estimation and reference frame selection process. Obviously, if no reference area is available, the option set is restricted to intra modes only, or if no satisfying match is found in the accessible reference area, intra coding is applied. With this mode in use, an error might still be visible in the presentation of a single frame; however, error propagation and reference frame mismatch are completely avoided. In terms of performance it is possible to debate that for small delays, the gains are significant and for the same average PSNR the bit rate is less than 50% compared to the forward only mode. With increasing delay the gains are reduced, but compared with the highly complex mode decision without feedback, this method is still very attractive.

2.5.4.2 Feedback Mode 2: Synchronized Reference Frames

Feedback mode 2 as shown in Fig. 2.21 (b) differs from mode 1 since not only for positively acknowledged data units but also a concealed version of data units with decoder state $C_{enc,i} = NAK$ is allowed to be referenced. This is formalized by applying the encoding distortion in (), that is, $\hat{d}_{b,m} = d_{b,m}$, but the restricted reference area and the option set in this case also include concealed image parts. The critical aspect when operating in this mode results from the fact that for the reference frames to be synchronized the encoder must apply exactly the same error concealment as the decoder.

The advantage of feedback mode 2 with respect to feedback mode 1 can be seen in two cases: for low bit rates and for delays. This is so because referencing concealed areas is preferred over intra coding by the rate–distortion optimization. For higher bit rates this advantage vanishes as the intra mode is preferred anyways over the selection of "bad" reference areas.

2.5.4.3 Feedback Mode 3: Acknowledged Reference Area Only

Feedback modes 1 and 2 are mainly suitable in cases of higher loss rates. If the loss rates are low or negligible, the performance is significantly degraded by the longer prediction chains due to the feedback delay. Therefore, in feedback mode 3 as shown in Fig. 2.21 (c) it is proposed only to alter the prediction in the encoder in case of the reception of a NAK. This mode obviously performs well in cases of lower error rates. However, for higher error rates error propagation still occurs quite frequently.

In summary from the above analysis as well as subjective observations, it can be concluded that avoiding error propagation is basically the most important issue in error-prone video transmission. If no feedback is available, an increased percentage of intra MBs, selected by channel-adapted optimization schemes, performs best. Whenever feedback is available, it is suggested that interactive error control be applied. For short delays or low error rates, it is suggested to modify the prediction only in case of the reception of NACK messages, in all other cases, it is suggested to reference only those areas for which the encoder is sure that the decoder has exactly the same reference area.

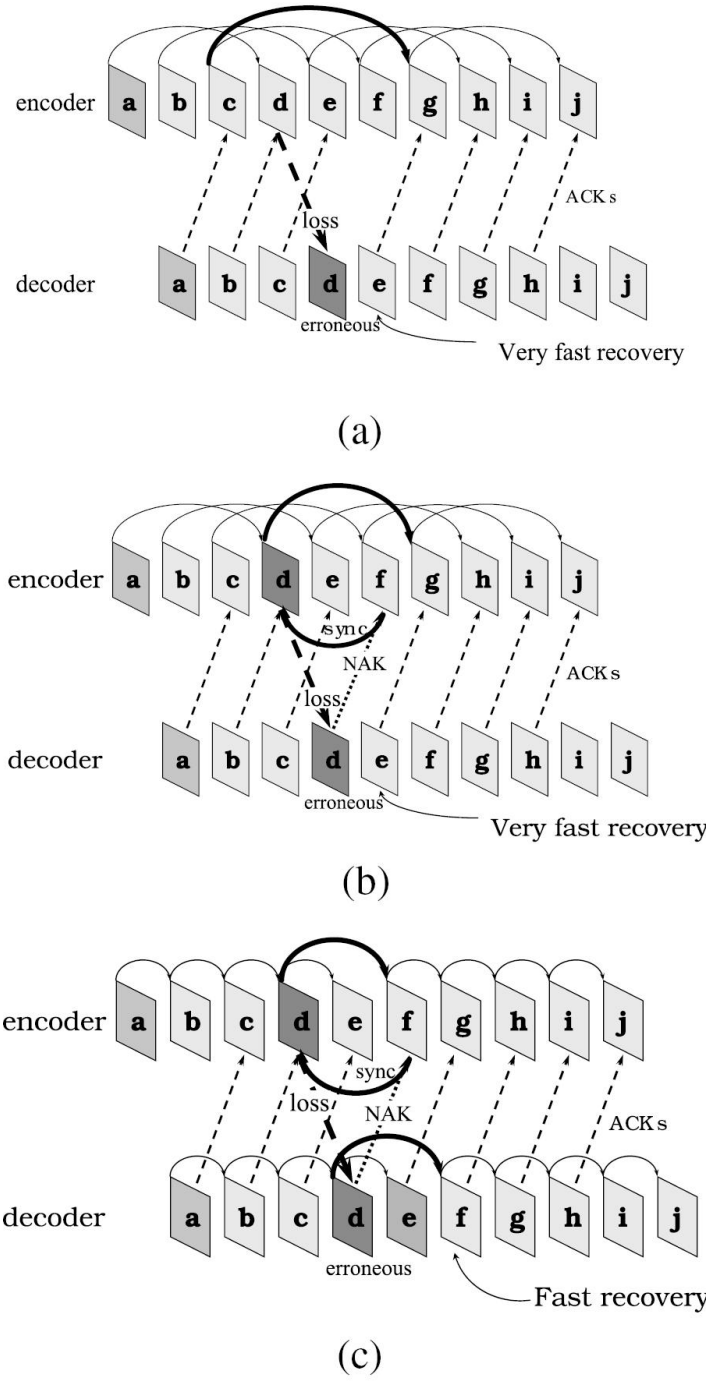

Figure 2.21: Operation of different feedback modes. (a) Feedback Mode 1. (b) Feedback Mode 2. (c) Feedback Mode 3.

Chapter 3

Forward Error Control for Packet Loss and Corruption

3.1 Introduction

In many ways, the Internet (or a wireless network) can be regarded simply as a communication channel in a classical communication system. This chapter discusses the fundamentals of channel protection and the error control techniques used to control the delivery of a multimedia content over the Internet and wireless networks.

The goal of a classical communication system is to transfer the data generated by an information source efficiently and reliably over a noisy channel. The basic components of a digital communication system are: a source encoder (described in chapter 2), a channel encoder, a modulator, a demodulator, a channel decoder, and, finally, a source decoder.
The channel encoder adds redundancy to the information sequence so that channel errors can be detected or corrected. The output of the channel encoder is a finite sequence of symbols called a channel codeword. The set of possible channel codewords is called a channel code. The modulator maps the channel codeword to a signal that is suitable for transmission over a physical channel. The demodulator converts the received signal into a discrete sequence of real numbers of the same length as the channel codeword. The channel decoder tries to recover the input to the channel encoder from the output of the demodulator. Finally, the source decoder produces an estimate of the information sequence.

3.2 Channel Coding and Error Control

Shannon's channel coding theorem states that if the channel capacity is larger than the data rate, a coding scheme can be found to achieve small error probabilities. The basic idea behind channel coding is to add redundancy to each information (payload) packet at the transmitter. At the receiver this redundancy is used to detect and/or correct errors within the information packet. The binary (n, k) BCH code is a common Error Control scheme based on block coding. This code adds redundancy bits to payload bits to form code words and can correct a certain number of bit errors, see [70] for details. An important subclass of non–binary BCH codes are the Reed Solomon (RS) codes. An RS code groups the bits into symbols and thus achieves good burst error suppression capability.
An advantage of the channel coding is constant throughput with fixed (deterministic) delays independent of errors on the channel. To achieve error–free (or close to error-free) communication, however, channel protection schemes must be implemented for the worst case channel characteristics. This results in unnecessary overhead on the typically highly variable wireless links [70]. In particular, when the channel is currently good, the channel protection, dimensioned for the worst case conditions, results in inefficient utilization of the wireless link. In addition complex hardware or software structures may be required to implement the powerful, long codes required to combat the worst case error patterns. It is worth to note that by adding redundancy bits the latency of all packets increases by a constant value. This additional delay may not be acceptable for a streaming session with very tight timing constraints.

In order to overcome the outlined drawbacks of fixed channel protection, *Adaptive Protection* is needed. AP adds redundancy as a function of the current channel characteristics. In particular AP for wireless communication has been studied extensively over the past five years. A large number of adaptive techniques have been specifically designed and evaluated for video streaming applications. The scheme proposed in [71] for instance, estimates the long term fading on the wireless channel and adapts the channel code ratio to proactively protect the packets from loss.
Generally, AP is an important component of an error control strategy and is often used in conjunction with another error control technique called ARQ mechanisms, to form hybrid error protection schemes.

3.3 Automatic Repeat Request, Hybrid FEC/ARQ

Another field of error protection techniques uses retransmissions. In this field it is possible to find ARQ techniques, which are based on error detection and

retransmission of the corrupted packets. Then type I hybrid ARQ protocols that combine error correction coding and ARQ techniques. Finally, type II hybrid-ARQ protocols where the transmitter answers a retransmission request by sending additional parity symbols.

3.3.1 Pure ARQ Protocols

In a pure ARQ system, an information block of length k is encoded into a channel codeword of length n with an error-detecting code. The codeword is sent over the channel and the received word is decoded. If no errors are detected, the transmitted codeword is assumed to be received correctly and needs not be retransmitted. Otherwise, the codeword must be sent again until it is received correctly. In order to send feedback information to the transmitter, the receiver can use a positive acknowledgment (ACK) to indicate that the codeword was received correctly or a negative acknowledgment (NACK) to indicate a transmission error. The efficiency of an ARQ scheme is measured by its reliability and throughput. The literature adopts two main objective measures to evaluate the performance of the above mentioned error control technique. The reliability, that is the probability that the receiver accepts a word that contains an undetectable error and the throughput that is the ratio of the average number of bits successfully accepted per unit of time to the total number of bits that could be transmitted per unit of time [72]. In the following paragraphs are reviewed the most important ARQ schemes.

3.3.1.1 Stop-and-Wait ARQ

In Stop-and-Wait ARQ, the transmitter sends a codeword and waits for an acknowledgment for that codeword. If an ACK is received, the next codeword is sent. If an NACK is received, the same codeword is retransmitted until it is received correctly. Stop-and-wait ARQ has a very simple implementation and is used in many protocol implementation such as at the MAC layer of the IEEE 802.11 standard [73]. The major drawback of the Stop-and-Wait ARQ is the idle time spent by the transmitter waiting for an ACK.

3.3.1.2 Go-Back-N ARQ

In Go-Back-N ARQ, the transmitter sends the codewords continuously without waiting for an acknowledgment. Suppose that the acknowledgment for codeword ci arrives after codewords $c_i, ..., c_{i+N-1}$ have been sent. If this acknowledgment is of the ACK type, the transmitter sends codeword c_{i+N}. Otherwise the codewords $c_i, ..., c_{i+N-1}$ are sent again. On the receiver side when an error is detected in a

received word, this word and the $N-1$ subsequently received ones are ignored. Note that a buffer for N codewords is required at the transmitter side.

3.3.1.3 Selective-Repeat ARQ

Selective-repeat ARQ is similar to go-back ARQ. The difference is that when an NACK for codeword c_i is received, only c_i is retransmitted before the transmission proceeds. In addition to the N-codeword buffer at the transmitter, a buffer is needed at the receiver so that the decoded codewords can be delivered in the correct order. An alternative is to combine selective-repeat ARQ with go-back-N ARQ as in [72] where the transmitter switches from selective-repeat ARQ to go-back-N ARQ whenever μ retransmissions of a codeword have been done without receiving an ACK. Also this protocol has been implemented in a recent standards such as the IEEE802.11e [74].

3.3.2 Hybrid ARQ Protocols

Channel protection and ARQ can be combined to provide, for channels with high error rates, better reliability than fixed or adaptive protection alone and larger throughput than ARQ alone.

3.3.2.1 Type-I Hybrid ARQ Protocols

In a type-I hybrid ARQ system, each information block is encoded with a channel code with error detecting and error correcting capabilities. This can be a single linear code or a concatenation of an error detection code as an outer code and an error correction code as an inner code. If the received word can be correctly decoded, then the decoded codeword is accepted. Otherwise, a retransmission is requested for the codeword.

3.3.2.2 Type-II Hybrid ARQ Protocols

The basic difference between a type-I hybrid ARQ protocol and a type-II hybrid ARQ protocol is that in the latter the transmitter sends additional parity bits instead of the whole codeword when it receives a retransmission request for this codeword.

3.4 Forward Error Control

3.4.1 Motivation

Many techniques have been proposed to protect media data against channel errors. A possible approach is error-resilient source coding, which includes packetization of the information bit stream into independently decodable packets, exploitation of synchronization markers to control error propagation, reversible variable length coding and, where possible, multiple description coding. All this protection techniques are available at the application layer of the transmitter. Another approach is based on error concealment, where the lost or corrupted data is estimated at the receiver side with, for example, interpolation. Error control for media data may also exploit error detection and retransmission (ARQ). One further approach is Forward Error Correction (FEC) with error correcting codes. Finally, one may combine any of the aforementioned methods to obtain the best performance. The choice of an appropriate error control method is not easy because it requires a deep understanding of both the source and the channel behavior. In this respect, many important questions have to be answered: What is the type of the data? Is the data compressed? If yes, what is the compression scheme used? Is the data being transmitted over a wireline or a wireless network? Is there a feedback channel? What are the channel conditions? Moreover, the user requirements must also be taken into consideration. What is more important: reconstruction fidelity or transmission speed? In this section the error control schemes that rely on forward error correction only are described with particular regard to the video transmission. While ARQ techniques have traditionally been the error control method of choice, there are many situations in which they are not suitable. For example, ARQ is not possible when there is no feedback channel. Also, in some applications, such as video multicasting or broadcasting, ARQ can overwhelm the sender with retransmission requests.

3.5 Priority Encoding Transmission

In a packet network, the transmitted packets can be dropped, delayed, or corrupted. By ignoring delayed packets and discarding corrupted ones it is possible to model the channel as a packet erasure channel, which assumes that a transmitted packet is either correctly received or lost.
In order to achieve the best performance for this channel model the systematic Maximum Distance Separable (MDS) codes are applied across blocks of packets. Such codes could be Reed-Solomon (RS) codes, punctured RS codes, or shortened RS codes. RS codes are used for two main reasons. First, as MDS codes, they are

optimal in the sense that the smallest possible number of received symbols is used for full recovery of all information symbols. Second, both the encoding and decoding are very fast when the length of the channel codeword is not too large [75] and may be implemented with relative low complexity at the application layer.
In Priority Encoding Transmission [76], the information bitstream is partitioned into segments with different priorities. Each segment is protected with a systematic RS code. Since the packet number is indicated in the packet header, the receiver knows the location of the erased symbols in each codeword. So, if the RS code used for a given segment is known, the receiver is able to reconstruct the segment when the number of packets lost does not exceed the number of parity symbols for this code.

3.6 Error Protection for Wireless Networks

In fading channel, the transmitted packets experience different bit error rates and usually a fading channel is modelled using a two state Markov model (the two states are good and bad): packets transmitted when the channel is in the bad state are exposed to much higher bit error rates than those transferred during the good state of the channel. Thus, to avoid decoding failures, codes should be designed for the bit error rate in the channel's bad state. This causes overprotection during the good state of the channel (which usually lasts much longer) and bounds the achievable performance from the theoretical limits. In order to alleviate this phenomenon others successful extensions of the classical CRC/RCPC system of [77] for fading channels are provided. The first system [78] introduces interleaving, while the other systems described in [79] is based on product channel codes.

3.6.1 Interleaving

Interleaving tends to spread deep fade and to transform a memory channel into a memoryless one. It improves the performance during transmission over fading channels at the expense of increased complexity and time delay. A system that exploits block interleaving to alleviate the problems of channel burst errors during a deep fade is proposed in [78]. As in the system of [77], an embedded bit stream is encoded with a punctured convolutional coder, and the decoding of the received bit stream is stopped when the first decoding error is detected.

3.6.2 Product Code System

[80] proposed a transmission system based on a product channel code to protect the embedded information bit stream. The product code uses the concatenated

CRC/RCPC code of [77] as the row code and a systematic RS code as the column code. The main idea is to strengthen the protection of the CRC/RCPC code by using channel coding across the packets.
Finding an optimal RCPC code rate, an optimal RS code rate, and an optimal interleaver for the system is a very difficult problem. Moreover, no efficient method that computes a near-optimal solution is known. Only in [80] it is suggested to select RCPC code so that it can efficiently protect the transmitted data while the channel is in the good state. In addition to equal error protection, several ways of implementing unequal error protection were proposed in [80]. The most successful one protects the earliest symbols of the embedded bit stream by additional RS codes.

Chapter 4

IP Network-Adaptive Media Transport

4.1 Introduction

For the best end to-end performance, Internet media transmission must adapt to changing network characteristics; it must be network adaptive. It should also be media aware, so that adaptation to changing network conditions can be performed intelligently.

Internet packet delivery is characterized by variations in throughput, delay, and loss, which can severely affect the quality of real-time media. The challenge is to maximize the quality of such video service at the receiver, while simultaneously meeting bit-rate limitations and satisfying delivery time constraints. For the best end-to-end performance, Internet media transmission must adapt to changing network characteristics; it must be *network adaptive*. It should also be *media aware*, so that adaptation to changing network conditions can be performed intelligently. In general the issues of adaptation may be solved with the so-called *network-adaptive media transport* that perform intelligent adaptation when network conditions changes.
A streaming media system is composed by four major components that should be designed and optimized together:

1. The *encoder application* compresses video signals and uploads them to the media server;

2. The *media server* stores the compressed media streams and transmits them on demand, often serving hundreds of clients simultaneously;

3. The *transport mechanism* delivers media packets from the server to the

clients for the best possible user experience, while sharing network resources;

4. The *client application* decompresses and renders the video and audio packets and implements the interactive user controls.

The streaming media client typically employs error detection and concealment to mitigate the effects of lost packets. These techniques have been discussed in Chapter 2. To adapt to network conditions, the server receives feedback from the client, e.g., as positive or negative acknowledgments. More sophisticated client feedback might inform about packet delay and jitter, link speeds, or congestion. The media server can implement intelligent transport by sending the right packets at the right time, but the computational resources available for each media stream are often limited because a large number of streams must be served simultaneously. Out of the burden of an efficient and robust system is therefore on the encoder application, which, however, cannot adapt to the varying channel conditions and must rely on the media server for this task.

4.2 Rate Distortion Optimized Streaming

The base work on *IP-Network Adaptive Media Transport* starts with Chou and Miao in the so-called Rate-Distortion Optimized (RaDiO) streaming [81]. They consider streaming as a stochastic process, with the goal of determining both which packets to send and when to send them, with the constraint of minimizing the reconstructed distortion at the client for a given average transmission rate. The base scenario considers a media streaming server that has stored a compressed video streams that have been packetized into data units. Each data unit has a size in bytes B_l and a deadline by which it must arrive at the client in to be useful for decoding. The importance of each data unit is captured by its *distortion reduction* ΔD_l, a value representing the decrease in distortion that results if the data unit is decoded. The distortion is often expressed as MSE. If a data unit can be decoded often depends on which other data units are available. In the RaDiO framework, these interdependencies are expressed in a Directed Acyclic Graph (DAG) [15]. An example of dependency graph is shown for SNR-scalable video encoding with I, P and B frames (see Fig. 4.1). Each block represents a data unit and the arrows indicate the decoding order of each data. The RaDiO framework can be used to choose the optimal set of data units to transmit at successive transmission opportunities. These transmission opportunities are assumed to occur at regular time intervals that depends on the available channel bandwidth. Because of decoding dependencies among data units the importance of transmitting a packet at a given transmission opportunity often depends on which packets will be transmitted in

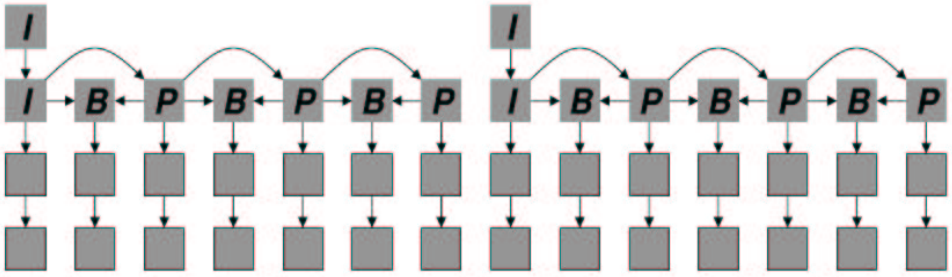

Figure 4.1: A Directed Acyclic Graph captures the decoding dependencies for an SNR-scalable encoding of video with I-frames, P-frames, and B-frames. Squares represent data units and arrows indicate decoding order.

the future. Therefore, the scheduler makes transmission decisions based on an optimized plan that may includes to anticipate later transmissions. For simplicity and in order to keep the system simple only a finite number of data units partecipate at the optimization process.
The plan that control the data unit transmissions is called a *transmission policy*, π. Assuming a time horizon of N transmission opportunities, π represents a set of length-N binary vectors π_l, with one vector for each data unit l. In this representation, the N binary elements of π_l indicate whether the data unit l will be transmitted at each of the next N transmission opportunities. The policy needs to take into account future acknowledgments that might arrive from the client to indicate that the packet has been received. Each transmission policy leads to its *error probability*, $\epsilon(\pi_l)$, defined as the probability that data unit l arrives at the client side. Each policy is also associated to an expected number of times that the packet is transmitted under the policy, $\rho(\pi_l)$. The goal of the scheduler is to find a transmission policy π with the best tradeoff between expected transmission rate and expected distortion. At any transmission opportunity the optimal π minimizes the Lagrangian cost function:

$$J(\pi) = D(\pi) + \lambda R(\pi) \tag{4.1}$$

where the expected transmission rate is:

$$R(\pi) = \sum_{l} \rho(\pi_l) B_l \tag{4.2}$$

while the expected reconstruction distortion is:

$$D(\pi) = D_0 - \sum_l \Delta D_l \prod_{l' \preccurlyeq l} (1 - \epsilon(\pi'_l)) \tag{4.3}$$

The Lagrange multiplier λ controls the tradeoff between rate and distortion. In (4.3) D_0 is the distortion if no data units arrive at the receiver, while D_l is the distortion reduction if data unit l arrives on time and can be decoded successfully. The product term $\Pi(1 - \epsilon(\pi_l))$ is the probability that this fact occur. The notation $l \preccurlyeq l'$ identifies the set of data units that must be presented in order to correctly decode data unit l. In the above formulation, delays and losses experienced by packets transmitted over the network are assumed to be statistically independent. Packet loss is typically modelled as Bernoulli with some probability, and the delay of arriving packets is often assumed to be a shifted-Γ distributed. Expressions for 4.2 and 4.3 can be derived in terms of the Bernoulli loss probabilities, the cumulative distribution functions for the Γ-distributed delays, the transmission policies and transmission histories, and the data units arrival deadlines.
The scheduler re-optimizes the entire policy π at each transmission opportunity to take into account new information since the previous transmission opportunity, and then executes the optimal π for the current time. An exhaustive search to find the optimal π is not generally tractable in terms of computational complexity; the search space grows exponentially with the number of considered data units, M, and the length of the policy vector, N as explained in [82]. Even though rates and distortion reductions are assumed to be additive (note that this assumption is valid only when losses are separated so that burst losses are not accounted), the graph of packet dependencies leads to interactions, and exhaustive search would have to consider all 2^{MN} possible policies. Chou and Miao's RaDiO framework [15] overcomes this problem by using an iterative algorithm. Their Iterative Sensitivity Adjustment (ISA) algorithm minimizes (4.1) with respect to the policy π_l of one data unit while the transmission policies of other data units are fixed. Data units' policies are optimized one at a time until the Lagrangian cost converges to a (local) minimum.

Several techniques have been proposed to further reduce the complexity of the basic RaDiO algorithm. Chou and Sehgal have presented simplified methods to compute approximately optimized policies [83]. An attractive alternative to ISA is a randomized algorithm recently developed in [84–85] in which heuristically and randomly generated candidate policies are compared at each transmission opportunity. The best policy from the previous transmission opportunity is one of the candidates and thus past computations are efficiently reused. With a performance similar to ISA, the randomized algorithm usually requires much less computation. Despite the enormous literature contributions in developing a solution for the Ra-

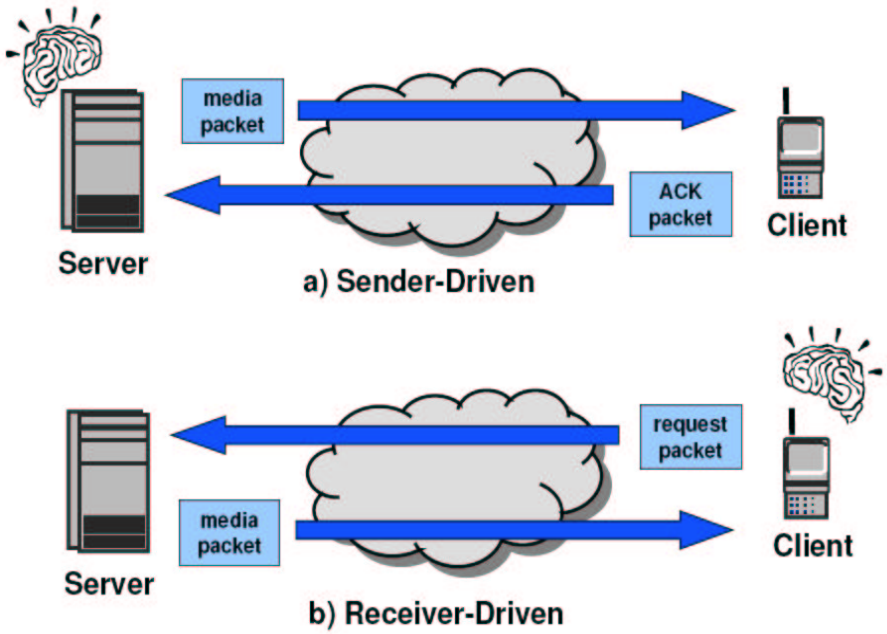

Figure 4.2: (a) Sender-driven streaming: the server computes an optimal sequence of media packet to transmit and the client acknowledges packets upon reception. (b)Receiver-driven streaming: the complexity is shifted to the client. The client computes an R-D optimized sequence of requests to send to the server and the server only needs to respond to the client's requests.

DiO framework, the computational complexity associated to the algorithm that find the optima policy vector remains the main limiting factor.

4.2.1 Advances RaDiO techniques: Receiver Driven Streaming

When transmitting many video streams simultaneously, a media server might become computation-limited rather than bandwidth-limited. It is therefore desirable to migrate the computation required for network-adaptive media transport from the server to the client. Fortunately, rate-distortion optimized streaming can also be performed when the algorithm runs at the client so that very little computation is required at the server side [86]. For *Receiver-Driven Streaming*, the client is provided the information about the sizes, distortion reduction values and interdependencies of the data units available at the server ahead of time. The size of this *hint track* or *rate-distortion preamble* is small compared to the media stream and may be transmitted with limited overhead. The receiver uses this information to compute a sequence of requests that specify the data units that the server should

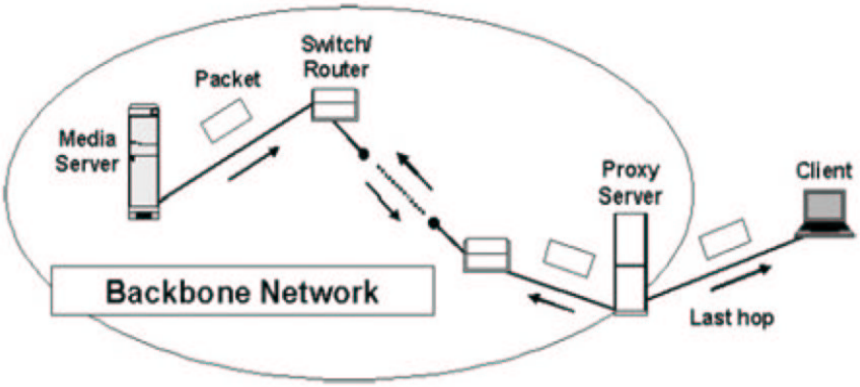

Figure 4.3: Proxy-driven RaDiO Streaming. A proxy server located between the backbone network and a last hop link uses a hybrid of receiver- and sender-driven RaDiO streaming to jointly optimize requests to send to the server and media packets to forward to the client.

transmit. In order to do this it is necessary to adapt the algorithm discussed in the previous section to compute a sequence of requests that give an optimal tradeoff between the expected transmission rate of the media packets that the server will send and the expected distortion that will result [86].
Fig. 4.2 illustrates the differences between sender-driven and receiver-driven streaming approaches. By combining sender-driven and receiver-driven techniques, the RaDiO framework can be extended to different network topologies. For example, RaDiO might be implemented in a proxy server placed between the backbone network and the last hop link (Fig. 4.3) [87]. The proxy coordinates the communication between the media server and the client using a hybrid of receiver and sender driven streaming. End to end performance is improved if compared to a sender or receiver-driven RaDiO system because the traffic created by retransmissions of media packets lost in the last hop to the client does not congest the backbone link.

4.2.2 Advances RaDiO techniques: Rich Acknowledgments

In one extension to the RaDiO framework, streaming performance is improved through the use of *rich acknowledgments* [88]. In sender-driven RaDiO streaming using conventional acknowledgments, when a client receives a media packet, the client sends an ACK to the server. If the ACK packet is lost, the server may decide to unnecessarily retransmit the packet at the expense of other packets. With

rich acknowledgments, the client does not acknowledge each data unit separately. Instead, it transmits periodically a packet that positively acknowledges all packets that arrived so far and negatively acknowledges (NACK) packets that have not yet arrived to the destination. In this way a rich ACK packet provides a snapshot of the state of the receiver buffer. Rich acknowledgments require some changes to the basic RaDiO framework described previously as shown in [15]. The *rich acknowledgment* scheme outperforms conventional RaDiO scheme with conventional ACKs for all transmission rates. The improved performance of the rich acknowledgment scheme is due to the robust transmission of the feedback information. With rich acknowledgments, the effect of a lost feedback packet is mitigated because subsequent feedback packets contain the same (updated) information. In addition, because rich acknowledgment packets also provide NACKs, there is less ambiguity for the server to solve. In the case of conventional feedback, a non-acknowledged transmission may be due to a lost media packet or to a lost acknowledgment packet.

4.2.3 Congestion Distortion Optimized scheduling (CoDiO)

RaDiO streaming and its various extensions described do not consider the effect that transmitted media packets may have on the delay of subsequently transmitted packets. Delay is modelled as a random variable with a parameterized distribution. All parameters are adapted slowly according to feedback information. In case of the media stream is transmitted at a rate that is negligible compared to the minimum link speed on the path from server to client, this may be an acceptable model. In the case where there is a bottleneck link on the path from server to client, packet delays can be strongly affected by *self-congestion* resulting from previous transmissions. It is proposed in [85] a Congestion-Distortion Optimized (CoDiO) algorithm which takes into account the effect of transmitted packets on delay. The scheme is intended to achieve an R-D performance similar to RaDiO streaming but specifically schedules packet transmissions so that it yields an optimal tradeoff between reconstruction distortion and congestion, measured as average delay, on the bottleneck link. As with RaDiO, transmission actions are chosen at discrete transmission opportunities by finding an optimal policy over a time horizon. However, in CoDiO the optimal policy minimizes the Lagrangian cost $D + \lambda\Delta$ where D is the expected distortion due to the policy and Δ is the expected end-to-end delay which measures congestion. CoDiO's channel model assumes a succession of high-bandwidth links shared by many users, followed by a bottleneck last hop only used by the media stream under consideration. CoDiO needs to know the capacity of the bottleneck, which can be estimated, e.g., by transmitting a sequence of packets [89]. The channel model is used to calculate the expected distortion D due to packet loss and the expected end-to-end

delay Δ. CoDiO outperforms RaDiO because it distributes transmissions in time and attempts to send packets as late as safely possible. This reduces the load in the bottleneck queue and hence the average end-to-end delay. Other applications sharing the network experience less congestion's RaDiO, on the other hand, is less network-friendly. As the scheduler considers only average rate, its traffic tends to be more bursty.

4.3 Conclusions

In this chapter we have discussed network adaptive media transport through the RaDiO framework for rate distortion optimized media streaming. After reviewing the basic framework several extensions and enhancements that have been proposed, are considered. The framework can be implemented in a media server or, alternatively, at the client. However the main limiting factor remains the high computational complexity required to find the optimal packet to transmit at each transmission opportunity.

Chapter 5

Distortion Estimation Models

5.1 Introduction

Video channel distortion modelling represents a challenging task mainly due to the difficulties in mapping the channel characteristics into the video distortion model. These difficulties are related to:

- the negative effects that a single packet loss produces, not only on a single frame, but also on the following ones, because of the error propagation. In fact, by adopting predictive coding the loss of a single frame may result in a distortion of the lost frame and in a damage of all the other frames in the same GOP;
- the unpredictable distortion envelope produced by burst losses, related to the problems in recovering the missed frames.

As for the approaches already examined and presented in the literature, in some recent papers it has been often implicitly assumed that burst length may be neglected, focusing on the average packet loss rate as the most important feature. In [90], the authors carefully analyze the distortion due to a single frame loss, taking into account the error propagation, the intra refresh and the spatial filtering, their model considers the effects of multiple losses as the superposition of multiple independent losses, leading to an expected distortion that is proportional to the average PLR. This assumption provides realistic results when losses are spaced sufficiently far apart, but in low bit-rate wireless video communication this hypothesis may fail. In this scenario each coded frame may fit within a single packet and the losses may be bursty, resulting in the loss of multiple successive frames. In [91], the authors show that generally longer bursts lead to larger distortions. However, there is still a certain difference between their model and the real reconstructed quality. An analytical model for the distortion is proposed in [92],

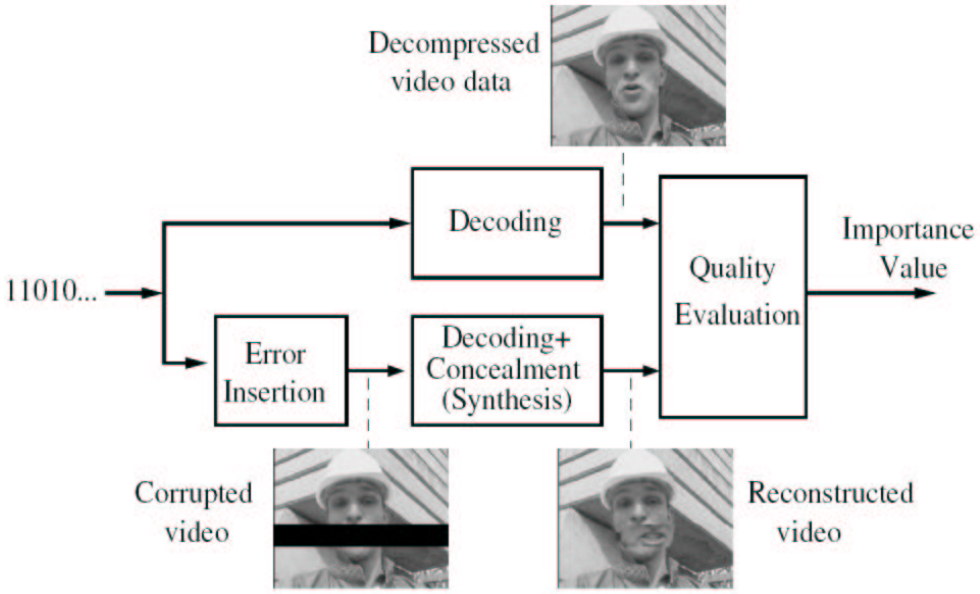

Figure 5.1: Conceptual scheme of the analysis-by-synthesis technique.

where it is assumed that all frames, after the lost frame, are not available at the decoder. By consequence some frames that may be used to improve the video quality, are not considered in their model. Moreover, most papers in the literature consider scenarios where PLR is low [93].

In the following sections several distortion estimation techniques will be reviewed detailing the main features and the main limitation aspects.

5.2 Perceptual Distortion Classification

The quality of multimedia communications over packet networks may be impaired in case of packet loss. The amount of quality degradation strongly vary depending on the importance of the lost data. In order to design efficient loss protection mechanisms, a reliable importance estimation method for multimedia data is needed. Such importance is often defined a priori, based on the average importance of the elements of the compressed bitstream, as the data partitioning approach. In order to provide a quantitative importance estimation method at a finer level of granularity, [94] defines the importance of a video coding element, such as a macroblock or a packet, as a value proportional to the distortion that would be introduced at the decoder by the loss of that specific element. The potential

distortion of each element, could, therefore, be computed using the Analysis-By-Synthesis technique. The conceptual scheme is depicted in Fig. 5.1. The video sequence has to be coded and packetized before the activation of the algorithm. The Analysis-By-Synthesis distortion estimation algorithm performs, for each packet, the following steps:

- decoding, including concealment, of the bitstream simulating the loss of the packet being analyzed (synthesis stage);
- quality evaluation, that is computation of the distortion caused by the loss of the packet. The original and the reconstructed picture after concealment are compared using MSE;
- storage of the obtained value as an indication of the perceptual importance of the analyzed video packet.

The previous operations can be implemented with small modifications of the standard encoding process. The encoder, in fact, usually reconstructs the coded pictures simulating the decoder operations, since this is needed for motion compensated prediction. If step (1) of the analysis-by-synthesis algorithm exploits the operations of the encoding software, complexity is only due to the simulation of the concealment algorithm. The analysis-by-synthesis technique, as a principle, can be applied to any video coding standard. In fact, it is based on repeating the same steps that a standard decoder would perform including the error concealment. Obviously, the importance values computed with the analysis-by-synthesis algorithm are dependent on a particular encoding, that is if the video sequence is compressed with a different encoder or using a different packetization, values will be different. Note, however, that in principle the analysis-by-synthesis scheme does not impose any particular restriction on encoding parameters or packetization. Due to the interdependencies usually present between data units, the simulation of the loss of an isolated data unit is not completely realistic, particularly for high packet loss rates. Every possible combination of events should ideally be considered, weighted by its probability, and its distortion computed by the analysis-by-synthesis technique, obtaining the expected distortion value. The application of the analysis-by-synthesis method is straightforward when considering elements of the video stream which does not contribute to later referenced frames, since the mismatch due to concealment does not propagate. If propagation is possible, the distortion caused in subsequent frames should be evaluated until it becomes negligible, for instance, at the beginning of the next GOP for MPEG video, or until its value falls below a given threshold. In this case, the complexity of the analysis-by-synthesis approach is high giving it suitable only for stored-video scenarios that allow precomputation of the perceptual distortion.

5.3 Distortion Matrix

The Distortion Matrix (DM) proposed in [95] allows to calculate the distortion caused by dropping frames in a GOP structured video stream. The distortion needs to be evaluated during offline simulations. When calculating the distortion, it is assumed that a simple copy previous frame error concealment scheme is used by the decoder. Once a specific P frame or I frame is lost, all the depending frames in this GOP are replaced with the latest successfully decoded frame. This assumption makes this model unsuitable because today all video decoders tend to mitigate the error propagation reducing the distortion after a single loss. In the scheme proposed in [95] the additional distortion for a particular dropping pattern is the sum of the individual frame distortions of the concealed pictures. Also this hypothesis gives unrealistic the model because the distortion is not additive. The Distortion Matrix proposed in [95], for a GOP with $IB_1B_2P_1B_3B_4P_2B_5B_6$ encoding structure, is given as follows:

$$\begin{pmatrix} D_I^R & D_{B_1}^R & D_{B_2}^R & D_{P_1}^R & D_{B_3}^R & D_{B_4}^R & D_{P_2}^R & D_{B_5}^R & D_{B_6}^R \\ / & D_{B_1}^I & D_{B_2}^I & D_{P_1}^I & D_{B_3}^I & D_{B_4}^I & D_{P_2}^I & D_{B_5}^I & D_{B_6}^I \\ / & / & / & / & D_{B_3}^{P_1} & D_{B_4}^{P_1} & D_{P_2}^{P_1} & D_{B_5}^{P_1} & D_{B_5}^{P_1} \\ / & / & / & / & / & / & / & D_{B_5}^{P_2} & D_{B_5}^{P_2} \\ / & / & D_{B_2}^{B_1} & / & / & / & / & / & / \\ / & / & / & / & / & D_{B_4}^{B_3} & / & / & / \\ / & / & / & / & / & / & / & / & D_{B_6}^{B_5} \end{pmatrix} \tag{5.1}$$

where $D_{F_{loss}}^{F_{ref}}$ are the MSE values observed when replacing frame loss F_{loss} by F_{ref} as part of the concealment strategy. The column left to the distortion matrix shows the replacement frame F_{ref} for every row of the matrix. For instance, $D_{B_1}^I$ represents the additional reconstruction distortion if the first B frame of the GOP is lost and therefore replaced by the I frame of that GOP. R is a frame from the previous GOP that is used as a replacement for all frames in the current GOP if the I frame of the current GOP is lost.
The number of entries of the distortion matrix can be calculated as follows:

$$N_{entries} = \frac{1}{2}L(3 + \frac{L}{N_B + 1}) \tag{5.2}$$

where L is the length of the GOP, and N_B is the number of B frames between two P or I frames; given this matrix, the RD-optimized frame dropping strategy for streaming video chooses between four possible dropping decisions that can be made for each stream: dropping I frame, dropping P frame, dropping B frame and then dropping nothing. As part of the dropping decision, all depending frames in the same GOP are also dropped. For example, if the dropping strategy decides

to drop the I frame of a GOP, this involves dropping all other frames from the GOP. Also, if the strategy expects to drop a P frame, this involves dropping all depending B and P frames of this GOP. Although many possible dropping choices are available, previous dropping decisions and also the position of the frames can limit the computational complexity as stated in [95]. When the DM is evaluated, simple copy previous frame error concealment is assumed and it is also used as the error concealment scheme at the decoder in [95]. Although the actual error concealment scheme might be more sophisticated the DM requires an high overhead for transmitting the DM elements and does not considers the fact that the decoder uses the correctly received frames to reduce the error propagation effect.

5.4 Advance Estimation Technique: ROPE

All the previous explained end to end distortion estimation techniques may be categorized as either "block-based" or "frame-based" methods. On the other hand are other more accurate techniques classified as "pixel-based" methods. In particular the block-based approach generates and recursively updates a block-level distortion map for each frame [60, 96–97]. However, since inter-frame displacements involve subblock motion vectors, a motion compensated block may inherit errors propagated from multiple blocks in prior frames. Hence, block-based techniques must involve a possibly rough approximation (for example, weighted averaging of propagated block distortion [60], whose errors may build up to significantly degrade estimation accuracy. In contrast, pixel-based approaches track the distortion estimate per pixel and have the potential to provide high accuracy. The obvious question concerns of complexity. One extreme approach was proposed in [98] where the distortion per pixel is calculated by exhaustive simulation of the decoding procedure and averaging over many packet loss patterns. Another pixel based approach was proposed in [99], where only the two most likely loss events are considered. However, it turns out that low complexity can be maintained without sacrificing optimality as has been demonstrated by the ROPE in [59]. ROPE recursively calculates the first and second moments of the decoder reconstruction of each pixel, while accurately taking into account all relevant factors, including error propagation and error concealment. ROPE has been applied for end to end estimation in numerous RD optimization based coding techniques, including: intra-/inter-mode selection [59], [100] and extension thereof to layered coding [101], [102], multiple description coding [103], prediction reference frame and/or motion vector selection [104], joint video coding and transport optimization [105–107] etc. Variants of ROPE have been applied in the transform domain to estimate the end to end distortion of DCT coefficients [108]. Beside distortion estimation, other applications of ROPE has been presented in robust video coding

error resilient rate control [109]. Finally, ROPE has been proposed also for video quality monitoring and assessment in video streaming over lossy networks [110].

5.4.1 ROPE: Open Issues and Limitations

However, despite the interest and extensive work on ROPE applications, there are unsolved problems that significantly restrict its application in practical video coding and streaming system scenarios. The most important open question regards the emergence of the *crosscorrelation* terms in the equation that estimate the user quality due to pixel filtering (or averaging) operations. In fact the various forms of pixel filtering operations performed by standard encoders, e.g., subpixel motion compensation, intra-prediction, deblocking filtering [14] are difficult to model. Also the Discrete Cosine Transform (DCT) can be viewed as a special form of pixel filtering/averaging. Moreover, pixel-averaging operations may also be performed by the decoder that adopt error concealment [111]. Within the exact ROPE procedure, such pixel filtering operations may require computation and storage of cross-correlation values for all pixel pairs in the frame, which is of impractical complexity, so that, effective and low-complexity Cross-Correlation Approximation (CCA) is highly desirable. Other alternative approaches have been proposed in literature where the cross-correlation terms are computed but only within a predefined inter-pixel distance. However, all the proposed techniques require substantial computation and storage complexity. Moreover the scenario considered in the majority of the cited papers include communication where the loss rate is relatively low. From this perspective the algorithms presented in the following chapter, that represents the main contribution of this thesis, are able to evaluate accurately the distortion envelope derived from environments characterized by high loss rates.

Part II

Original Contribution

Chapter 6

Video Distortion Estimation Algorithms

6.1 Introduction

In the following sections three novel algorithms for channel distortion estimation are presented. The algorithms are able to take explicitly into account the loss pattern caused by the transmission over an error prone network. In order to explain better the modifications added to the internal encoder structure and the algorithms operations, the following parts consider the Quarter Common Intermediate Format (QCIF) *Foreman* test sequence as a reference. The sequence is encoded following the JVT H.264/AVC standard (and in particular using the reference software JM98 where we have added all modifications).

During the encoding procedure, two outputs are provided by the encoder as it is showed in Fig. 6.1:

- the compressed bitstream;
- a reference uncompressed sequence.

The compressed bitstream is packetized, by the network transport layer, and then is transmitted over the channel. The reference sequence instead, is used by the encoder to perform internal operations such as motion prediction, motion compensation as well as rate control. The uncompressed sequence, stored in the internal encoder buffer, is used extensively to perform the distortion estimation.

In the following it is assumed that the encoder is able to evaluate two amplitude arrays, $A^1_{i,l}$ and $A^2_{i,l}$, with $i \geq l > 2$, defined by:

$$A^1_{i,l} = \mathrm{MSD}[f_i - f_{l-1}], \tag{6.1a}$$

$$A^2_{i,l} = \mathrm{MSD}[f_i - f_{l-2}], \tag{6.1b}$$

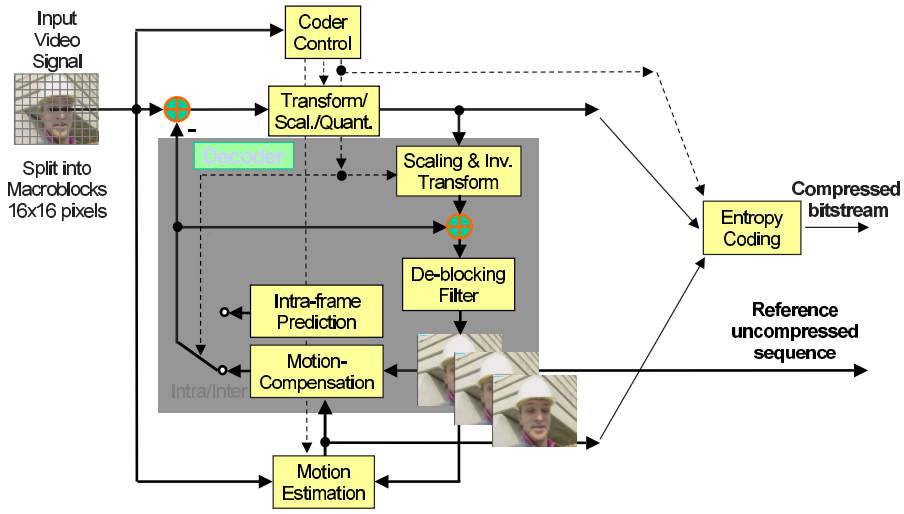

Figure 6.1: Main outputs provided by the encoder.

where $\mathrm{MSD}[f_i - f_{l-1}]$ represents the Mean Square Difference between frames f_i and f_{l-1}. The defined arrays give an estimate of the channel loss induced distortion, as explained in the following. More precisely, the MSD is used to estimate the actual MSE at the decoder. Moreover, it is assumed that a single frame fits in a single packet. It is worth noticing that this hypothesis will be removed but for simplicity now it is desirable to considers frame or packet as the same entity. By consequence, a single channel packet loss produces the loss of the whole frame.

Observe that MSE takes into account the distortion introduced by the channel only, and the source compression distortion is neglected. In fact, the intent of the employed algorithms is to model the channel distortion. So, in the sequel, it is shown that this assumption is acceptable in most of the channel conditions examined.

First of all, at the encoder side the bitstream is converted into prediction information and transform coefficients allowing the reconstruction of the current frame for internal rate control operations. In order to do this every uncompressed frame is forwarded also in a reference frame buffer, giving the chance to allow the prediction of the next frame inside the motion compensation process. The bitstream is then packetized and the obtained packet is transmitted by the network transport layer. In case of successful reception, the packet is forwarded directly to the decoder for the decoding operations while, if the packet is loss, the simplest

Figure 6.2: Decoder output due to loss of frame 4. At the top frames 5, 6 and 7 of the original sequence. At the bottom frames 5, 6 and 7, of the reconstructed sequence.

operation the decoder can perform is just to skip the decoding and not update the display buffer. In this case, the user will immediately recognize the loss, as the fluent motion and continuous display update is not maintained. However, this is not the only problem: not only the display buffer is not updated but also the reference decoder buffer has a picture gap. Even in case of successful reception of the next packet the corresponding decoded frame will differ from the reconstructed frame at the encoder side, given that the encoder and the decoder are referring to a different reference signal while decoding this packet. Therefore, the loss of a single packet has also effects on the quality of the following frames. To show this effect, the QCIF *Foreman* test sequence is encoded with the H.264 encoder with a GOP equal to 15. From the packetized bitstream the Frame 4 is removed and the obtained bitstream is fitted to the decoder. Fig. 6.2 shows the error propagation effects at Frames 5, 6 and 7 due to the loss of Frame 4. In order to enhance the quality of the picture, the decoder could be modified using a picture flag to specify that the lost frame should be replaced by the last correctly decoded frame. Moreover, the lost frame could be prevented from being used as reference frame for the decoding of the subsequent frames. Fig. 6.3 shows the visual quality improvement obtained by this simple modification. In the following this technique is referred as *Frame Copy* (FC) EC. When a burst of packet losses occurs, the FC-EC gives a typical frozen frame output effect. The image on the screen stops, when a loss

Figure 6.3: Output of the modified decoder due to loss of frame 4. At the top frames 5, 6 and 7 applying the FC-EC. At the bottom frames 5, 6 and 7 applying the MC-EC.

occurs, and is kept constant (i.e. frozen) during the entire burst. To remove this phenomenon and give a more fluid effect during the bursts, the *Motion Copy* (MC) EC technique has been implemented in the video decoder. The MC-EC is based on a motion vector copy of the last correctly received frame. The visual effect, obtained applying MC-EC is shown in Fig. 6.3.
In order to allow the estimation algorithms to operate, it is also assumed that the application is able to determine the actual sequence loss pattern by exchanging some suitable signaling information with the lower layers, which adopt, for instance, an acknowledgment based transmission technique.

In order to explain better the distortion estimation algorithms proposed in this thesis, assume that the following loss pattern occurs in a GOP.

Suppose now that frames from index l_i to index i, and frames from index l_j to index j are lost, being $i < l_j - 1$. The considered scenario, based on two non overlapped bursts, may be extended to a generic number of bursts, so that in the following it is considered, $l_i \leq i < l_j - 1 \leq j - 1$.

6.2 Channel Distortion Estimation Algorithms

6.2.1 Step Distortion Algorithm

The first estimation algorithm is called Step Distortion Algorithm (SDA) and is able to estimate the distortion at the k-th frame in a GOP using the following method:

$$D(k) = \begin{cases} 0 & k < l_i \\ A^1_{k,l_i} & l_i \leqslant k < i \\ A^1_{i,l_i} & i \leqslant k < l_j \\ A^1_{k,l_j} & l_j \leqslant k < j \\ A^1_{j,l_j} & k \geqslant j \end{cases} . \tag{6.2}$$

As stated, the SDA algorithm approximates the distortion envelope using a simple step function, and it is completely defined by the amplitudes at each time step k. In this way the SDA assumes that the decoder is able at least to keep constant the distortion. Differently from the work in [92], SDA assumes that the decoder has activated some error recovery techniques such as FC or MC error concealment.

6.2.2 Exponential Distortion Algorithm

The Exponential Distortion Algorithm (EDA) reproduces more accurately than SDA the distortion envelope caused by isolated losses.
Consider that, when a loss appears the distortion ramps up in correspondence of the missed frame, since the decoder applies some recovery techniques to alleviate the visual effect. Due to error propagation, which is caused by predictive coding, the MSE associated with subsequent frames exhibits a nonzero value. More precisely, the distortion decreases as a consequence of the spatial filtering and the intra refresh, as detailed in chapter 2, until it eventually becomes zero at frames sufficiently apart from the lost one.

To take into account this amplitude decay effect, the EDA models the distortion at the k-th frame as follows:

$$D(k) = \begin{cases} 0 & k < l_i \\ A^1_{k,l_i} & l_i \leqslant k < i \\ A^1_{i,l_i} e^{-b(k-i)} & i \leqslant k < l_j \\ A^1_{k,l_j} & l_j \leqslant k < j \\ A^1_{j,l_j} e^{-b(k-j)} & k \geqslant j \end{cases} , \tag{6.3}$$

where the parameter b is introduced to shape the error propagation effect. In particular, b can be split into two different parts, corresponding to the separate contributions due to the encoder and the decoder operations:

$$b = b_{\text{enc}} + b_{\text{dec}}. \tag{6.4}$$

From the encoder point of view b_{enc} depends on the intra coded macroblock ratio, on the rate-control algorithm, on the number of reference frames stored in the encoder buffer to perform motion estimation and motion compensation as well as on the intra refresh period. From the decoder point of view instead, the parameter b_{dec} depends primarily on the employed mitigation scheme.

6.2.3 Advances Distortion Algorithm

The SDA and the EDA provide an acceptable distortion approximation for isolated bursts of lost packets. When the distance between bursts get smaller (especially when the channel exhibit bad conditions), both the SDA and the EDA algorithms may lead to an optimistic evaluation of the channel induced distortion. A more precise estimation of the actual distortion is provided by the Advanced Distortion Algorithm (ADA).
In this case, the distortion at the k-th frame in a GOP may be evaluated as:

$$D(k) = \begin{cases} 0 & k < l_i \\ A^2_{k,l_i} & l_i \leqslant k < i \\ A^2_{i,l_i} e^{-b(k-i)} & i \leqslant k < l_j \\ A^2_{k,l_j} & l_j \leqslant k < j \\ A^2_{j,l_j} e^{-b(k-j)} & k \geqslant j \end{cases} . \tag{6.5}$$

So the unique modification, with respect to the EDA, is the selection of a different reference frame. In particular, ADA tries to provide a more accurate approximation of the distortion envelope using not the last received frame (as in the SDA and in the EDA), but the previous one. This simple modification, which implicitly introduces an additional distortion term in the estimation process, exploits the fact that successive frames in a scene contain little detail variations and allows a better approximation of the distortion envelope.
As stated for EDA, a suitable choice of the parameter b may lead to a more accurate distortion estimation.

6.3 Performance Measurements

This section presents the results of several experiments, that are run to evaluate the capability of the proposed algorithms in estimating the distortion envelope. Many encoding and decoding settings are extensively examined, to assess the generality of the proposed techniques.

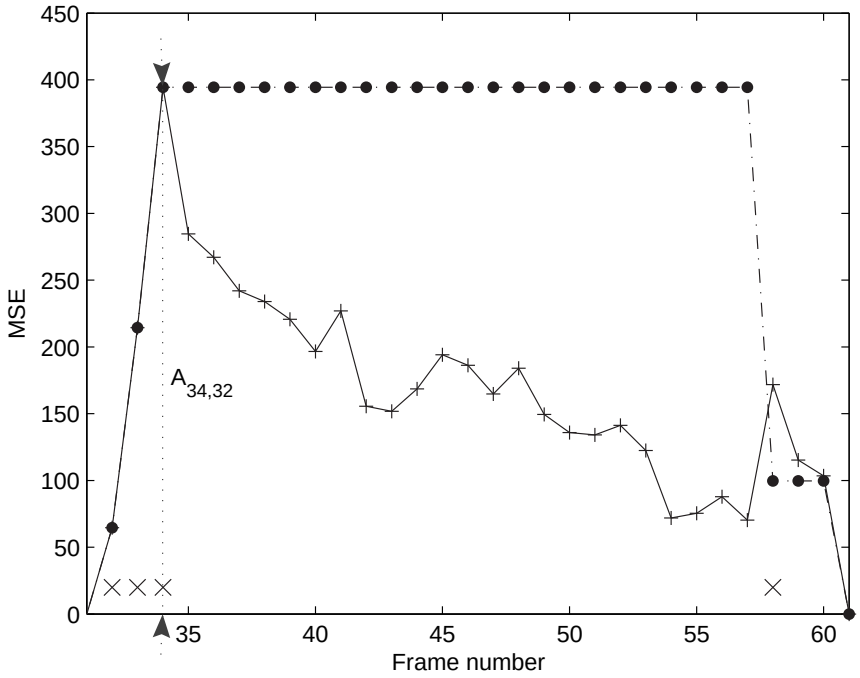

Figure 6.4: Frame per frame distortion estimation using SDA.
—+— Decoder with FC-EC - -●- - SDA ×Loss pattern

6.3.1 Algorithms Validation Test

Consider the video sequence *Foreman* in which each GOP consists of one I-frame followed by 29 P-frames, being the GOP size equal to 30. Assume, for example, that frames 32, 33, 34 and 58 (belonging to the second GOP) are not received by the decoder. Fig. 6.4 exemplifies SDA distortion estimation, assuming that the decoder performs FC-EC (the lost frames are represented by the time markers). The measured total distortion is compared, for the given loss pattern, with the one predicted by the algorithm SDA.
The amplitudes of the SDA steps are evaluated as the pixel-by-pixel difference between the last received frame (also stored in the encoder buffer) and the lost one at the end of the burst, as in (6.1). Therefore, $A^1_{34,32}$, in Fig. 6.4, is the MSD between frames 31 and 34 of the compressed sequence. It is worth noticing that the initial behavior of the SDA ($31 \leq \mathrm{Framenumber} \leq 34$) is superimposed to the actual distortion. Even if the encoder knows that the frames from 35 to 57 are correctly received it does not know the EC technique adopted by the decoder. Therefore, in SDA it is assumed that the distortion does not increase and that the

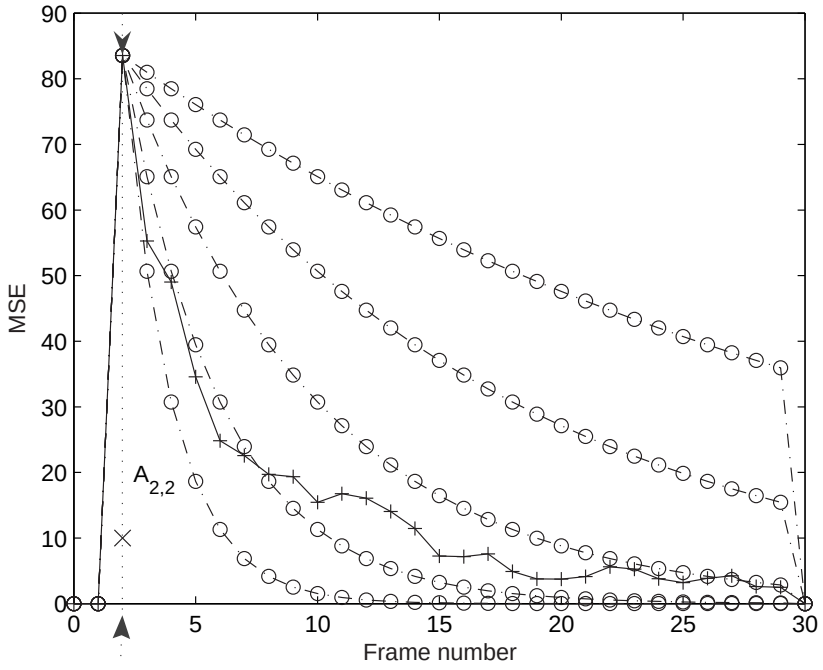

Figure 6.5: Frame per frame distortion estimation using EDA.
—+— Decoder with FC-EC -·o·- EDA $b = (1/2, 1/4, 1/8, 1/16, 1/32)$ from bottom to top ×Loss pattern

decoder is at least able to keep constant the distortion (i.e. step approximation). The estimated distortion envelope changes only if an I frame is received correctly or another P frame is lost. In particular, if an I frame is received correctly the error propagation is stopped and the SDA sets MSD equal to 0 (this is the case of frame 61 in Fig. 6.4, which is the first I frame of the third GOP). If another P frame within a GOP is lost, the procedure is repeated.
Even if SDA assumes a simple step approximation for the distortion envelope, it provides a reasonable estimation for sufficiently spaced single or bursty losses.

The benefits that may be obtained using EDA are illustrated in Fig. 6.5. Here it is assumed that only the second P frame is lost and the decoder performs classical FC-EC scheme. The measured total distortion is compared, for this loss pattern, with the MSD predicted by the EDA. With reference to the same figure, the distortion $D(2)$ is equal to $A^1_{2,2} \simeq 83.54$ and the frame k=3 is correctly received. The EDA algorithm evaluates a distortion $D(3) = A^1_{2,2}e^{-b}$ for the third frame received, a distortion $D(4) = D(3)e^{-b} = A^1_{2,2}e^{-2b}$ for the fourth frame received

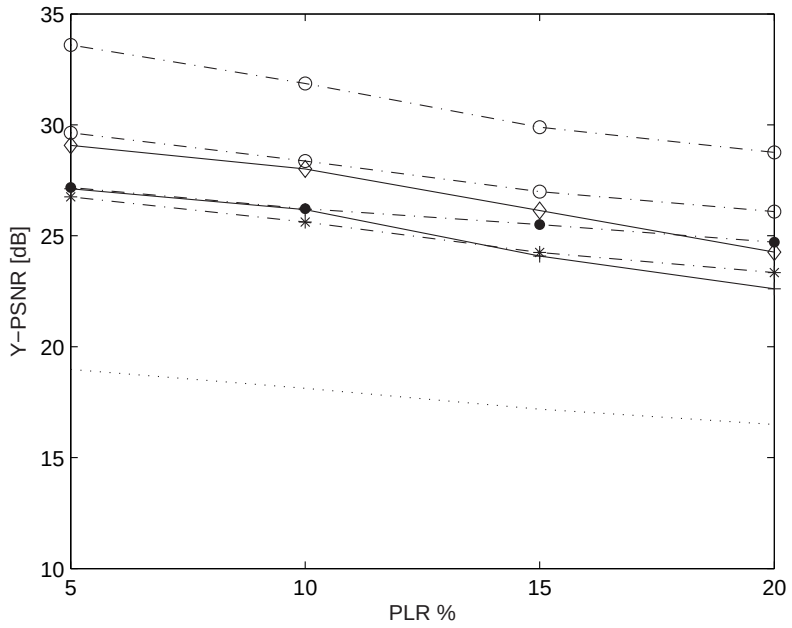

Figure 6.6: Distortion Estimation (GOP=30) for isolated losses.
–·●·– SDA –·○·– EDA $b = 1/2$ (upper), $b = 1/8$ (lower)
–·∗·– ADA $b = 1/8$ ····· Estimation in [92]
—+— Decoder (FC-EC) —◇— Decoder (MC-EC)

and so on.
The EDA may be able to provide a better approximation by a careful choice of the parameter b. The family of curves in Fig. 6.5 for EDA (starting from the bottom) are obtained for $b = 1/2, 1/4, 1/8, 1/16, 1/32$. Notice that, the smaller the b value is, the closer the EDA and the SDA approximation are; for sufficiently large value of b, however, the EDA may over-estimate the actual quality. It is worth noticing that a reasonable approximation of the distortion envelope is obtained for $b = 1/8$. The same b value is used in the experiments described in the sequel also by the ADA algorithm. The value founded for b is the most suitable value for the selected sequence.

6.3.2 Performance for Generic Loss Pattern

In evaluating the performance of the distortion estimation algorithms for generic loss patterns, consider now the whole video sequence *Foreman*. The complete sequence consists of 300 frames. The measured total distortion is compared, for

each loss pattern, with the one predicted by the algorithms SDA, EDA and ADA. The prediction performance is examined for PLR ranging from 5% to 20% while the encoding structure is IPPPP with a GOP size equals to 30. For each packet loss rate, a set of 100 random packet loss patterns is generated and for each loss pattern the corresponding packets are dropped from the packetized bitstream. Then, the frame-per-frame MSE of the luminance component is stored after decoding, and the distortion is estimated through the MSD. Finally, the average Peak to Signal Noise Ratio (PSNR) is evaluated by:

$$\overline{PSNR} = 20 \log 255 + 10 \log N_p \cdot N_f - \log \sum_{i,j} D_{i,j}, \tag{6.6}$$

where $D_{i,j}$ is the real picture MSE as well as the MDS estimated, N_f is the number of frames in the video sequence and N_p identifies a specific loss pattern. Fig. 6.6 shows the performance comparison between the actual and the estimated distortion, obtained applying the proposed algorithms, as a function of the PLR. The accuracy of the methods is evaluated referring to the effective behavior of the decoder employing both the FC-EC and the MC-EC schemes. In order to perform a comparison with the previous proposed techniques, the estimated distortion, obtained using the method described in [92], is also reported in the same figure with the dotted curve.
EDA accuracy is evaluated for $b = 1/2$ and $b = 1/8$, to show the influence of the parameter b on the average estimation performance. For ADA, instead, the value of b is set to $b = 1/8$.

All the proposed techniques provide better predictions of the real expected distortion, with respect to [92]. In fact, the algorithm in [92] does not use all received frames and simply supposes that the decoder rejects all frames after the lost one, applying FC concealment for all frames until a refresh occurs. This hypothesis leads to an underestimation of the received quality. The proposed algorithms, instead, try to model the distortion using all the frames available at the receiver, rejecting only the lost ones. For the entire PLR range (low and medium PLR) both the EDA and the SDA overestimate the actual PSNR, whereas the ADA is shown to be able to closely approximate it.
To investigate further the robustness of the proposed estimation algorithms, a set of simulations is run under very severe channel conditions, which may occur when the channel is in the bad states (i.e. during deep fades). Note that, in this condition, bursts of lost packets may span more than one GOP. Fig. 6.7 shows the performance of the three estimation techniques as a function of the PLR. In the figure it is shown that ADA provides a closer approximation to the actual distortion in the entire PLR interval. This is due to the additional distortion accounted in the estimation process. Thus, it is possible to conclude that ADA is a suitable technique to predict the actual objective distortion, even under severe channel conditions.

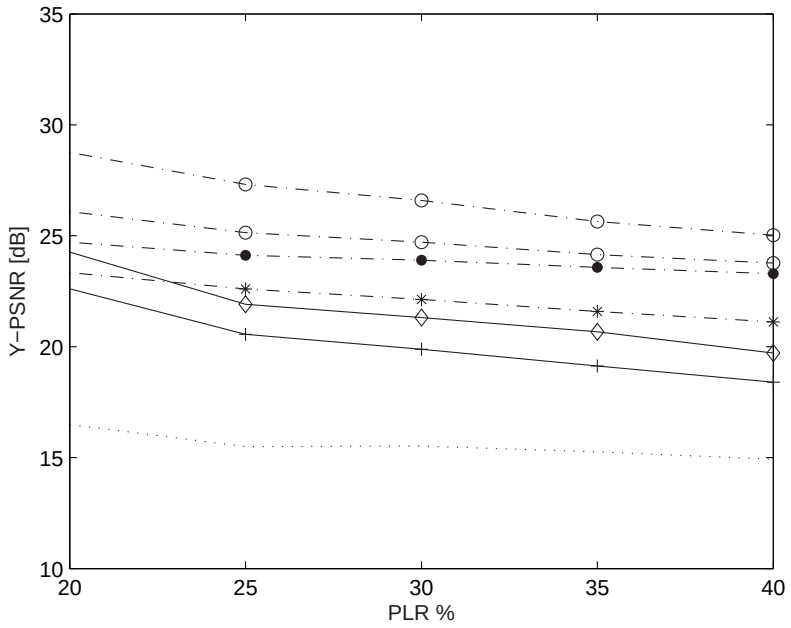

Figure 6.7: Distortion Estimation (GOP=30) for multiple losses.
–·●–·– SDA –·○–·– EDA $b = 1/2$ (upper), $b = 1/8$ (lower)
–·∗–·– ADA $b = 1/8$ ····· Estimation in [92]
—+— Decoder (FC-EC) —◇— Decoder (MC-EC)

6.3.3 Estimation Accuracy of ADA

In this subsection it is presented a detailed analysis of the influence of the reference frame in the ADA algorithm, to assess the generality of the employed technique. When multiple losses appear in a GOP, ADA matches quite well the real distortion. In fact the additional distortion accounted using the second last received frame may accurately model the cross correlation between subsequent lost frames. By changing the reference frame, i.e., by choosing the third last or the fourth last received frame it will be accounted an additional distortion offset. This offset is not constant and depends both on the temporal characteristics of the sequence and on the position of the lost frames. In Fig. 6.8 a frame by frame PSNR comparison between the estimation algorithms is drawn. The same figure shows the actual distortion obtained by using the decoder. The results are shown for the *Foreman* sequence at 15% packet loss rate for 20 frames. In this snapshot, the error occurs in frames 61, 69 and 70.
Fig. 6.9 shows the average performance of the ADA using the last third and the

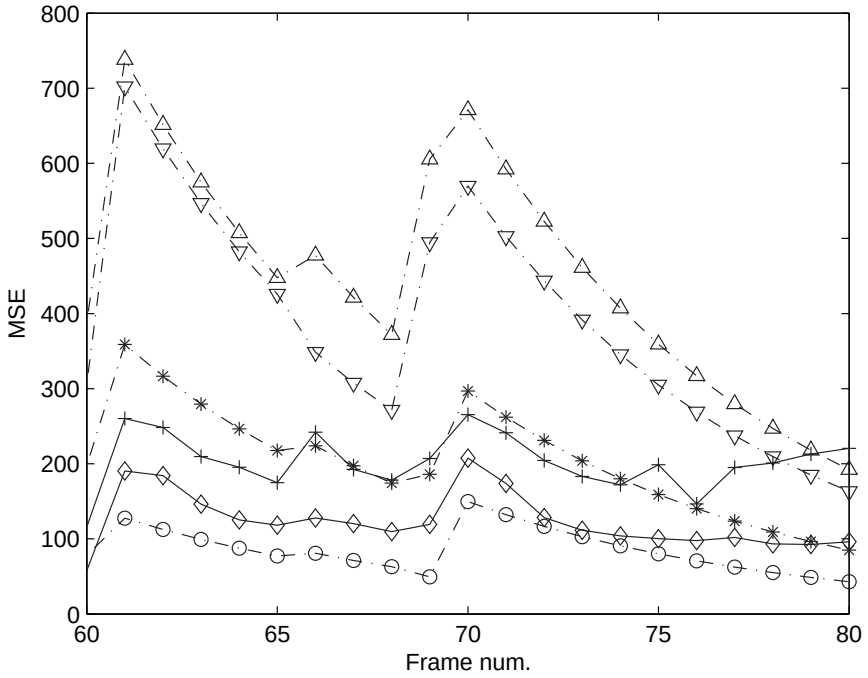

Figure 6.8: Frame per frame distortion comparison between the estimation algorithms and the real distortion for the *Foreman* sequence. The average PLR is 15% and the errors occurs in frames 61, 69 and 70.
EDA $b = 1/8$ ADA using the third reference frame $b = 1/8$
ADA $b = 1/8$ ADA using the fourth reference frame $b = 1/8$
Decoder (FC-EC) Decoder (MC-EC)

fourth frames received as a reference frame for the estimation. The PSNR, evaluated between the decoded sequence and the compressed sequence, does not consider the source compression distortion. The estimated quality applying ADA, using the second last received frame, closely parallels the actual frame level distortion. Using more distant reference frames instead, the approximation worsens, due to the excessive additional distortion.

Summarizing all the above mentioned experiments it is possible to conclude that for isolated losses the best approximation is reached by EDA algorithm. Actually, EDA uses the last received frame to perform the estimation and also the decoder uses the last decoded frame too, to perform the error concealment. For this reason EDA is able to emulate the actual decoder behavior. ADA instead, accounts an additional distortion contribution, equals to the MSD between the

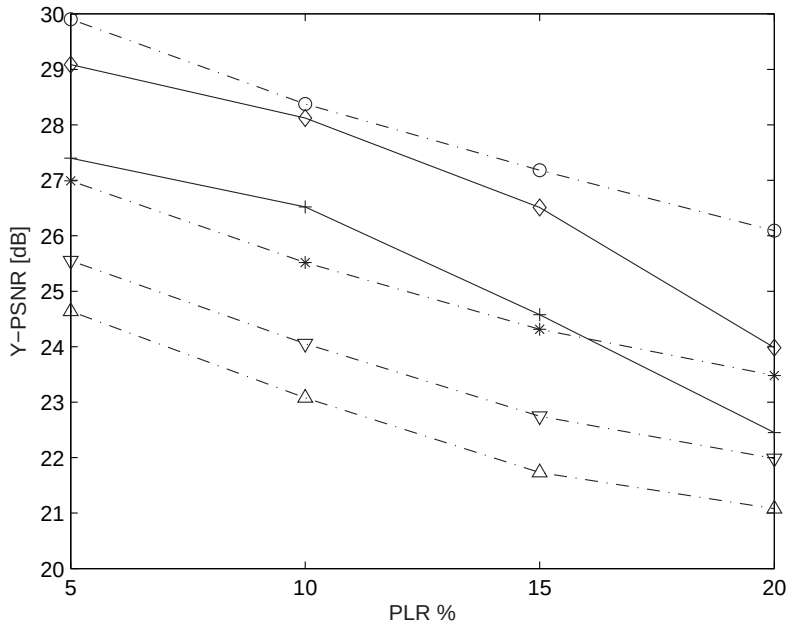

Figure 6.9: Distortion Estimation using the third or the fourth last received frame in ADA for *Foreman* sequence.
EDA $b = 1/8$ ADA using the third reference frame $b = 1/8$
ADA $b = 1/8$ ADA using the fourth reference frame $b = 1/8$
Decoder (FC-EC) Decoder (MC-EC)

second last and the last received frames allowing a good approximation in an high loss environment, as it is shown in Fig. 6.9. Using more distant reference frames instead, is counterproductive, given that an excessive additional distortion is introduced.

6.4 Influence of Encoder Settings

6.4.1 Estimation accuracy changing the input sequence

To assess whether the proposed estimation techniques are able to provide useful results independently from the chosen sequence, a first set of simulations is run by concatenating and repeating QCIF test sequences *Miss America* (100 frames) and *Carphone* (100 frames) in order to obtain a sequence length of 400 frames. The encoding structure imposed at the encoder side is to use only one I frame at

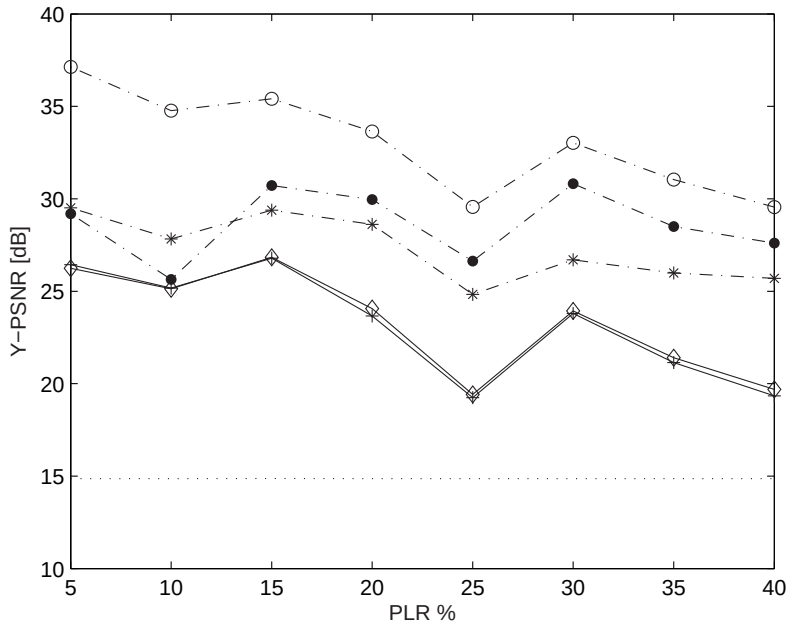

Figure 6.10: Distortion Estimation with a large GOP (only one I frame) for *Miss America-Carphone* sequence.
–•– SDA –○– EDA $b = 1/2$
–∗– ADA $b = 1/8$ ····· Estimation in [92]
—+— Decoder (FC-EC) —◇— Decoder (MC-EC)

the beginning of the sequence. All the other frames are encoded as P frame so GOP size approaches infinity. To improve the robustness of the encoding process, the intra macroblock ratio option available at the encoder side is set to 9, so that a complete refresh occurs every 11 frames. The prediction performance is examined for PLR in the range $5 \div 40\%$. Fig. 6.10 shows the performance of the three estimation techniques as a function of the PLR. As it can be seen from the figure, the estimation behavior does not change: all the algorithms overestimate the real quality. In all the performed tests EDA may be useful to estimate the distortion envelope deriving from isolated losses, while ADA is preferable to approximate the distortion deriving from multiple bursts. It is worth noticing that, without setting the GOP size, a dramatic visual error propagation occurs during the play of the sequence. In fact multiple loss of packets cause the loss of synchronization between encoder and decoder that may be recovered only with a reception of an I frame or a scene change. Only the reception of an I frame, stopping the error propagation, can recover the synchronization between the encoder and the decoder. The intra

Sequence	Number of frames	Resolution	PSNR [dB]
Miss America	200	QCIF	36.42
Carphone	200	QCIF	35.11
Silent	400	QCIF	37.87
Salesman	300	QCIF	36.02
News	600	CIF	36.53

Table 6.1: Sequence used for validation.

macroblocks, inserted in the P frames, mitigate the error propagation but many artifacts still appear in the decoded picture. Therefore, the PSNR evaluation may be misleading, given that the subjective evaluation reveals many mistakes.

A more extensive second set of tests is performed to further validate the distortion estimation models. The sequences, summarized in Table 6.1, are coded with the Baseline profile, frame mode IPPPP and Quantization Parameter (QP) equal to 28. The sequences are coded using constant quality compression, so the total frame per frame quality depends only on the channel induced distortion. The packet loss patterns 3%, 5%, 10% and 20% are employed as candidates for the tests. The lossy bitstream is decoded applying only FC-EC at the decoder and the distortion is estimated using only ADA with a proper choice of the parameter b. In particular, for every sequence reported in Table 6.1, the parameter b is determined during the initialization phase, in which a single channel loss is generated. Table 6.2 summarizes the PSNR differences between the real average distortion and the estimated one, showing a good agreement, both for sequences with limited or significant motion.

Sequence	PLR=5%	PLR=10%	PLR=20%
Miss America	0.11	0.26	0.08
Carphone	1.02	-0.05	-1.27
Silent	0.86	0.25	-0.30
Salesman	0.32	-0.12	-0.81
News	0.03	-0.43	-1.81

Table 6.2: PSNR difference between the real distortion and the estimated one using ADA for different packet loss rates.

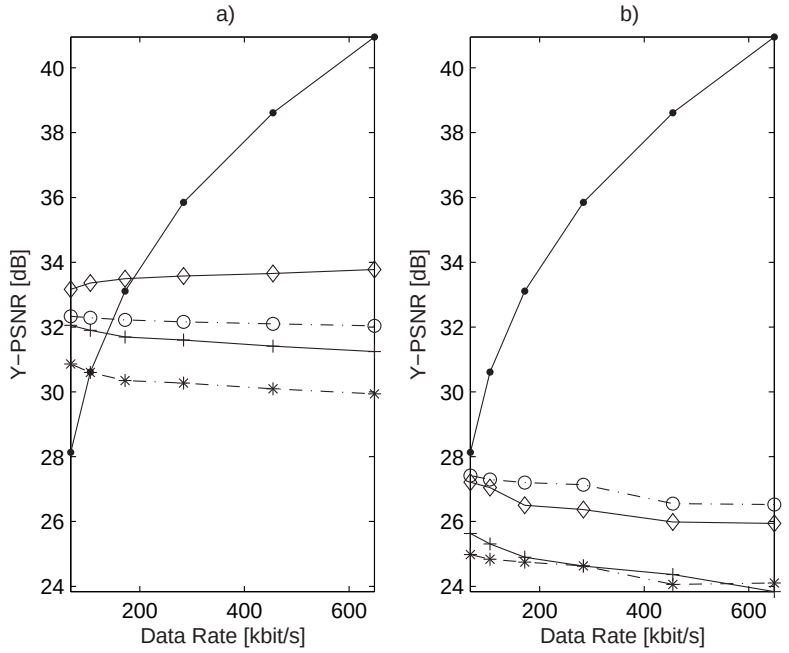

Figure 6.11: Estimation algorithms performance vs. data-rate for the *Foreman* sequence. a) PLR 3%. b) PLR 15%.
EDA $b = 1/8$ ADA $b = 1/8$
Decoder (FC-EC) Decoder (MC-EC)
Source compression distortion

6.4.2 Estimation accuracy changing the source compression

A further validation test can be performed by changing the source data-rate compression. The sequence chosen to perform the experiment is the *Foreman* QCIF sequence compressed using different quantization values. The GOP is equal to 30 and the PLR simulating the channel losses are 3% and 15%.

Fig. 6.11 resumes the results of the experiment. In particular, for Fig. 6.11 a), at higher rates, the total source-channel distortion is due to the loss of the channel because the source compression distortion is negligible. At lower source data rates instead, the quality of the decoded sequence depends mainly from the source distortion and the channel induced distortion is negligible, this is due to the fact that at lower rates the encoding process removes many sequence details.

While in Fig. 6.11 b), the distortion primarily depends on the contribution of the channel. The approximation using the proposed algorithms matches with the real distortion. For PLR equal to 15% the FC-EC curve is well approximated by

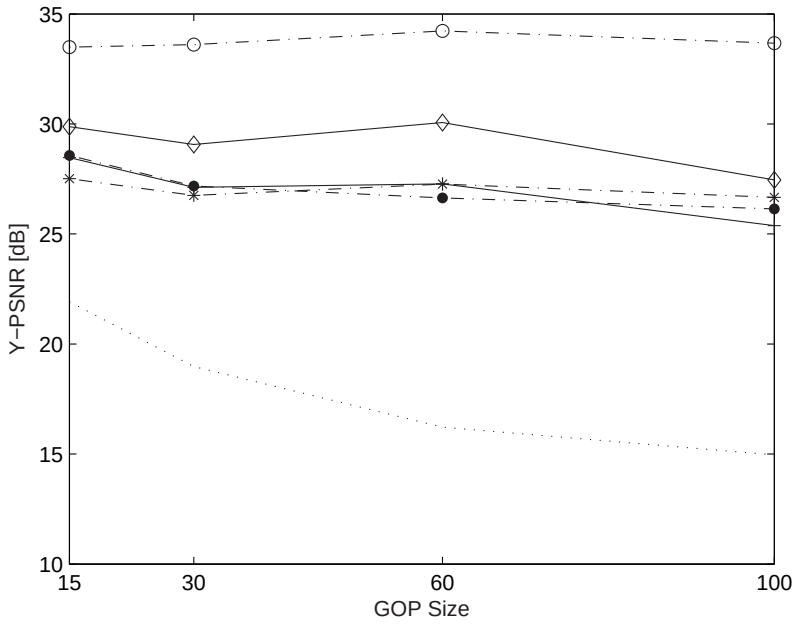

Figure 6.12: Estimation performance vs GOP size (PLR=5%).
SDA EDA $b = 1/2$
ADA $b = 1/8$ Estimation in [92]
Decoder (FC-EC) Decoder (MC-EC)

the ADA algorithm while for PLR equal to 3% the best approximation is reached by the EDA algorithm. From the same figure it follows that the compression rate does not affect the approximation accuracy.

6.4.3 Influence of the GOP size

The performance of the proposed estimation techniques is also evaluated as a function of the GOP size, as it is shown in Figs. 6.12 and 6.13, assuming that PLR is equal to 5% and 40%, respectively. Observe that the performance of all the three estimation algorithms and the actual decoding have a very weak dependence on the GOP size. The decoder performance, whereas, may be influenced by the intra macroblock ratio set at the encoder side. Actually, the intra macroblocks inserted in each P frame are able to mitigate the error propagation effects, improving the visual quality. For this reason it may be desirable to enable adaptive intra refresh methods at the encoder side in order to mitigate the error propagation. This fact may be taken into account in both the EDA and the ADA, by a proper selection of

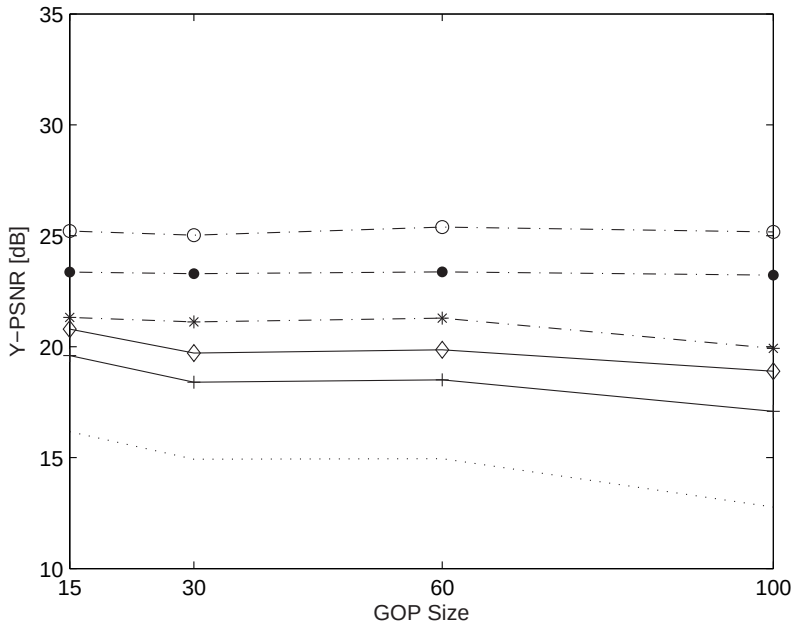

Figure 6.13: Estimation performance vs GOP size (PLR=40%).
SDA EDA $b = 1/2$
ADA $b = 1/8$ Estimation in [92]
Decoder (FC-EC) Decoder (MC-EC)

the parameter b.

6.4.4 Influence of the parameter b

EDA and ADA depend primarily from the choice of the parameter b; a small variation of b provides a noticeable variation of the estimated distortion for all the algorithms under investigation. The results of a set of tests, performed to evaluate the influence of b value on ADA and EDA performance, are summarized in Table 6.3 that compares the PSNR difference between the real and the estimated distortion for several test sequences and for different choice of b. Observe that EDA usually underestimates the actual distortion, while ADA may be capable of a correct estimation, provided that a suitable, not too small b value is chosen. The transmitter may determine the b value during the initialization phase, by measuring the actual distortion envelope deriving from a test loss event.

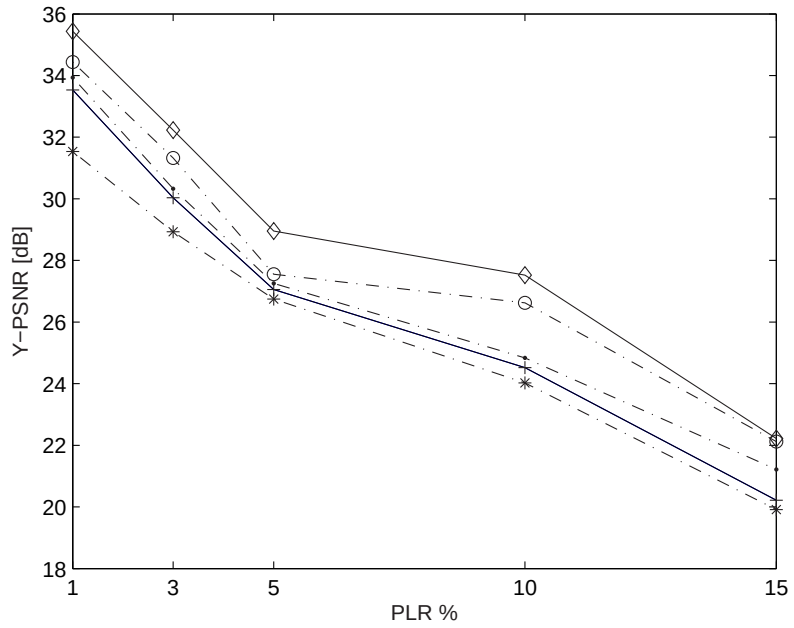

Figure 6.14: Distortion Estimation with GOP=30 and IBPBP structure vs PLR.
–·●·– SDA –·○·– EDA $b = 1/6$ –·∗·– ADA $b = 1/6$
—+— Decoder (FC-EC) —◇— Decoder (MC-EC)

6.4.5 Distortion estimation with B-frame

Estimation accuracy does not depend on frame types, actually, when a B-frame is lost, there is no error propagation since successive frames do not use the B-frame as a reference frame to perform the decoding of the next frames. Therefore, the proposed techniques can be used to model the channel distortion in successive P-frames, even if there are B-frames in between. In particular, when a loss of a B frame occurs, the estimation algorithm simply do not consider the error propagation effect setting $b = 0$. Fig. 6.14 shows the performance comparison between the actual and the estimated distortion as a function of the PLR for both the FC-EC and the MC-EC schemes.

6.4.6 Distortion estimation changing the sequence resolution

In the previous validation tests it is assumed that one frame fits in a single packet. However, for high resolution sequences, such as S-CIF used for instance in HDTV, one frame may span over multiple packets. This is made feasible both by slices

Sequence	**EDA** $b = 1/2$	**EDA** $b = 1/32$	**ADA** $b = 1/2$	**ADA** $b = 1/32$
Miss America	-5.11	-2.34	-1.21	3.42
Carphone	-6.25	-3.45	-0.21	2.16
Silent	-4.63	-1.14	-1.21	5.09
Salesman	-3.32	-2.56	-1.21	4.13
News	-5.03	-2.13	-1.21	3.40

Table 6.3: PSNR difference between the real distortion and the estimated one using EDA or ADA with different values of b.

Figure 6.15: S-CIF picture obtained mixing CIF sequence. Each picture has 4 slice groups.

grouping and Network Adaptation Layer Units (NALUs). As explained in [112], one slice is a single video data unit and multiple slices can be encapsulated in a NALU. Moreover, a single frame may be transmitted using more than one NALU. It is worth noticing that DEAs may be used to estimate the distortion also in this condition, because of the linear property of the MSD.

The accuracy evaluation test can be summarized as follows. A S-CIF sequence is obtained mixing the first 200 frames of the CIF sequences *Foreman*, *Miss America*, *Carphone* and *News*. A single S-CIF frame is created using the four CIF frames of the sequences. Moreover each MB is assigned to a slice group using a macroblock allocation map file that groups together all MBs of a CIF picture as illustrated in Fig. 6.15. The resulting sequence is encoded using an IPPPP structure with a GOP size equal to 30. A set $N_p = 100$ random packet loss patterns with

	PLR=5%	**PLR=10%**	**PLR=15%**
$\mathrm{PSNR_r} - \mathrm{PSNR_{SDA}}$	-1.21	-1.03	0.92
$\mathrm{PSNR_r} - \mathrm{PSNR_{EDA}}$	0.15	1.58	2.25
$\mathrm{PSNR_r} - \mathrm{PSNR_{ADA}}$	-2.8	-1.27	-0.06

Table 6.4: PSNR difference between the real distortion and the estimated one using DEAs for different packet loss rates.

average PLR of 5%, 10% and 15% is generated and used as test case. Average values of the resulting MSE using FC-EC and the MSD using DEAs are evaluated. Table 6.4 summarizes the results showing the difference between the real PSNR ($PSNR_r$) and the one estimated using DEAs.
It may be concluded that DEAs are able to provide useful results independently from the chosen sequence, showing a good agreement, both for sequences with limited or significant motion.

6.5 Conclusion

In conclusion, DEAs are able to provide useful results independently from the chosen sequence, showing a good agreement, both for sequences with limited or significant motion. Also the sequence resolution i.e. QCIF, CIF or S-CIF does not affect the estimation accuracy. Moreover, large GOP sizes will results in a severe visual error propagation during the sequence reproduction at the decoder side. However, DEAs are able to model the true distortion envelope also in this condition. The frame per frame accuracy, however, depends both on the chosen estimation algorithm and on the test conditions. For a large GOP size and high loss rates, only the ADA is able to reproduce the real distortion envelope accurately. Finally, estimation accuracy does not depend on frame types: the proposed techniques, in fact, may be used also to model the channel distortion in successive P-frames, even if there are B-frames in between.

Chapter 7

Applications of DEA

In the present chapter are presented and evaluated several application scenarios, where the developed algorithms may be used proactively by some agents distributed on the networks to enhance the user experience. The application scenarios are either simulated or evaluated developing real test beds. In this way it is possible to quantify and measure the improvements that a real implementation in a prototype may give with respect to traditional delivery policies.

7.1 Quality evaluation

By using the proposed algorithms to estimate the end user distortion, reconstructed video quality may be directly evaluated from the packet loss pattern and the output sequence at the encoder side, with no need to perform decoding. The packet loss pattern can be obtained, for example, running transmission tests with network simulators such as *ns2* [113] or *Opnet* [114], or running real wireless transmission tests using wireless cards.

The same approach may be used in actual systems in which the decoding is not feasible because, for example, the decoder is not able to manage an high loss rate (in the sense that it breaks down, stopping the decoding operations). Moreover, in high loss environments the decoder output sequence and the reference sequence have a different number of frames due to the skipping operations at the decoder. By consequence, a correct quality measurement requires the implementation of the video display buffer with its refresh interval, dictated by the frame rate. On the contrary, using the proposed algorithm, a video distortion may be evaluated simply using the packet loss pattern and the undistorted reconstructed video sequence.

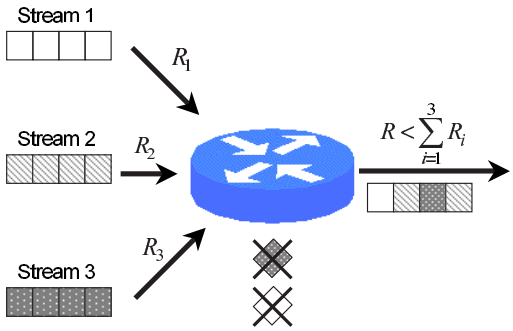

Figure 7.1: Bandwidth Adaptation trough packet dropping.

7.2 Bandwidth Adaptation using DEAs

The proposed distortion algorithms may be used in to perform efficiently bandwidth adaptation. The scenario is commonly encountered in the Internet and it occurs whenever the data rate on the incoming link at a network node exceeds the data rate on the outgoing link. During network congestion transient periods router queues overflow, so that it is required a bandwidth adaptation. The scenario under consideration is illustrated in Fig. 7.1 where the incoming traffic at the node consists of multiple video streams multiplexed by the router on a single outgoing link. Each RTP/UDP video packet frame $P_{i,j}$, where i refers to the packet number while j specifies the flow (i.e., $j = 1, 2, 3$), reserves some bytes to store the value of $A^1_{i,i}$ and $A^2_{i,i}$ that take into account the distortion impact associated to every packet. The router, using the distortion value of each packet, is able to employ a transmission scheduling strategy on the outgoing link (i.e., the scheduler has a mean to select and to schedule the transmission of a new packet on the basis of his distortion impact, with a transmission policy granting a privilege to the packets with the higher distortion). In particular, the network node is interested in maximizing the overall quality over all input streams without exceeding the fixed bandwidth on the outgoing link. Note that, in this environment, the optimal transmission scheduling for the incoming packets is computed directly by the network node that performs some cross layer information exchange because it reads some application layer reserved bytes in the payload of every packet. In other words, by using the rate-distortion information associated to each incoming packet, the node decides which one will be dropped out from every stream, due to an insufficient bandwidth on the outgoing link, and which ones will be forwarded. Moreover, the proposed algorithms may be used to design a priority queue policy, based on application quality requirements, leading to a smooth quality degradation [115].

7.3 Wireless video scheduling using DEAs

Another application may provide high quality video delivery in Wireless LANs (WLAN). Assume that there are multiple sources of video traffic communicating over a shared wireless medium. The communication is performed via an Access Point (AP) that supports the WLAN environment. Using the proposed algorithms, each source can independently optimize the transmission schedule for its own packets, so that the video quality of the stream sent over the shared channel is maximized. It is assumed that each source node is able to capture and encode its own stream, and that each node is able to communicate with an AP through a wireless card. Each node runs a channel access scheme to share the resources. In particular, the access scheme Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is employed for sharing the wireless medium among the multiple users. With CSMA/CA, the users have to contend first for using the communication channel before their actual transmissions. CSMA/CA is adopted by most IEEE 802.11 WLAN standards.
Moreover, each user has a knowledge of the packet loss pattern by means of the DATA-ACK packet handshake available at the MAC layer only for the point-to-point communications. Using the packet loss pattern a user may estimate the distortion by applying DEAs. Using this information, each user may perform an optimal scheduling decision to minimize the distortion at the receiver. For example, in case of unsuccessful access attempt, a packet is marked as available for retransmission and using the loss information, the end-user quality is also estimated. When a new transmission opportunity occurs, the transmitter may choose between the transmission of a new packet or the retransmission of a previously stored packet. Therefore, by running the proposed estimation algorithm and by adopting a suitable threshold mechanism, the transmitter is able to select the packet with the highest impact on the perceived distortion. Note, however, that a user who may have many packets with a high impact on the perceived distortion, is not allowed to transmit continuously because the protocol rules must guarantee the fairness to all users.

7.4 Implementation of DEAs in a real test-bed

This section describes the tools developed to design and evaluate the performance of a complete video streaming framework that uses the distortion estimated by DEAs as a simple reactive method to prioritize video packets depending on the distortion impact. In the following it is provided a detailed description of the system setup and of the main tools developed. Then, a real campaign measurement is conducted to evaluate the overall system performance in terms of enhancements

of the user experience during the video streaming sessions.

7.4.1 System Tools

The modification added to the video encoder allows the generation of another output parameter: a resume file containing the description of the NALUs together with the associated distortion impact estimated using EDA. The total distortion produced by the loss of each NALU is evaluated integrating the estimated distortion in the actual GOP. The packetization rules adopted in the system setup, are obtained fitting a single NALU into the payload of a RTP packet, while the RTP header values are filled as defined in the RTP specification [32].
The distortion impact of each packet is attached in the RTP payload to allow other network nodes to have access to this information. The distortion impact associated to each packet is stored in a compressed manner only using a single byte to minimize overhead in the payload of the packet. The quantized distortion D_q takes values from 0 to 255, where lower values indicate negligible distortion impact, while higher values indicate large distortion. If D is the distortion produced by the loss of a single packet, the quantized value is simply obtained using the following expression:

$$D_q = \lfloor \frac{D * 255}{M} \rfloor, \tag{7.1}$$

where M represents the maximum distortion value obtained from an offline analysis of the encoded test sequences.

A simple transmission tool, that extracts both video slice packets and the associated distortion, has been developed. In particular, the tool establishes an RTP connection with a specific IP address of one destination and then sends the H.264 video packets to a specified UDP port. The transmitter sends each packet according to timing information by which the stream has been encoded. If the frame rate is 30fps, all packets in one frame are sent in 33,3ms. The distortion attached in each video packet, instead, measures the actual importance of each packet and may be used directly by other network nodes without performing additional operations such as decoding. In particular, an intermediate node may simply capture the packet, extract the first byte of the payload and perform the optimization. Observe that, using this approach, an intermediate node is able to perform some cross-layer information exchange using a realistic estimation of the perceived video quality.

In the test, the main purpose is the end user quality evaluation and for this reason the transmitter and the receiver dump all packets by means of standard capture tools such as *tcpdump* [116] and *wireshark* [117] by which it is possible to store packets allowing us to perform both offline decoding and quality evaluation. The packet trace files of either the transmitter and the receiver are fit to another developed tool that reconstructs the transmitted video as it is seen by the decoder,

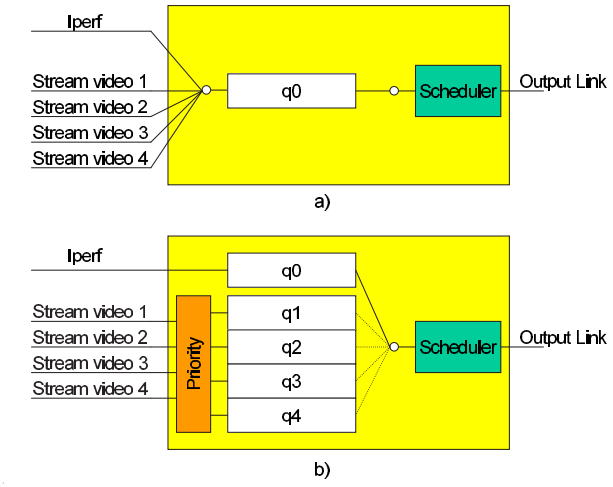

Figure 7.2: Prioritization strategies at the router node. a) Single queue. b) Multiple queues with distortion prioritization.

comparing traces and simply discarding dropped or excessively delayed packets from the original bitstream. The received packets are then filled to the decoder in order to produce the YUV output raw file. Finally, the objective video quality between the undistorted and the distorted sequence is evaluated using the *psnr* tool.

7.4.2 Streaming System Setup

This section describes the complete streaming system implementation, discussing the related issues regarding the developed software, the computational complexity and the adopted hardware. In the presented setup EDA is selected to measure the actual distortion importance because the bottleneck link experiences a moderate loss (as detailed in the following paragraphs). The estimated distortion is used to provide a real implementation of a selective packet dropping strategy in a router using consumer hardware.

In the proposed scenario the data rate on the incoming link at a network node exceeds the data rate on the outgoing link so that it is required the bandwidth adaptation. The router multiplexes all streams on the outgoing queue. In particular, one transmitter node sends both RTP video flows and additional competing traffic using the *iperf* tool [118]. The router multiplexes all received traffic and forwards it on the outgoing link to reach the destination. The outgoing link is a wireless *ad-hoc* network with a PHY link rate fixed at 1Mbps. The maximum UDP throughput sustainable on the wireless link, measured with *iperf*, is around

Name	Length	Bit rate	Y-PSNR(dB)
Foreman	300	157kbps	36.35
Carphone	300	197kbps	37.87
Miss America	300	65kbps	35.70
Silent	300	75kbps	36.28

Table 7.1: Main characteristics of the four test video streams.

850Kbps.
Two different prioritization strategies are implemented in the router as illustrated in Fig 7.2. In the fist all the incoming traffic are multiplexed on a single outgoing queue while, in the second, the RTP flows are forwarded on four different queues on the basis of their distortion impact, according to a transmission policy granting a privilege to the packets with the higher distortion values. In other words, by using the distortion information associated to each video flow the router assigns each packet on a different priority queue. The operating system adopted in the router is based on Linux Ubuntu and both the routing and scheduling capabilities are offered by the popular Click Modular Router [119] software. Click is used because it enables the design of a modular system with several functionalities implemented in different components, called *Elements*. In the developed architecture, Click captures incoming packets from the Ethernet network interface, classifies them, and passes them to the corresponding queue. A new Click Element has been developed to extract the distortion impact and to differentiate the incoming traffic by the port number. In this way it is possible to differentiate competing traffic from video stream flows. In particular, the iperf UDP traffic is transmitted on port 22000 while the video flows uses ports from 1000 to 1004. The distortion impact extracted from the video packets is used to map video packets on the router queues from q_1 to q_4. A *Classifier* Click Element is used to compare the distortion values of the video flows with a predefined thresholds evaluated by offline simulations. The *Classifier* assigns a priority to each packet as follow: queue q_0 gives the higher priority to the iperf traffic while video packets are assigned to queues from q_1 to q_4. The three thresholds used in the test to map video packets on the queues are: 64, 128 and 192. In this way packets with quantized distortion ranging from 0 to 63 are assigned to the queue q_4 and so on.

The queue status is evaluated at regular intervals by the scheduler to check if there is any packet to transmit. When a new packet is detected, the scheduler fits the packet on the wireless card buffer. The wireless network is managed using the Multiband Atheros Driver WiFi (MADWIFI) driver [120].

The testing scenario consists on collecting the trace statistics for offline analysis during the transmission of the RTP video flows and the additional competing

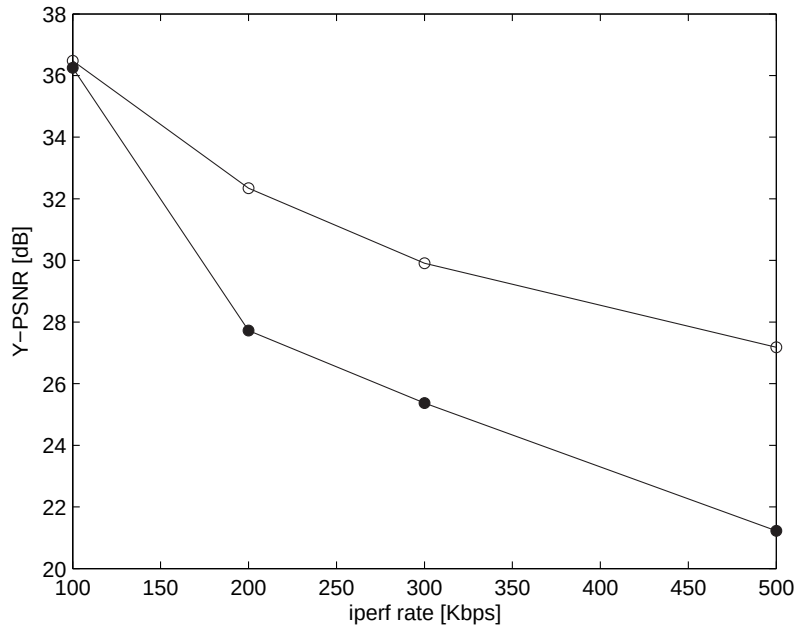

Figure 7.3: Y-PSNR (dB) versus competing traffic rate.
—●— Single queue.—○— Multiple queue.

traffic. Table 7.1 summarizes the main characteristics of the four video streams encoded with the H.264 video codec. Several simulations are run using different congestion levels by changing the rate of the competing iperf traffic. In particular the UDP rate is changed from 100 Kbps to 500 Kbps allowing the scheduler to discard some packets using the distortion information.

7.5 Experimental Results

Fig 7.3 shows the average Y-PSNR (dB) of the four sequences resulting from the two prioritization schemes as a function of the competing traffic rate (Kbps). It is possible to notice that the multiple queues with distortion prioritization scheme outperforms the single queue scheme with a significant margin over the whole range of the competing traffic rate. This confirms that the proposed scheme is able to exploit the distortion information, allowing the scheduler to prioritize packets with higher distortion impact. From Fig 7.3, it may be derived that the performance gain of the prioritized scheme becomes more significant when the competing traffic rate increases. It is worth to notice that, with a competing traffic rate of

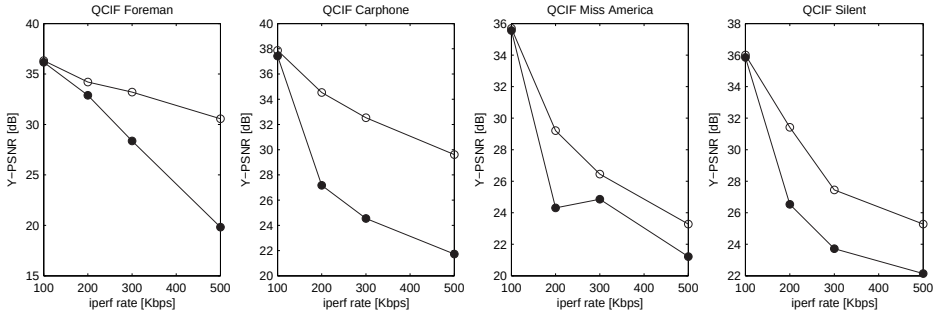

Figure 7.4: Y-PSNR (dB) versus competing traffic rate per sequence (Kbps) for *Foreman*, *Carphone*, *Miss America*, and *Silent*.
—●— Single queue.—○— Multiple queues.

500 Kbps, the performance improvement using the prioritized scheme is around 6 dB in PSNR. Finally, the received quality of individual sequences as a function of the competing traffic rate for both the prioritized scheme and the single queue scheme is shown in Fig 7.4. From these results, it may be derived that the proposed method makes comparable the quality of the different video streams, for the examined background traffic conditions. More precisely, from the figure it may be observed that when the rate of the competing traffic increases, the prioritized scheme assigns more resources to *Foreman* and *Carphone* than to *Miss America* and *Silent*. In fact, *Foreman* and *Carphone* have a more significant impact on the overall quality because of the higher distortion values of their packets. As the competing traffic rate decreases, the distortion prioritized scheme sends larger percentage of *Miss America* and *Silent* packets, as shown in Fig 7.4. At lower competing traffic rates there is enough rate for these two sequences, so that the scheduler can transmit also packets on lower priority queues.

7.6 DEAs in a Multi-hop Environment

This section describes another application scenario that may take advantage of DEAs to improve the received video quality. In this experiment a streaming server sends out a single stream on a four hop communication path. Every intermediate node receives the video content and retransmit it to the next hop. The intermediate hops elegantly discard some frames to adapt the output rate to the bandwidth while alleviating the degradation in the decoded video quality in a distortion-constrained way. Since the single hops in a multi-hop path are arranged in a descendent order by the available bandwidth, the bandwidth to receive data is larger than one used to send packets for every intermediate node. In order to avoid sending buffer

Figure 7.5: Linear Multi-hop path with 4 hops and relative bandwidth.

overflow and the transmission of late video packets, intermediate nodes must drop some frames to shape the receiving bit-rate to the sending bandwidth. The frames received by every hop are selectively dropped according to their contribution to the decoding quality.

Three different frame priority schemes are adopted in this simulation. In the first the frame priority is simply determined based on the frame type. This mechanism is called Frame Drop Priority (FDP) scheme. According to the contribution to the decoding quality, the priorities of the frames decrease by the type I, P and B. In the second and third scheme the distortion impact of each packet is considered to perform the scheduling decision. In particular, in the second scheme each node has knowledge of the distortion impact of every packet simply extracting the distortion information contained in every frame. At each transmission opportunity an hop scan all the received packets discarding excessive delayed ones. Then the packet with the higher distortion is selected and scheduled for the next transmission. This simple scheme is called Distortion Drop Priority (DDP). In the third prioritization scheme the sender has exact knowledge of the loss pattern of each hop and is able to communicate the new packet distortion impact accordingly to every hop using some signaling packets. In this way every intermediate node has the knowledge of the effective distortion produced by the loss of a new packet. During the first hop transmission some packets may be discarded due to bandwidth limitations. By consequence, the single packet distortion needs to account of the previous losses. The sender communicates the new distortion impact to allow intermediate nodes to use a more useful distortion values. This prioritization scheme is called Advanced Distortion Drop Priority (ADDP).

In the experiment the standard sequence *Foreman* with QCIF resolution is encoded using the H.264 encoder at 30 fps with average bit rate of 326 Kbps. The complete path is shown in Fig. 7.5 and the bandwidth of the links are fixed for convenience in a descending order at 300 Kbps, 280 Kbps, 260 Kbps and 240 Kbps. Fig. 7.6 shows the total estimated MSE that corresponds to the loss of each frame in every intermediate node. The bottom curve for example, represents the distortion impact of every frame when a single loss occurs. The other curves represent the distortion obtained at each hop considering that some packets have

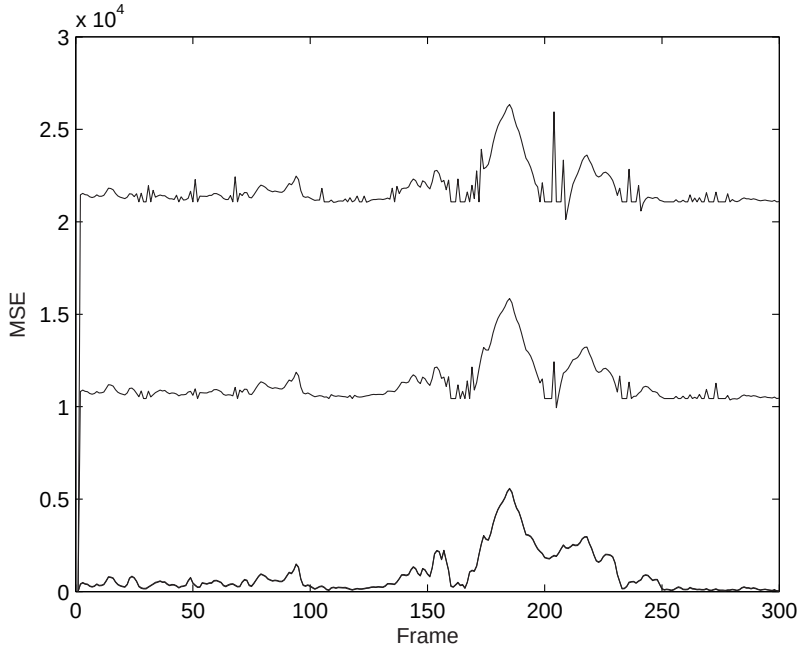

Figure 7.6: Total distortion experienced at every intermediate node as a function of the lost packet.

been discarded during previous transmissions. Notice that the curves seem to have a constant offset that corresponds to the total distortion in each intermediate hop. Fig. 7.7 resumes the channel distortion measured in PSNR and obtained decoding the received packets at each intermediate node using the three priority schemes. PSNR are obtained comparing the decoded sequence with the error free sequence so the PSNR does not account the source compression distortion. By the same figure it follows that the performance of the DDP scheme outperforms the ones of FDP. In fact, FDP is not able to differentiate between two P frames while DDP captures the real distortion impact produced by the loss of every packet. Moreover, ADDP has performance quite similar to DDP and this is due to the fact that the distortion values used by ADDP and DDP has a similar envelope (that differs only from a constant offset). Only in the third and fourth hop a little margin may appear. Notice that using distortion information in a multi-hop communication environment may provide significant benefits with respect to simple FDP scheme.

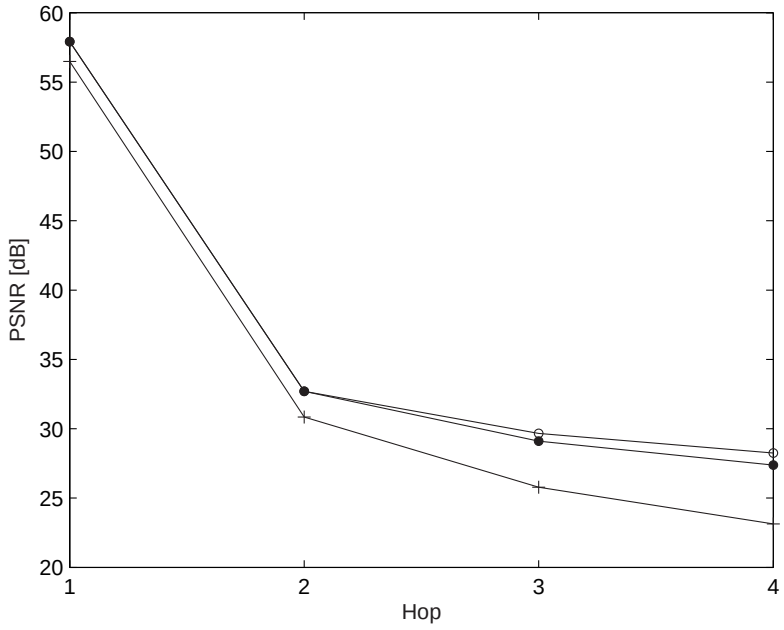

Figure 7.7: PSNR in the nodes of the multi-hop path.
—○— ADDP —●— DDP —+— FDP

Chapter 8

Conclusion

8.1 Summary of Accomplished Research

This thesis has considered some aspects of multimedia communication over unreliable and resource-constrained IP-based packet-switched networks. The focus and objectives of the work are related to estimating the distortion deriving from streaming a video content over an error prone network. Towards this end, the first contribution of the thesis has been focused on estimating the video quality in a video communication system as it is seen by the receiver. Specifically, low-complexity methods for estimating the envelope of the MSE in a video have been developed, and many validation tests have been performed to measure the capability of such algorithms. In detail, the estimation algorithms are based on a low-complexity method for estimating the block-edge impairments in video. The proposed methods may be applied at the encoding side of a video communication system and the predicted distortion may be transmitted with the video content. Validation results show that the employed algorithms are able to estimate in a better way the distortion envelope in many different conditions such as changing the sequence and the main encoder/decoder settings.
The next valuable contribution of the thesis considers how the performance of streaming media application could be improved using the employed algorithms. Different application scenarios are proposed and evaluated both through simulation and real campaign measurements.
The last part of the thesis is an application and system study, focusing on how the performance of an adaptive video streaming can be improved and enhanced by utilizing the information provided by the distortion estimation algorithms. In particular it is developed and evaluated a new rate-adaptation algorithm based on the distortion impact of every packet.
Moreover, an adaptive video-on-demand solution based on H.264/AVC scalable

video coding is presented, which is shown to enable efficient video distribution over IEEE 802.11 wireless networks. The proposed rate adaptation is based on a simple idea and allows streaming media applications to improve their own performance, while allowing effectively traffic prioritization based on application metric. A prototype of the video streaming system has been developed and presented, and the overall system performance is evaluated using a real 802.11 network infrastructure. Moreover, the performance enhancements in respect to the traditional transmission policies over a multi hop wireless network remarks the gain obtained with the employed algorithms.

8.2 Future Work

The development of a new techniques able to estimate video quality remain an important research goal, and will be essential in developing fully automated quality adaptation systems for multimedia services and applications. The proposed algorithms quantifies the effect of network impairments, which are more relevant for MPEG-4 and H.264/AVC codecs. With the recent standardization of the scalable version of the H.264 named SVC, the proposed algorithms may be extended and improved to predict better the distortion derived from the loss on the network. In particular for H.264/SVC, block-edge impairments are no longer the dominant compression-related distortion, and methods for determining the impact of blurriness and ringing will become more important. Further, evaluating the impact of transmission related distortions, such as packet loss for these new compression schemes in an automated manner, is a difficult problem which is not yet fully solved. Finally, both AVC and SVC support temporal scalability, that is to say, methods for automatically evaluating the perceived impact of varying the frame rate will be essential. With the adoption of SVC, this may also apply to other types of video adaptation techniques.
The development and validation of new distortion estimation algorithms require formal subjective testing in addition to comparison with state-of-the-art objective video quality models. In order to evaluate and predict the perceptual effects of compression, transmission and adaptation-related distortions, highly realistic delivery scenarios and appropriate processed test material must be used as a starting point for subjective testing and distortion algorithm validation. Therefore, continued efforts are required to develop new distortion estimation algorithms, and more realistic simulation scenario and real testbeds. Only in this way our studies can better reflect real-life conditions.
With regard to the adaptive streaming solution presented in this thesis, future works could include studying other scenarios, and further investigating on the impact of varying system parameters. In addition, it would be interesting to ex-

plore the effects that dynamic resource allocation mechanisms have on the performance of the streaming video system. For example, the measure of the benefits obtained in a real multi hop communication path will be explored and the benefits obtained using other prioritization techniques available at the lower layer such as the QoS prioritization mechanism available in IEEE 802.11e. In particular, the development of a new mechanism able to map each packet in one of the 4 queues available in the 802.11e standard that considers the distortion associated to each packet. In this way a method that adaptively assigns each packet to one queue basing on distortion information.

Acknowledgements

Looking back on my graduate study at the University of Trieste, I am truly grateful to many people who helped me become a mature and confident person, ready for challenges in my future career.
First and foremost, special thanks go to my advisor, Prof. Fulvio Babich, for his valuable insights, encouragement and consistent support. He not only introduced me to the multimedia communication and networking area but also inspired me to meet the challenges along the way.
I would like to thank Prof. Lucio Manià, Dr. Francesca Vatta and the whole Telecommunication Group for taking their valuable time and providing many points of discussion also out of my research activity.
I would also like to thank Massimiliano, Aljosa and all the other students that work with me in the Laboratorio Protocolli for providing constructive comments during my activity.
Special thanks go to my girlfriend Maddalena for her everlasting love and continuous encouragement during my PhD studies. She has been my source of strength during the hard times.
I'd also like to thank all my friends starting from Alberto, Fabio, Pier and including my colleagues at Emaze Networks and many others, for their incessant help in my work and life.
Finally, I am indebted to my mother Maria Rosa and my father Agostino for their support and motivations during this years.

Bibliography

[1] B. Cohen, "Incentives build robustness in bittorrent," http://bitconjurer.org/BitTorrent/bittorrentecon.pdf, May 2003.

[2] X. Zhang, J. Liuy, B. Liz, and Tak-Shing, "CoolStreaming DONet: A Data-Driven Overlay Network for Efficient Live Media StreamingIncentives build robustness in bittorrent," in *IEEE INFOCOM*, vol. 3. INFOCOM, March 2005, pp. 2102– 2111.

[3] Napster Inc., "Napster," http://www.napster.com, 2006.

[4] iTunes Inc., "iTunes," http://www.iTunes.com, 2006.

[5] YouTube Inc., "Broadcast yourself," http://www.youtube.com, 2006.

[6] Skype Inc., "It's free to download and free to call other people on Skype," http://www.skype.com/, 2006.

[7] VoIPStunt Inc., "VoIPStunt," http://www.voipstunt.com/, 2006.

[8] Asterisk Inc., "Asterisk," http://www.asterisk.com/, 2006.

[9] FastWeb Inc., "FastWeb," http://www.FastWeb.com/, 2006.

[10] "Advanced Video Coding for Generic Audiovisual Services," ITU-T and ISO/IEC JTC 1., 2003.

[11] G. Sullivan and T. Wiegand, "Video compression from concepts to the H.264/AVC standard," in *Proceeding of the IEEE*, vol. 93, June 2005.

[12] A. M. Tourapis, K. Sühring, G. Sullivan, *Revision of the H.264/MPEG-4 AVC Reference Software Manual*, Dolby Laboratories Inc., Fraunhofer-Institute HHI, Microsoft Corporation, JVT-W041, April 2007.

[13] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video An Introduction to MPEG-2*. Springer, December 2002.

[14] A. Luthra, G. Sullivan, T. Wiegand, *Special Issue on the H.264/AVC video coding standard*. IEEE Transaction on Circuits and Systems for Video Technology, July 2003, vol. 13, no. 7.

[15] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Transactions on Multimedia*, vol. 8, no. 2, p. 390–404, April 2006.

[16] J. Postel, "User Datagram Protocol," Request for Comments (standard) 768, Internet Engineering Task Force (IETF), August 1980.

[17] C. E. Shannon, "A Mathematical Theory of Communications," *Bell Systems Technology*, p. 379–423, 1948.

[18] Postel J., "DoD standard transmission control protocol," Request for Comments (standard) 761, Internet Engineering Task Force (IETF), January 1980.

[19] Socolofsky T. and Kale C., "A TCP/IP tutorial," Request for Comments (standard) 1180, Internet Engineering Task Force (IETF), January 1991.

[20] "Video coding for narrow telecommunication channels at 64 kbit/s," ITU-T Recommendation H.263, 1996.

[21] S. Bauer, J. Kneip, T. Mlasko, B. Schmale, J. Vollmer, A. Hutter, and M. Berekovic, "The mpeg-4 multimedia coding standard: Algorithms, architectures and applications," *J. VLSI Signal Process. Syst.*, vol. 23, no. 1, 1999.

[22] "Video Codec for audiovisual services at p x 64 kbits/s," ITU-T Recommendation H.261, 1990.

[23] D. Banks and L. A. Rowe, "Analysis tools for mpeg-1 video streams," EECS Department, University of California, Berkeley, Tech. Rep. UCB/CSD-97-936, Jun 1997. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/1997/5497.html

[24] Bernd Girod, "Comparison of the H.263 and H.261 Video Compression Standards."

[25] T. Chujoh and T. Watanabe, "Reversible variable length codes and their error detecting capacity," in *Proceeding of the IEEE*. Portland, (OR): Picture Coding Symposium, 1999, pp. 341–344.

[26] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. Invited Paper, scheduled March 2007.

[27] Stockhammer T. and Bystrom M., "H.264/AVC data partitioning for mobile video communication," in *IEEE International Conference on Image Processing (ICIP)*, October 2004, p. 545–548.

[28] Li A. H., Kittitornkun S., Hu Y. H.,Park D. S. and Villasenor J. D., "Data partitioning and reversible variable length codes for robust video communications," in *IEEE International Conference on Data Compression (DCC)*, March 2000, pp. Snowbird(UT), USA.

[29] Goshi J., Mohr A. E., Ladner R. E., Riskin E. A. and Lippman A. F., "Unequal loss protection for H.263 compressed video," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 15, no. 2, p. 412–419, March 2005.

[30] Karczewicz M. and Kurceren R., "The SP and SI frames design for H.264/AVC," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 13, no. 7, July 2003.

[31] M. Kalman, P. Ramanathan, and B. Girod, "Rate distortion optimized streaming with multiple deadlines," in *In Proceedings of IEEE International Conference on Image Processing (ICIP)*, Barcelona Spain, Ed., September 2003.

[32] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "A Transport Protocol for Real-Time Applications," Request for Comments (standard) RFC 1889, Internet Engineering Task Force (IETF), January 1996.

[33] P. Salama, N. Shroff, E.J. Coyle, and E.J. Delp, "Error concealment techniques for encoded video streams," in *In Proceedings of IEEE International Conference on Image Processing (ICIP)*, Washington DC, Ed., vol. 1, October 1995, pp. 9–12.

[34] W. Zeng and B. Liu, "Geometric-structure-based error concealment with novel applications in block-based low-bit-rate coding," *IEEE Transaction on Circuits and System for Video Technology*, vol. 9, no. 4, pp. 648–665, June 1999.

[35] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," *IEEE Transaction on Image Processing*, vol. 4, no. 7, pp. 470–477, April 1995.

[36] M.J. Chen, L.G. Chen and R.M. Weng, "Error concealment of lost motion vectors with overlapped motion compensation," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 7, no. 3, June 1997.

[37] E. Asbun and E.J. Delp, "Real-time error concealment in compressed digital video streams," in *IEEE International Conference of Picture Coding Symposium*, April 1999, p. Portland (Oregon).

[38] Y.K. Wang, M.M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26L test model," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, September 2002 Rochester(NY) USA, pp. 729–732.

[39] G. Bjontegaard, "Definition of an error concealment model TCON," ITU-T, Boston (USA), ITU-T/SG15/LBC-95-186, June 1995.

[40] Y.O. Park, C.S. Kim, and S.U. Lee, "Multi-hypothesi s error concealment algorithm for H.26L video," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, September 2003 Barcelona, pp. 465–468.

[41] B. Jung, B. Jeon, M.D. Kim, B. Suh, and S.I. Choi, "Selective temporal error concealment algorithm for H.264/AVC," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, October 2004 Singapore, pp. 465–468.

[42] H. Sun and J. Zedepski, "Adaptive error concealment algorithm for MPEG compressed video," in *Proceedings SPIE Visual Communications and Image Processing (VCIP)*, November 2002 Boston(MA) USA, pp. 814–824.

[43] S. Belfiore, M. Grangetto, E. Magli and G. Olmo, "Concealment of Whole-Frame Losses for Wireless Low Bit-Rate Video Based on Multiframe Optical Flow Estimation," *IEEE Transactions on Multimedia*, vol. 7, no. 2, pp. 316–329, April 2005.

[44] T. P.C. Chen and T. Chen., "Second-generation error concealment for video transport over error prone channels," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, September 2002 Rochester(NY).

[45] D. S. Turaga and T. Chen, "Model-based error concealment for wireless video," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 12, no. 6, p. 483–495, June 2002.

[46] S. Hemami and T. Meng., "Transform coded image reconstruction exploiting interblock correlation," *IEEE Transaction on Image Processing*, vol. 4, no. 7, p. 1023–1027, July 1995.

[47] J. W. Park, J. W. Kim, and S. U. Lee, "DCT coefficients recovery-based error concealment technique and its application to the MPEG-2 bit stream error," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 7, no. 6, p. 845–854, December 1997.

[48] P. Salama, N. Shroff, and E. J. Delp, "A Bayesian approach to error concealment in encoded video streams," in *IEEE International Conference on Image Processing (ICIP), Lausanne, Switzerland*, September 1996, p. 49–52.

[49] ——, "A fast suboptimal approach to error concealment in encoded video streams," in *IEEE International Conference on Image Processing (ICIP), Santa Barbara(CA), USA*, September 1997, p. 101–104.

[50] S. Shirani, F. Kossentini, and R. Ward, "A concealment method for video communications in an error-prone environment," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1737–1747, June 2000.

[51] B. W. Wah, X. Su, and D. Lin, "A survey of error concealmen t schemes for real-time audio and video transmissions over the internet," in *In IEEE International Symposium on Multimedia Software Engineering, Taipei, Taiwan*, December 2000, p. 17–24.

[52] 3GPP, "Video adhoc group database for video codec evaluation," 3GPP SA4 Video Adhoc Group, Technical Report S4-050789, September 2005.

[53] A. Ortega and K. Ramchandran, "Rate distortion methods in image and video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, November 1998.

[54] G.J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, November 1998.

[55] T. Wiegand and B. Girod, "Lagrangian multiplier selection in hybrid video coder control," in *IEEE International Conference on Image Processing*, October 2001, p. Thessaloniki (Greece).

[56] ——, "Adaptive intra update for video coding over noisy channels," in *IEEE International Conference on Image Processing*, vol. 3, October 1996, pp. 763–766.

[57] P. Haskell and D. Messerschmitt, "Resynchronization of motioncompensated video affected by ATM cell loss," in *In Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, 1992, pp. 545–548.

[58] Q.F. Zhu and L Kerofsky, "Joint source coding, transport processing, and error concealment for H.323-based packet video," in *In Proceedings of SPIE Visual Communications and Image Processing (VCIP)*, San Jose(CA) (USA), Ed., vol. 3, January 1999, pp. 52–62.

[59] R. Zhang, S.L. Regunthan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 966–976, June 2000.

[60] G. Cote, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error prone networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 952–965, June 2000.

[61] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "A Transport Protocol for Real-Time Applications," Request for Comments (standard) RFC 1889, Internet Engineering Task Force (IETF), January 1996.

[62] H. Yang and K. Rose, "Source channel prediction in error resilient video coding," in *In Proceedings of IEEE International Conference on Multimedia (ICME)*, Baltimore(MD) USA, Ed., July 2003.

[63] C.W. Kim, D.W. Kang, and I.S. Kwang, "High-complexity mode decision for error prone environment," in *Doc. JVT-C101, Joint Video Team (JVT)*, Fairfax(VA), USA, Ed., May 2002.

[64] T. Wiegand, N. Farber, K. Stuhlmller, and B. Girod, "Error resilient video transmission using long-term memory motion compensated prediction," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1050–1062, June 2000.

[65] T. Stockhammer, T. Wiegand, and D. Kontopodis, "Rate-distortion optimization for JVT/H.26L coding in packet loss environment," in *In Proceedings of IEEE International Packet Video Workshop*, Pittsburgh(PY) - USA, Ed., April 2002.

[66] E. Steinbach, N. Farber, and B. Girod, "Standard compatible extension of H.263 for robust video transmission in mobile environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 6, p. 872–881, December 1997.

[67] B. Girod and N. Farber, "Feedback-based error control for mobile video transmission," in *Proceeding of the IEEE*, October 1999, p. 1707–1723.

[68] W. Wada, "Selective recovery of video packet losses using error concealment," *IEEE Journal on Selected Areas in Communications*, vol. 7, p. 807–814, June 1989.

[69] T.Wiegand, N. Färber, K. Stuhlmller, and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, p. 1050–1062, June 2000.

[70] H. Liu, H.Ma, M. E. Zarki, and S. Gupta, "Error control schemes for networks: An overview," *Mobile Networks and Applications*, no. 2, p. 167–182, June 1997.

[71] W. Kumwilaisak, J. Kim, and C. Kuo, "Reliable wireless video transmission via fading channel estimation and adaptation," in *Proceedings of IEEE WCNC, Chicago, IL*, September 2000, p. 185–190.

[72] S. Lin and D. C. Jr., *Error Control Coding*, 2nd ed., ed., Ed. Prentice Hall, 2004.

[73] *IEEE P802.11. Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)*, IEEE Task Group 802.11, November 1997.

[74] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and PHYsical Layer (PHY) Specifications: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, IEEE Task Group 802.11e, November 2005.

[75] L. Rizzo, "On the feasibility of software FEC," DEIT Technical Report," LR-970131, 2003.

[76] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Transactions on Information Theory*, vol. 42, pp. 1737–1747, November 1996.

[77] P. G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," in *IEEE Signal Processing Letters*, vol. 4, July 1997, p. 189–191.

[78] T. Stockhammer and C. Weiss, "Channel and complexity scalable image transmission," in *IEEE International Conference on Image Processing, Thessaloniki, Greece*, vol. 1, October 2001, p. 102–105.

[79] D. G. Sachs, R. Anand, and K. Ramchandran, "Wireless image transmission using multiple-description based concatenated code," in *IEEE VCIP , San Jose, CA,*, January 2000, p. 300–311.

[80] P. G. Sherwood and K. Zeger, "Error protection for progressive image transmission over memoryless and fading channels," *IEEE Transaction on Communications*, vol. 46, p. 1555–1559, December 1998.

[81] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," Microsoft Research, Beijing, China, Tech. Rep. 35, 2001.

[82] M. Podolsky, S. McCanne, and M. Vetterli, "Soft ARQ for layered streaming media," University of California, Computer Science Department, Berkeley, CA, Technical Report UCB/CSD-98-1024, November 1998.

[83] P. A. Chou and A. Sehgal, "Rate-distortion optimized receiver-driven streaming over best-effort networks," in *Proceeding of the International Packet Video Workshop, Pittsburgh, PA, USA*, April 2002.

[84] E. Setton, J. Noh, and B. Girod, "Congestion-distortion optimized peer-to-peer video streaming," in *IEEE International Conference on Image Processing, ICIP-2006, Atlanta, GA*, October 2006.

[85] E. Setton and B. Girod, "Congestion-distortion optimized scheduling of video over a bottleneck link," in *In IEEE Workshop on Multimedia Signal Processing, Siena, Italy*, September 2004.

[86] P. A. Chou and A. Sehgal, "Rate-distortion optimized receiver-driven streaming over best-effort networks," in *In Packet Video Workshop, Pittsburgh, PA*, April 2002.

[87] J. Chakareski, P. A. Chou, and B. Girod, "Rate-distortion optimized streaming from the edge of the network," in *In IEEE Workshop on Multimedia Signal Processing, St. Thomas, US Virgin Islands*, December 2002.

[88] J. Chakareski and B. Girod, "Rate-distortion optimized video streaming with rich acknowledgments," in *In Proceedings SPIE Visual Communications and Image Processing VCIP, Santa Clara, CA*, January 2004.

[89] V. Paxson, "Measurement and Analysis of End-to-end Internet Dynamics," Ph.D. dissertation, UC Berkeley, 1997.

[90] K. Stuhlmuller, N. Faber, M. Link and B. Girod, "Analysis of Video Transmission over Lossy Channels," *IEEE Transaction on Selected Areas in Communications*, vol. 18, no. 6, pp. 1012–1032, June 2000.

[91] Y. J. Liang, J. G. Apostolopoulos and B. Girod, "Analysis of Packet Loss for Compressed Video: Does Burst-Length Matter?" *IEEE Transactions on Circuits and Systems for Video Technology*, 2007.

[92] L. Choi, M. Ivrlac, E. Steinbach and J. Nossek, "Analysis of distortion due to packet loss in streaming video transmission over wireless communication links," in *IEEE International Conference on Image Processing (ICIP)*, September 2005, pp. 11–14.

[93] J. Chakareski, J. G. Apostolopoulos, S. Wee, W. Tan and B. Girod, "Rate-Distortion Hint Tracks for Adaptive Video streaming," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 15, no. 10, pp. 1257–1269, October 2005.

[94] E. Masala and J. C. De Martin, "Analysis-by-synthesis distortion computation for rate-distortion," in *IEEE International Conference on Multimedia and Expo*, Baltimore, MD, Ed., vol. 3, July 2003, p. 345–348.

[95] W. Tu, W. Kellerer, and E. Steinbach, "Rate-Distortion Optimized Video Frame Dropping on Active Network Nodes," in *IEEE Proceedings of the International Packet video Workshop*, Irvine, CA, Ed., December 2004.

[96] S. Ekmekci and T. Sikora, "Recursive decoder distortion estimation based on source modeling for video," in *Proceeding of International Conference on Image Processing(ICIP)*, 2004, p. 187–190.

[97] Y. Zhang, W. Gao, H. Sun, Q. Huang, and Y. Lu, "Error resilience video coding in H.264 encoder with potential distortion tracking," in *Proceeding of International Conference on Image Processing(ICIP)*, vol. 1, 2004, p. 173–176.

[98] T. Stockhammer, T.Wiegand, and S.Wenger, "Optimized transmission of H.26L/JVT coded video over packet-lossy networks," in *Proceeding of International Conference on Image Processing(ICIP), Rochester, NY*, vol. 2, 2002, p. 173–176.

[99] T. Wiegand, N. Farber, K. Stuhlmuller, and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 1050–1062, June 2000.

[100] Y. Shen, P. C. Cosman, and L. Milstein, "Video coding with fixed length packetization for a tandem channel," *IEEE Transaction on Image Processing*, vol. 15, no. 2, pp. 273–288, February 2006.

[101] A. R. Reibman, L. Bottou, and A. Basso, "DCT-based scalable video coding with drift," in *Proceedings of the IEEE International Conference of Image Processing (ICIP)*, vol. 2, December 2001, p. 989–992.

[102] H. Yang, R. Zhang, and K. Rose, "Drift management and adaptive bit rate allocation in scalable video coding," in *Proceedings of the IEEE International Conference of Image Processing (ICIP)*, vol. 2, December 2002, p. 49–52.

[103] B. A. Heng, J. G. Apostolopoulos, and J. S. Lim, "End-to-end rate-distortion optimized mode selection for multiple description video coding," in *Proceedings of ICASSP*, vol. 52, 2005, p. 905–908.

[104] H. Yang and K. Rose, "Rate-distortion optimized motion estimation for error resilient video coding," in *Proceedings of ICASSP*, 2005, p. 187–190.

[105] F. Zhai, C. E. Luna, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Joint source coding and packet classification for video streaming over differentiated services networks," *IEEE Transaction on Multimedia*, vol. 7, no. 4, p. 716–726, January 2005.

[106] F. Zhai, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Rate-distortion optimized hybrid error control for packetized video communications," *IEEE Transaction on Image Processing*, vol. 15, no. 1, p. 40–53, January 2005.

[107] E. Masala, H. Yang, and K. Rose, "Rate-distortion optimized slicing, packetization and coding for error-resilient video transmission," in *Proceedings of IEEE DCC*, 2004, p. 182–191.

[108] M. Fumagalli, M. Tagliasacchi, and S. Tubaro, "Improved bit allocation in an error-resilient scheme based on distributed source coding," in *Proceedings of ICASSP*, vol. 2, May 2004, p. 61–64.

[109] H. Yang and L. Lu, "A novel source-channel constant distortion model and its application in error resilient frame-level bit allocation," in *Proceedings of ICASSP*, vol. 3, 2004, p. 277–280.

[110] M. Fumagalli, R. Lancini, and S. Tubaro, "Video quality assessment from the perspective of a network service provider," in *Proceedings of International Workshop on Multimedia and Signal Processing*, October 2006, p. 324–328.

[111] T. Stockhammer, M. M. Hannuksela, and T.Wiegand, "H.264/AVC in wireless environments," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, July 2003, p. 657–673.

[112] S. Wenger, "H.264 over IP," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, July 2003.

[113] Kevin Fall and Kannan Varadhan, *The ns Manual (formerly ns Notes and Documentation)*, UC Berkeley, LBL, USC/ISI, and Xerox PARC, http://www.isi.edu/nsnam/ns/, 2005.

[114] *OPNET Users' Manual*, OPNET Architecture, http://forums.opnet.com.

[115] F. Babich and M. Vitez, "A Novel Wide-Band Audio Transmission Scheme over the Internet with a Smooth Quality Degradation," *ACM SIGCOMM Computer Communication*, vol. 30, no. 1, January 2000.

[116] Luis MG, "TcpDump Libpcap," http://www.tcpdump.org/tcpdumpman.html, 2008.

[117] -, "Wireshark: Network Protocol Analyzer." http://www.wireshark.org/, 2008.

[118] G. Navlakha and J. Ferguson, "IPerf Tool," 2008.

[119] E. Kohler, "The Click Modular Router," Ph.D. dissertation, MIT, November 2000.

[120] Multiband Atheros WiFi Project, "http://madwifi.org," 2008.