NORTHWESTERN UNIVERSITY

Optimal Cross-Layer Resource Allocation for Real-Time Video Transmission over Packet Lossy Networks

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Electrical and Computer Engineering

By

Fan Zhai

EVANSTON, ILLINOIS

June 2004

ABSTRACT

Optimal Cross-Layer Resource Allocation for Real-Time Video Transmission over Packet Lossy Networks

Fan Zhai

Real-time video applications, such as videoconferencing, videophony, and on-demand video streaming, have gained increased popularity. However, a key problem of video transmission over the existing Internet and wireless networks is the incompatibility between the nature of the network conditions and the QoS (Qualify of Service) requirements (such as those in bandwidth, delay, and packet loss) of real-time video applications. Cross-layer design is a natural approach to deal with the incompatibility problem. This approach aims to efficiently perform cross-layer resource allocation (such as bandwidth, transmission energy, and buffers) by increasing the communication efficiency of multiple network layers.

Our focus is on the end-system design. We assume that the lower layers provide a set of given adaptation components; from the encoder's point of view, these components can be regarded as network resource allocation "knobs". Assuming that our encoder can access and specify those adaptation components, we propose a resource-distortion optimization framework, which optimally assigns cross-layer resources to each video packet according to its level of importance.

The proposed framework is general and flexible. Within the framework, we can jointly consider the available error control components in different network infrastructures. In particular, we study the following problems: (1) joint source-channel coding (JSCC) for real-time Internet video transmission, (2) joint source-channel coding and power adaptation (JSCCPA) for real-time wireless video transmission, and (3) joint source coding and packet classification (JSCCPC) for real-time DiffServ (differentiated services) network video transmission. Besides single layer video source coding, we also consider using scalable video source coding for further error resilience. In addressing each of the above problems, we propose efficient algorithms for obtaining the optimal solutions. The simulation results, as expected, demonstrate the benefits of joint design of source coding and cross-layer resource allocation. In addition, the proposed framework serves as an optimization benchmark against which the performances of other sub-optimal systems can be evaluated, and also provides a useful tool in assessing the effectiveness of different error control components in practical system design.

Acknowledgments

Pursing a doctoral degree is a long journey that one cannot make alone. I would like to thank all those who have assisted me in one way or another along this journey.

First, this dissertation is dedicated to my grandma and my parents. Their unconditional love, sustained understanding and support all along renders this dissertation possible and my life meaningful and colorful.

My next acknowledgement is a tribute to my research advisor, Prof. Thrasyvoulos N. Pappas, for his ideas, patience, valuable discussions and guidance. I am especially grateful for his financial support, and truly inspired by his enthusiasm in research. I owe the most overwhelming debt to my other research advisor, Prof. Aggelos K. Katsaggelos. Aggelos has been a fantastic advisor in many respects. I have yet to see the limits of his wisdom, knowledge, vision, and concern for his students. His warm encouragement helped me through the hardest time of my Ph.D. study. In addition, I would like to thank both Thrasos and Aggelos for giving me the freedom to pursue my research, while also providing constructive advice and discussion along the way.

I would also like to thank Prof. Randall Berry who not only serves as my thesis committee member but also gets deeply involved with the development and maturation of my research. I am deeply grateful for his guidance during the weekly group meeting, and his valuable inputs and critiques during the writing of all the papers that constitute this dissertation.

Together with the above three professors, the other two colleagues with whom I closely work are Dr. Carlos E. Luna and Yiftach Eisenberg. I feel very fortunate that there is such a great group that we can work with, discuss with, and challenge each other. The weekly meetings with them have been one of the most important items in my schedule.

Besides Carlos and Yifty, I am also thankful for many other IVPL members, Zhilin Wu, Peshala Pahalawatta, Petar Aleksic, Passant Karunaratne, Sotiris Tsaftaris, Konstantinos E. Zachariadis, Haohong Wang, and Dr. Junqing Chen. Thanks for their friendship, help, and support.

I would like to thank my uncle, Dr. Tan Du, for his valuable suggestion in choosing my field of study, continuous encouragement, and help in bringing me into Texas Instruments as a summer co-op.

Words are inadequate to express my thanks to my host family, Lucy and Martin Reinheimer. They are not only the first Americans who hosted me in my first week in the USA, but also the *family* who helped me prepare the essential furniture, who helped me improve my spoken English, and who treated me to so many lunches and dinners.

I also would like to thank my girlfriend. I greatly appreciate her help in editing this dissertation and most of my papers. I cannot imagine coming up with a well-written piece on time without her help.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Real-time video transmission is largely achieved through applications that impose an end-to-end delay constraint on the video stream. Those real-time applications include conversational applications such as videoconferencing, distance learning, and videophony. Such applications usually have strict end-to-end delay constraint, e.g. less than 200 milliseconds. Real-time applications may also include streaming applications. Those applications allow the start of video playback before the whole video stream has been transmitted with an initial setup time usually of a few seconds. All those applications require real-time playback. That is, once the playback starts, it must be continuous without interruption.

Real-time video applications have gained increased popularity since the introduction of the first commercial products for Internet video streaming in 1995 [1]. As wireless networks are quickly becoming an important component of the modern communications infrastructure, Internet protocol (IP)-based architecture for the third-generation (3G) wireless systems grows to be the provider of the next generation wireless services such as voice, high-speed data, Internet access, and multimedia streaming on all IP networks [2, 3]. The high bit rate support (54 and 22 Mbps in IEEE 802.11a and 802.11b respectively) in the WLAN (Wireless Local Area Network) standard makes it possible to transmit video in WLAN [4,5]. All the above-mentioned cutting-edge developments, however, confront the high technical hurdles associated with high bit rate, quality of service (QoS), and real-time requirements of video applications [6, 7].

1.1 Challenges for Real-Time Video Transmission

Generally speaking, the main challenge to the real-time video communications is how to reliably transmit video packets over error-prone networks, where meeting the transmission deadline is complicated by the variability in throughput, delay, and packet loss in the network. In particular, a key problem of video transmission over the existing Internet and wireless networks is the incompatibility between the nature of the network conditions and the QoS requirements (such as those pertaining to bandwidth, delay, and packet loss) of multimedia applications. With a best-effort approach, the current IP network was originally designed for data transmission, having no guarantee of QoS for multimedia applications. Similarly, the current wireless networks were designed mainly for voice communication, which does not require as large bandwidth as video applications do. Different types of IP applications have different types of quality impairment under the same network conditions, and therefore call for different QoS. For example, “elastic” applications such as web browsing, data file transfer, and electronic mail, are not sensitive to delay. However, for the deployment of multimedia applications with video stream, which is more sensitive to delay but more tolerant to packet loss, the lack of QoS guarantees in today's Internet and wireless networks introduces huge complications [8,9]. Specifically, several technological challenges need to be addressed in designing a high-quality video transmission system.

First, to achieve acceptable delivery quality, transmission of a real-time video stream typically has a minimum bandwidth requirement. However, the current Internet does not provide bandwidth reservation to meet the bandwidth requirement. At the same time, compared to wired links, wireless channels are much noisier due to fading, multi-path, and shadowing effects, which result in a much higher bit error rate (BER) and consequently an even lower throughput [6,7].

Second, in the Internet, a packet can be lost due to congestion caused by buffer overflow and excessive delay. In wireless networks, a packet with unrecoverable bit errors is usually discarded at the link-layer according to the current standards. This difficulty is not as severe for traditional IP applications such as data transfer and email, where reliable transmission can always be achieved through retransmission, as it is for real-time video applications, where retransmission-based techniques may not be available due to the tight delay constraints.

Third, the network resources are limited and may vary with time and space. Resource is a general term in communication networks. Network resources include transmission bandwidth, buffers in the routers and switches, buffers at the sender or the receiver end, computation capability for encoding, decoding and transcoding, transmission cost in networks with pricing charge enabled, transmission power in wireless communications, delay, etc. Some constraints on resource, such as buffer size, computation speed, and display precision at the user end, are “hard”; but other constraints are “soft” in that they aim to make the system stable or to treat other users in the communication system fairly. One example of a *soft constraint* at hand is the TCP-friendly protocol used to perform congestion control for media delivery applications, where the source bit rate is constrained so that all kinds of traffics can fairly share the network resources [8].

1.2 Cross-Layer Resource Allocation

We can address the above challenges by enforcing error control, especially through unequal error protection (UEP) for video packets that are usually of different importance. Error control techniques, in general, include error resilient source coding, forward error correction (FEC), retransmission, power control, network QoS support, and error concealment. To maximize the error control efficiency, limited network resources should be optimally allocated to video packets, which typically requires cross-layer design.

The traditional layered protocol stack, where various protocol layers can only communicate with each other in a restricted manner, has proved to be inefficient and inflexible in adapting to the constantly changing network conditions [10]. For the best end-to-end performance, multiple protocol layers should be jointly designed and should be able to react to the channel conditions in order to make the end-system *network-adaptive*.

In addition, conventional video communication systems have focused on video compression, namely, rate-distortion optimized source coding, without considering other layers [11]. While these algorithms can produce significant improvements in source-coding performance, they are inadequate for video communications over hostile channels. This is because Shannon's separation theorem [12], that source coding and channel coding can be separately designed without any loss of optimality, does not apply to general time-varying channels, or to systems with a complexity or delay constraint (i.e. any real time system). Therefore, recent video coding research has been focused on the investigation of joint design of end-system source coding with manipulations in other layers, such as channel coding, power adaptation in wireless networks, and QoS support from the network (e.g., differentiated services networks and integrated services networks) [1].

One of the main characteristics of video is that different portions of the bitstream have different importance in their contribution to the quality of the reconstructed video. For example, in an MPEG video bitstream, I (Intra) frames are more important than P (Predictive) and B (Bi-directional predictive) frames. If the bitstream is partitioned into packets, Intra-coded packets are usually more important than Inter-coded packets. If error concealment is used, the packets that are hard to conceal are usually more important than easily concealable ones. In the scalable video bitstream, the base layer is more important than the enhancement layer. Therefore, UEP is naturally preferable in video transmission. UEP commonly enables a prioritized protection for video packets through different levels of FEC and/or retransmission [13]. UEP can also be realized through prioritized transmission by techniques based on transmitter power adaptation, channel/path diversity, or DiffServ (Differentiated Services) [14, 15].

Thus, in different network infrastructures, we consider different applicable error control components to support UEP. In particular, we jointly consider source coding and cross layer resource allocation to perform optimal UEP.

1.3 Scope and Contributions

Figure 1.1 illustrates the Open Systems Interconnection (OSI) 7-layer communication model, where layer 5 (session layer), layer 6 (presentation layer) and layer 7 (application layer) are merged into one application layer to better describe our work. The resulting 5-layer model is also called TCP (Transport Control Protocol) model. The general functions of these layers are as follows.

Layer 5, 6, 7: The application layer–This layer provides applications services to user and programs, such as file transfers, http, electronic mail, etc.

Layer 4: The transport layer–This layer provides transparent transfer of data between hosts and is responsible for end-to-end error-correction and flow control. TCP and UDP (User Data Protocol) work at this level.

Layer 3: The network layer–This layer deals with network addressing and routing of data between two hosts and any congestion that might develop.

Layer 2: The link (or data-link) layer–This layer defines the format of data on the network. In IP-based wireless networks, the link layer is divided into two sublayers: the MAC (Medium Access Control) layer and LLC (Logical Link Control) layer. The MAC sublayer is responsible for communications between stations by coordinating medium access. The LLC sublayer manages frame synchronization, error correction, and flow control.

Layer 1: The physical layer–It deals with physical aspects such as physical media, electrical impulse, transmitter power, modulation and etc.

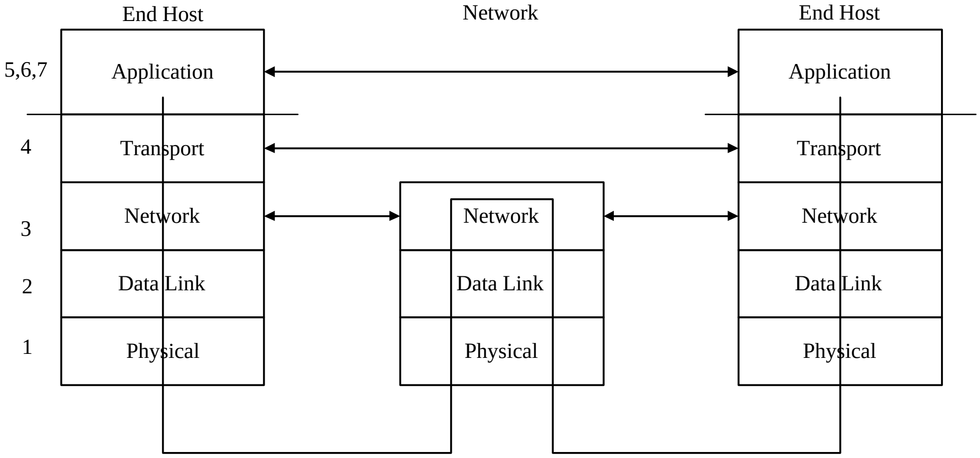


Figure 1.1: OSI networking model.

This dissertation is about the end system design for real-time video applications. Such end system design consists of three major components: video codec, rate control or congestion control, and error control. This dissertation focuses on error control. More specifically, our focus is on the interaction between the video encoder and the underlying layers. Thus the decoder-based techniques, such as post-processing, joint source-channel decoding [16], and receiver-driven channel coding [17,18] are not discussed in this dissertation. With respect to cross-layer design at the sender side, traditionally, there are two approaches to address the challenges described above for video communications. In the networking community, the approach is to develop protocols and mechanisms to adapt the network to the video applications. One example is to modify the mechanisms implemented in the routers/switches to provide QoS support to guarantee bandwidth, bounded delay, delay jitter, and packet loss (such as DiffServ and InteServ) for video applications. Another example is to use additional components such as an overlay network (e.g., an edge proxy which provides application layer functions like transcoding, rate adaptation, FEC ect.). The networking community approach is beyond the scope of this dissertation.

In the video community, on the other hand, the approach is to adapt the end system to the network, which is what we employ in this dissertation. We focus on the end-system design for video communications based on the current, recently proposed, and emerging network protocols and architectures. We assume that the lower layers provide a set of given adaptation components; from the encoder's point of view, these components can be regarded as network resource allocation "knobs". Based on the assumption that our encoder can access and specify those resource allocation knobs, the major contribution of this dissertation is a proposed general resource-distortion optimization framework, which assigns network resources from multiple layers to each video packet according to its level of importance. Depending on different adaptation components, this framework is embodied in the forms of joint source-channel coding, joint source coding and rate adaptation, joint source-channel coding and power adaptation, joint source coding and packet classification etc. Within the framework, error resilient source coding, channel coding, and error concealment are jointly considered.

In particular, we study the following problems using the proposed framework: (1) joint source-channel coding (JSCC) for Internet video transmission, (2) joint source-channel coding and power adaptation (JSCCPA) for wireless video transmission, and (3) joint source coding and packet classification (JSCCPC) for DiffServ (differentiated services) network video transmission. Besides single layer video source coding, we also consider scalable video source coding for further error resilience. In addressing these problems, we propose efficient algorithms for obtaining the optimal solutions. The simulation results, as expected, demonstrate the benefits of joint design of source coding and cross layer resource allocation.

The proposed framework proves to be general and flexible. It allows for the comparison of different error control techniques (such as pure FEC, pure retransmission, and hybrid FEC and selective retransmission), and different packetization schemes. For example, in the JSCC work, FEC and application layer retransmission can each achieve an optimal result depending on the packet loss rates and round-trip-time. But when the two are jointly employed in the proposed hybrid technique, improved results are obtained due to the increased flexibility. The framework is also applicable to the emerging network architectures (such as DiffServ), in which source coding can be jointly designed with packet classification.

The proposed framework provides an optimization benchmark against which the performances of other sub-optimal systems can be evaluated. It also provides a useful tool for assessing the effectiveness of different error control components in practical system design. For example, in the JSCCPA study, the simulation results suggest that channel coding and power adaptation each has its effective working region. Thus, in a practical wireless video streaming system, under certain conditions, adjusting only channel coding or power control, but not both, might be adequate to achieve near-optimal results.

1.4 Dissertation Organization

The rest of dissertation is organized as follows.

Chapter 2 presents the necessary background information. We give a general overview of video communication systems, communication networks, and error control techniques.

Chapter 3 describes the details of end-system design based on cross-layer resource allocation. Particularly, we propose a general and unified resource-distortion optimization framework, where error resilient source coding, channel coding, power adaptation, network QoS, and error concealment are jointly considered.

Chapter 4 studies the problem of joint source-channel coding for video transmission over the Internet. Two problems are specifically addressed: application-layer packetization and application-layer hybrid FEC and selective retransmission.

Chapter 5 studies the problem of joint source-channel coding and power adaptation for video transmission over IP-based wireless networks.

Chapter 6 studies the problem of joint source coding and packet classification for video transmission over DiffServ networks.

Chapter 7 extends the work in Chapter 4, 5, and 6 to scalable video. In particular, we study the joint source-channel coding for Internet scalable video transmission, and optimal power allocation for energy efficient wireless video communications based on MPEG-4 FGS video.

Chapter 8 draws conclusions with a summary of the research results, contribution of this dissertation, and future research directions.

Chapter 2

Background

To introduce the background for the dissertation, in this chapter, we first give a brief overview of video communication systems, followed by that of communication networks. In the end, we briefly discuss the error control techniques for video communications.

2.1 Video Communication Systems

As shown in Fig. 2.1, a video communication system has five major components: 1) The source encoder that compresses video and audio signals into media packets, which are sent directly to lower layers or uploaded to the media server for storage and later transmission on demand; 2) The application layer in charge of channel coding, packetization, and etc.; 3) The transport layer that performs congestion control and delivers media packets from the sender to the receiver for the best possible user experience, while sharing network resources fairly with other users; 4) The transport network which delivers packets to the client; 5) The receiver that decompresses and renders the video packets, and implements the interactive user controls based on the specific applications [1, 13].

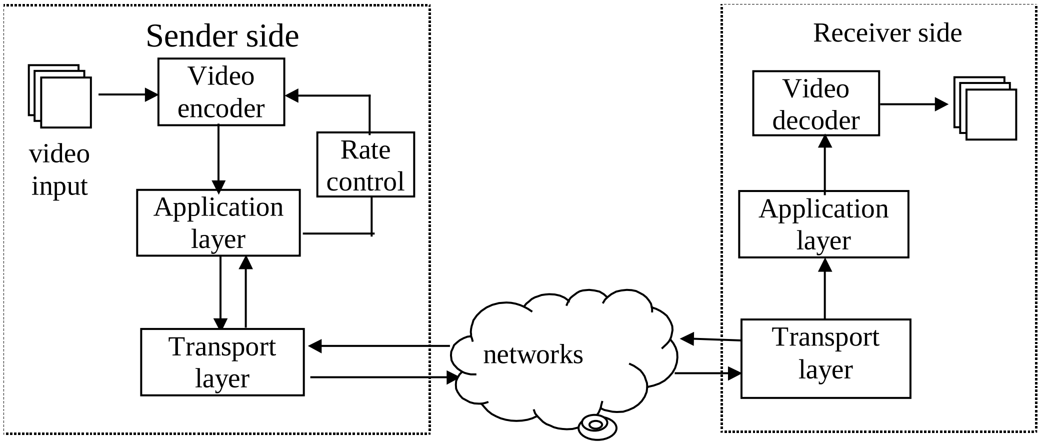


Figure 2.1: Video transmission system architecture.

At the sender end, video packets (referred to as *source packets*) are generated by a video encoder. The source bit rate is constrained by a rate controller that is responsible for allocating bits to each video frame or packet. This bit rate constraint is set based on the estimated channel state information (e.g., available channel bandwidth) reported by the lower layers. After passing through the network protocol stack (e.g. RTP/UDP/IP), *transport packets* are fed into a first-in-first-out (FIFO) encoder buffer before entering a packet lossy network, which can be the Internet, a wireless network, or a heterogeneous network. The network may have multiple channels (e.g., a wireless network) or paths (e.g., a network with path diversity), or support QoS (e.g., integrated services or differentiated services networks). Some packets may be dropped in the network due to congestion, or at the receiver because of excessive delay or unrecoverable bit error in a wireless network. To combat packet losses, parity check packets used for FEC may be generated in the application/transport layer. In addition, lost packets may be retransmitted if applicable. Packets that reach the decoder on time are buffered in the decoder buffer. We define an *initial setup time* (also referred to as the maximum end-to-end delay), T_{max}, as the duration between the time when the first packet is captured at the encoder and its playback at the decoder. The longer the initial setup time, the more robust the system is to channel variations. The setup time is application dependent, and is limited by how long a user is willing to wait for the video to be displayed. The transport layer and application layer are responsible for de-packetizing the received transport packets from the decoder buffer, channel decoding (if FEC is used), and forwarding the intact and recovered video packets to the video decoder. The video decoder then decompresses video packets and displays the resulting video frames in real-time (i.e., the video is displayed continuously without interruption at the decoder). The video decoder typically employs error detection and concealment techniques to mitigate the effects of packet loss.

Next, we discuss each component of the video communication system in detail.

2.1.1 Video Encoder

The source encoder takes raw video data and compresses the media source by reducing the temporal and spatial redundancy. In the past, due to the significant development of digital video applications, several successful standards have emerged under the joint force of academia and industry. There are two main families of video compression standards: the H.26x family and the MPEG (Moving Picture Experts

Group) family. These standards are application-oriented and address a wide range of issues such as bit rate, complexity, picture quality, and error resilience.

The H.26x family of standards, developed by the International Telecommunications Union-Telecommunications Sector (ITU-T), aim at telecommunication applications and have developed from ISND and T1/E1 service to embrace PSTN (Public Switched Telephone Network), mobile wireless networks, and LAN/Internet network delivery. The first standard of this family is H.261 ('90), which was designed for video communications at rates of $p \times 64$ kbps where $6 \leq p \leq 30$ with low coding delay [19]. The H.263 standard ('95) (typically operates below 64 kbps), albeit originally designed for very low bit rate applications, could provide a significant improvement over H.261 at any bit rate [20]. As an extension of H.263, H.263+ and H.263++ ('97) [21] provide 12 new negotiable modes and additional features such as unrestricted motion vector mode, slice structure mode, scalability, etc. These modes and features further improve compression performance and error resilience. H.26L, an on-going standard, aimed to achieve substantially higher video quality than the existing video standards at all bit rates [22]. It was merged with MPEG-4 AVC (Advanced Video Coding), which we will discuss later.

The other family of standards is MPEG, developed by the MPEG group of International Standards Organization (ISO). MPEG-1 standard ('94) was designed for CD-ROM applications with rates below 1.5 Mbps [23]. MPEG-2 ('95), also called H.262, was designed for DVD, HDTV (High Definition Television) and digital satellites applications with rates between 2 and 20 Mbps [24]. As a big improvement, MPEG-4 visual (MPEG-4 part 2) standard('99) [25], which extends to object-based video, aims at low bit rate applications as well as interactive multimedia applications.

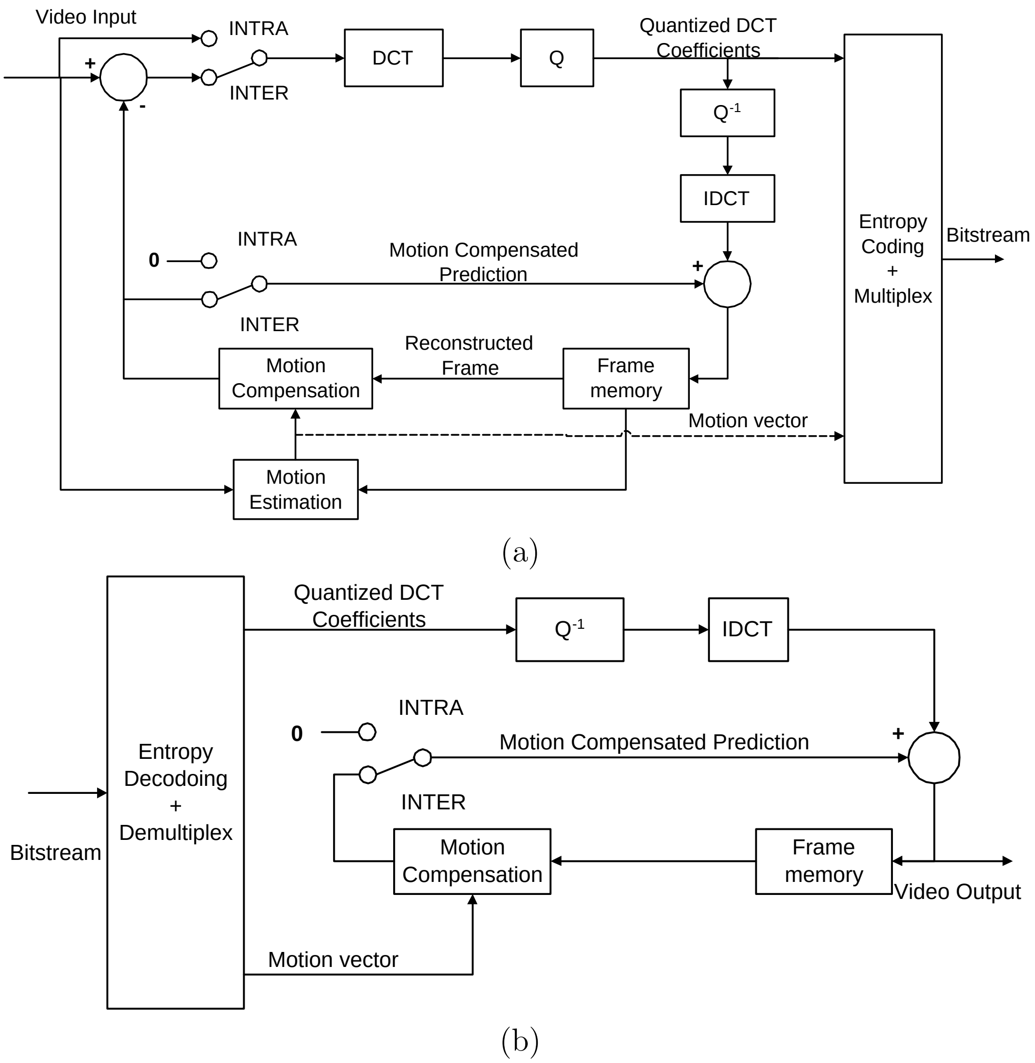


Figure 2.2: Hybrid block-based motion-compensated (a) video encoder and (b) video decoder.

The goal of MPEG-4 standard is to support new functionalities, such as improved coding efficiency, error robustness, and content-based access, manipulation, and scalability. The newest standard is H.264/AVC, aiming to provide the state-of-the-art compression technologies. It is the result of the merger between the MPEG-4 group and the ITU H.26L committee in 2001, known as JVT (Joint Video Team), and is a logical extension to the previous standards adopted by the two groups. Thus, it is also called H.264, AVC or MPEG-4 part 10 [26]. The standardization of H.264/AVC is still ongoing. For an overview and comparison of the video standards, see [27]. MPEG-7 and MPEG-21 standards target the multimedia content description interface, which is different from traditional multimedia coding. It is important to note that all the standards are decoder standards, i.e., they standardize the syntax for the representation of the encoded bitstream and define the method for decoding process, but leave substantial flexibility in the design of the encoder. This limitation on the scope of standardization allows the maximal latitude of optimization for specific applications [26].

Nevertheless, from the compression point of view, all the above mentioned video compression standards share the same block diagram, as shown in Fig. 2.2. This type of video codec follows the so-called block-based hybrid motion-compensated approach, where each video frame is presented in block-shaped units of associated luma and chroma samples (16×16 region) called MBs (macroblocks). As shown in Fig. 2.2(a), the core of the encoder is motion compensated prediction (MCP). The first step in MCP is motion estimation (ME), aiming to find the region from the previous frame that best matches each MB in the current frame. The offset between the MB and the prediction region is known as a motion vector. The motion vectors form a motion field, which is entropy encoded. The second step is motion compensation (MC), where the reference frame is produced by applying the motion field to the previously reconstructed frame. The prediction error, known as the displaced frame difference (DFD), is obtained by subtracting the reference frame from the current frame.

Following MCP, there are three major blocks to process the DFD, namely, transform, quantization, and entropy coding. The key reason in using transform is to decorrelate the data so that the associated energy in the transform domain is more compact and thus the resulting transform coefficients are easier to encode. DCT (Discrete Cosine Transform) is one of the most widely used transforms in image and video coding due to its high transform coding gain and low computational complexity. Quantization introduces loss of information, and is the primary source of actual compression. Quantized coefficients are entropy encoded, e.g. using Huffman or Arithmetic coding. As shown in the figure, the DFD is first divided into 8×8 blocks, and DCT is then applied to each block, with resulting coefficients quantized. In these standards, a given MB can be intraframe coded, interframe coded using motion compensated prediction, or simply replicated from the previously decoded frame. These prediction modes are denoted as INTRA, INTER, and SKIP mode, respectively. Quantization and coding are performed differently for each MB according to its mode. Thus, the coding parameters for each MB are typically represented by its prediction mode and quantization parameter.

In the decoder, as shown in Fig. 2.2(b), the quantized DCT coefficients are inversed DCT (IDCT) transformed to obtain a reconstructed version of the DFD; the reconstructed version of the current frame is obtained by adding DFD to the

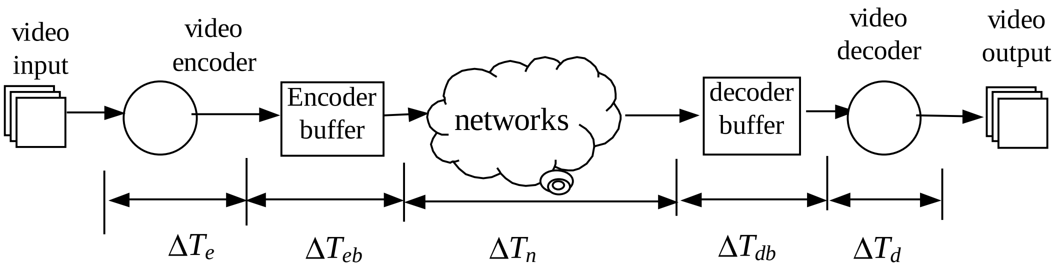


Figure 2.3: Video transmission system block diagram.

previously reconstructed frame.

2.1.2 Delay Components

In a video transmission system, the end-to-end delay (i.e., the time between when a frame is captured at the encoder and when it is displayed at the decoder) should be constant, if the encoder and decoder are to operate at the same frame rate of F frames per second [28]. As shown in Fig. 2.3, the end-to-end delay T of each frame can be decomposed into

$$T = \Delta T_e + \Delta T_{eb} + \Delta T_n + \Delta T_{db} + \Delta T_d, \tag{2.1}$$

where ΔT_e, ΔT_{eb}, ΔT_n, ΔT_{db}, and ΔT_d are, respectively, the encoder delay, encoder buffer delay, network delay, decoder buffer delay, and decoder delay for each frame [28, 29]. Rate control mechanisms are needed to perform bit allocation among frames to ensure that the encoder and decoder buffer do not overflow or underflow.

We assume that a rate controller specifies a deadline by which each frame must be transmitted. We then translate the frame deadline into a delay constraints for each packet. Let M be the number of packets in a video frame and k the packet index. Without loss of generality, we assume that the processing times for both encoding and decoding a packet are constant and equal to $T_p = 1/(MF)$. Note that the k-th packet enters the encoder buffer after the previous $(k-1)$ packets and itself have been processed (i.e., at time kT_p). In addition, in order for the corresponding frame to be displayed on time, this packet must arrive at the decoder in time to allow itself and the following $M-k$ packets to be processed (i.e., $(M-k+1)T_p$ before display) [28,29]. Thus, for each packet to experience constant end-to-end frame delay, it must be that

$$\Delta T_{eb}(k) + \Delta T_n(k) + \Delta T_{db}(k) = T - kT_p - (M-k+1)T_p = T - (M+1)T_p. \quad (2.2)$$

In order to avoid decoder buffer underflow, i.e., to satisfy $\Delta T_{db}(k) \geq 0$, the total encoder buffer delay and network delay must be

$$\Delta T(k) = \Delta T_{eb}(k) + \Delta T_n(k) \leq T_{max}, \quad (2.3)$$

where $T_{max} = T - (M+1)T_p$. For simplicity, we assume a sufficiently large decoder buffer, so that we do not need to consider decoder buffer overflows.

2.1.3 Rate Control

At a certain available transmission rate[1], the task of rate control is to smartly allocate bit budget between frames (frame-level rate control) and within each frame (MB-level rate control) in order to maximize the overall transmission quality. Basically, the number of bits should be allocated to frames and MBs according to their contents. For example, more bits are usually needed to encode the frames and MBs with scene change, high motion, or rich details. We want to limit our discussion to frame-leve rate control; thus the term "rate control" refers only to frame-level rate control hereof.

[1] The detection of the available transmission rate is achieved by congestion control, which is discussed in Sect. 2.2.2.

Rate control has been studied from the memory perspective, i.e., designing rate control to avoid overflowing of the available encoder and decoder buffers [30, 31]. Alternatively, it can also be studied from the perspective of end-to-end delay [28, 32, 33]. From the design perspective, rate control techniques can usually be classified into two categories: heuristic model-based [30, 34–36] and Lagrange multiplier-based [31, 37–39]. From the transport perspective, according to [13], rate control can be classified into three categories, namely, source-based, receiver-based, and hybrid rate control. Readers can refer to [13] for details. In this dissertation, since we focus on error control, we have not designed our own frame-level rate controller. In order to evaluate the performance of our proposed system, we can either adopt some rate controller, e.g., Test Model 5 (TM5) [30], or omit it and fix the bit budget or delay constraint for each frame at the same level.

2.1.4 Distortion Measurement

All the video compression standards discussed above are based on lossy compression. In this context, video quality should be evaluated in terms of the difference between the original picture and the reconstructed one in the decoder. Generally speaking, human vision system should be taken into account in the metric. The development of perceptually relevant image quality metrics is an active field of research [40, 41]. However the design of perceptual metrics for video is even harder because video is usually much more complicated than still images [42]. The mean squared error (MSE) and the Peak Signal-to-Noise Ratio (PSNR) are used for reporting results in most of the image and video processing literature. In the case of an image $f(m, n)$ of dimension $M \times N$ and its corresponding reconstruction $\hat{f}(m, n)$, the

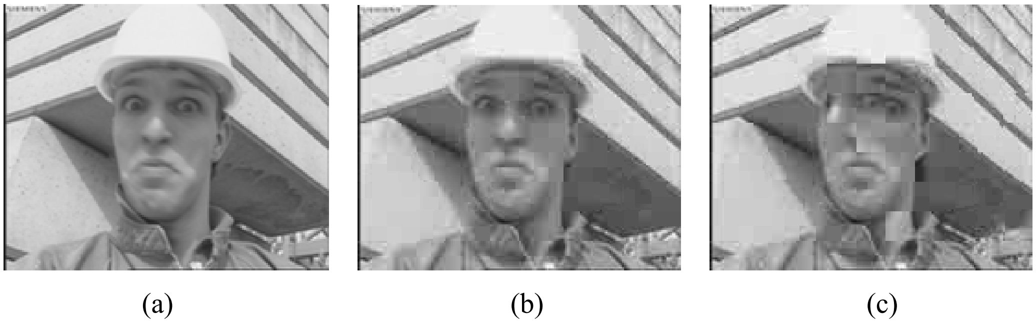


Figure 2.4: (a) Original frame (b) Reconstructed frame in the encoder (c) Reconstructed frame in the decoder (QCIF Foreman sequence, frame 92).

MSE is defined as,

$$\mathrm{MSE} = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} [f(m,n) - \hat{f}(m,n)]^2. \tag{2.4}$$

PSNR is inversely related to MSE as shown below

$$\mathrm{PSNR(dB)} = 10 \log \frac{255^2}{\mathrm{MSE}}, \tag{2.5}$$

where 255 is the maximum intensity value of a pixel for an 8-bit image.

Although these metrics are defined without reference to any human perception model, it has been reported [11] that compression systems optimized for MSE performance also yield good perceptual quality. Note that the framework presented in this dissertation can adapt to any video quality metric.

In a error prone channel, the reconstructed images at the decoder usually differ from those at the encoder due to packet losses, as shown in Fig. 2.4. In this case, instead of using the distortion between the original image and the reconstructed image at the encoder to evaluate the video quality, we consider the end-to-end distortion. That is, we measure the distortion between the original image, $f^{(i)}$, and the reconstructed image at the decoder, $\tilde{f}^{(i)}$, where the superscript (i) denotes the frame index. The expectation is calculated with respect to the probability of packet loss. Specifically, by omitting the frame index, we calculate the end-to-end distortion for packet k as

$$E[D_k] = \frac{1}{K} \sum_{j=1}^{K} [f_j - \tilde{f}_j]^2, \tag{2.6}$$

where the subscript j denotes the pixels that belong to packet k, and K is the number of pixels in packet k. Alternatively, $E[D_k]$ can be written as

$$E[D_k] = (1 - \rho_k)E[D_{R,k}] + \rho_k E[D_{L,k}], \tag{2.7}$$

where $E[D_{R,k}]$ is the expected distortion when the packet is received correctly, $E[D_{L,k}]$ is the expected distortion when the packet is lost, and ρ_k is the probability of loss for the k-th packet. Due to channel losses and error propagation, the reference frames at the decoder and the encoder may not be the same. Thus, both $D_{L,k}$ and $D_{R,k}$ are random variables.

2.2 Communication Networks

In this section, we briefly introduce the major components of the current IP networks, and focus on the major functions provided by these components in supporting real-time video transmission. We first describe the network protocols at multiple layers, and then discuss the network interface. In the end, we introduce the channel models used in our simulations.

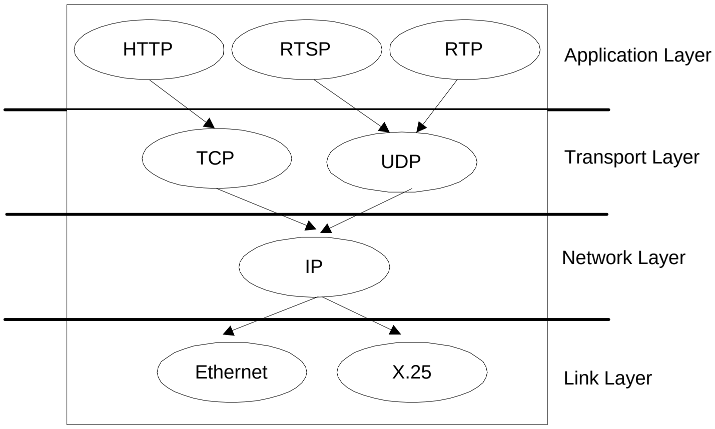


Figure 2.5: Illustration of protocol layers.

2.2.1 Network Protocols

Figure 2.5 shows the protocol stack used for an IP network. We focus on the multiple layers network protocols that are related to real-time video transmissions.

Internet Protocol

Internet Protocol is currently the most commonly used network layer protocol [43]. It provides connectionless delivery services, which means that each packet is routed separately and independently regardless of its source or destination. In addition, IP provides best-effort and thus unreliable delivery services. IP defines a set of standards for data transmission. First, it defines the format of the basic unit of data transmission that can be sent to the network. The normal size for an IP header is 20 bytes, plus options and padding. Furthermore, IP defines a set of rules that regulate the conditions in which a datagram can be discarded. It works with ICMP (Internet Control Message Protocol), which is responsible for generating error messages when an error occurs during transmission. The transport layer protocols, such as TCP or

UDP (User Datagram Protocol), deal with error packets based on the error message that they receive.

Transport Control Protocol

TCP is a connection-oriented and reliable service. It is full-duplex, and different connections between the same pair of hosts are differentiated by the port numbers being used [44].

TCP provides reliability by means of a window-based positive acknowledgement (ACK) with a go-back-N retransmission scheme. A window of small size is imposed on the data stream. Data within the window can be transmitted without acknowledgements. Once acknowledgements are received, the window slides across the data to incorporate new messages. The size of the sliding windows is varied in order to perform flow control on the data. If data is received out of sequence, it is resequenced by means of sequence numbers that are synchronized at the connection establishment. Duplicated segments are discarded. Note that TCP is one of the few transport-layer protocols that has its own congestion control mechanisms.

User Datagram Protocol

UDP is an alternative to the TCP in the transport layer, and, together with IP, is sometimes referred to as UDP/IP. Like TCP, UDP is connection-oriented. Unlike TCP, however, UDP does not provide reliable transmission. First, it does not provide sequencing of the packets as the data arrives. This means that the application that uses UDP must be able to ensure that the entire message has arrived and is in the right order. Second, UDP does not enforce retransmission of the lost packets. Nevertheless, TCP introduces unbounded delay due to persistent retransmission, which may not be suitable for applications that have strict delay constraint. For this reason, UDP is widely used in real-time video applications.

UDP also differs from TCP in that it does not have its own congestion control. Thus, when UDP/IP is used, additional congestion control needs to be deployed on top of UDP to constraint the bit rate for the applications. This function is discussed in detail in the next section. We also emphasize here that UDP is suitable for video applications not only because of the strict delay constraint, but also due to the QoS requirements. Compared with traditional elastic applications such as web browsing and electronic mails, real-time video applications are much more sensitive to delay but more tolerant to packet loss.

In addition, UDP has, although optionally, a checksum capability to verify that the data has arrived intact. Usually, only intact packets are forwarded to the application layer. This makes sense for wired IP networks, where entire packets may be lost due to buffer overflow. However, packets present bit errors in a wireless IP network. In this case, those packets with bit errors may still be useful for applications. For example, if the bit errors occur at the DCT coefficients part and the motion vectors are intact, these motion vectors would be helpful for either reconstructing the corresponding packet or concealing the neighboring packets. For this reason, modifications of the current UDP have been proposed and studied. One proposed study is called UDP Lite [45]. Note that in IP-based wireless networks, video packets can also be transported by UDP protocol [46].

Real-time Transport Protocol

RTP (Real-time Transport Protocol) runs on the top of UDP/IP for real-time applications. It provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast networks. It can be used for media-on-demand as well as interactive services. RTP consists of a data part and a control part. The latter is called RTCP (Realtime Transport Control Protocol) [47].

The data part of RTP is a thin protocol providing support for real-time applications, including timing reconstruction, loss detection, security and content identification. RTCP provides support for data delivery in a manner scalable to large multicast networks. This support includes time-stamping, sequence numbering, identification, as well as multicast-to-unicast translators. It offers QoS feedback from receivers to the multicast group as well as support for the synchronization of different media streams.

While UDP/IP is RTP's initial target networking environment, efforts have been made to make RTP transport-independent so that it can be used with other protocols. In addition, RTP does not address the issue of resource reservation or QoS control; instead, it relies on resource reservation protocols such as RSVP (Resource ReSerVation Protocol) [47].

Real-Time Streaming Protocol

The Real-Time Streaming Protocol (RTSP) is an application level protocol that establishes and controls one or more time-synchronized continuous media delivery with real-time constraints. According to the RFC 2326 [48], RTSP acts as a *network* *remote control* for multimedia servers. Controls include absolute positioning within the media stream, play, stop, pause, fast forward and possibly device control. One example of RTSP applications is RealPlayer.

RTSP does not depend on any specific transport mechanisms, although typically RTSP requests are sent using TCP. However, for real-time video and audio applications, RTSP can be used in conjunction with RTP/UDP.

2.2.2 Network Interface

The module between applications and networks, known as the network interface, consists of all the five major layers, namely, application layer, transport layer, network layer, link layer and physical layer. The main functionalities of the network interface are to take the compressed video bitstream, perform packetization, and send these packets to the receiver through the network by certain deadline imposed by applications, while meeting a constrained rate based on the estimation of network conditions. Common issues concerning the network interface include packetization, channel coding, retransmission, congestion control, packet loss detection, sender-driven or receiver-driven rate adaptation and etc [1]. We next discuss those functionalities, i.e., packetization, network condition estimation, and congestion control, in details.

Packetization

In the sender of a video transmission system, video packets (also termed as source packets) are generated by a video encoder. We refer to this stage as *source* *packetization*. In the application layer, source packets can be re-packetized into *intermediate packets* (e.g., for the reason of interleaving or FEC). Parity check packets used for FEC may also be generated in this layer. After source packets pass through the network protocol stack (e.g. RTP/UDP/IP), they form *transport packets* be sent over the network. The functionality of packetization that converts source packets to transport packets in the transport layer is referred to as *transport packetization*. In this section, we only discuss source packetization, since we employ the same source packetization scheme in all the work reported in this dissertation except for scalable video in Chapter 7. However, the transport packetization schemes used in different applications are different; detailed discussion for each application is left to the ensuing chapters. Next, we introduce the general rules for source packetization for real-time video applications.

First, each video packet should be independently encoded without using predictive coding across packets, so that each packet can be independently decoded, and packets introduce boundaries that can be used to limit the propagation of errors in the received bitstream. The RTP standard is based on the concept of Application Level Framing introduced in [49]. The idea is that packetization should take into account natural boundaries in the data set by the application layer. For video applications, general rules for generating packetization schemes to be used with RTP can be found in [50, 51]. Examples for specific video format can be found in [52] for MPEG-1/2 streams, [53] for MPEG-4, [54] for H.261, and [51] for H.264/AVC.

In setting packet size, the fragmentation limit of intermediated nodes on IP networks should be taken into account. In the Internet, the maximum transfer unit (MTU), which is the maximum size of a packet that can be transmitted without being split/recombined at the transport and network layer, is 1500 bytes. For wireless networks, the MTU size is typically much smaller and 100 bytes are commonly assumed in most research including JVT's wireless common conditions [51]. On one hand, it is not necessary to limit the size of source packets to the size limitation imposed by RTP, since big or small source packets can be fragmented or aggregated in the IP layer to adapt to the appropriate RTP packet size. However, if IP fragmentation is employed, protection efficiency will be lost, because the techniques based on UEP, such as data partitioning, are obviously not available. Additionally, the creation of one big source packet at the encoder end introduces coding efficiency through predictive coding across MBs. Such efficiency will be lost when small source packets are created at the encoder end[2] and aggregated in the IP layer to adapt to the RTP size limitation[3]. For the above reasons, translation of one source packet directly into one RTP packet is usually preferred in most real-time video applications.

In addition, in order to use RTP/UDP/IP protocols, a typical RTP packet in IP networks requires a header of approximately 40 bytes per packet [55]. To minimize packetization overhead, the size of the payload data should be substantially more than the header size. Therefore, considering the fragmentation limit of intermediated nodes on the IP networks, the conceivable lower and upper bounds for the payload data per packet over the Internet may be one row of MBs and one entire coded frame, respectively.

Following the above rules, in our simulations, we consider a packetization

[2] Note that predictive coding across source packets is prohibited.

[3] Note that source packets much smaller than the RTP size limit do not have to be aggregated in the IP layer. But aggregation improves efficiency when a fixed RTP/UDP/IP header has to be attached to each RTP packet.

scheme where each GOB (group of blocks)[4] is coded as one source packet, and every source packet is independently decoded. This may not be as efficient if larger packet size is chosen, but is highly robust to channel errors [56]. On the other hand, we choose packets with natural boundaries instead of fixed packet size to efficiently utilize error concealment. In the chosen packetization scheme, the decoder clearly knows the packet boundaries, and lost motion information can be easily estimated through the received neighboring packets. Note that the work in this dissertation can be extended to include other packetization schemes. The only requirement is that the packet boundaries are known *a priori*, i.e., which MBs are grouped into the same packet.

Network Monitoring

The network monitoring techniques can be classified into active/passive, on-demand/continuous, or centralized/distributed monitoring according to different classification criteria [7].

Channel State Information (CSI) is usually estimated through the feedback channel (e.g., using RTCP). Specifically, from the headers of the transmitted and feedback packets, the parameters, such as sequence number, time stamp, and packet size etc., can be collected at a regular time interval. Based on those parameters, CSI that are used for applications [such as packet loss ratio, network round-trip-time (RTT), retransmission timeout (RTO), and available network bandwidth] are estimated. One example is given in [8].

As for IP-based wireless networks, e.g., the 3G wireless networks, one of the

[4] Since in H.263 standard [20], one row of MBs is called GOB, in the following text, GOB and one row of MBs will be used interchangeably.

main services provided by the physical layer is the measurement of various radio link quantities, such as radio-link BER, channel signal-to-noise ratio (SNR), Doppler spectrum, and channel capacity. In order to facilitate the efficient support of QoS for video applications, these measurements are reported to the upper layer for channel state estimation from the perspective of link-layer or transport-layer. Translation of physical-layer channel measurements to the upper layer channel models is necessary because the physical-layer channel models do not explicitly characterize a wireless channel in terms of the necessary QoS required by the video applications, such as data rate and delay. One such scheme is presented in [46] to account for CSI estimation for video applications in using UDP. Another scheme is the link-layer channel model termed *effective capacity* (EC) developed in [57]. In this work a wireless link is modeled by two EC functions, namely, the probability of nonempty buffer and the QoS exponent of a connection; a simple and efficient algorithm is proposed to estimate those EC functions from the physical-layer channel model.

Congestion Control

Congestion control refers to the strategy employed to limit the sender's transmission rate to avoid the overwhelming of network resource by too much traffic. Congestion control is an indispensable component in most communication systems operating over best-effort networks. To transport media over the IP network efficiently, and to use the resource fairly, all IP service systems are expected to react to congestion by adapting their transmission rates. A good review of congestion control can be found in [13].

Congestion control used for multicast video is usually based on receiver-driven and hybrid techniques (combining both sender-driven and receiver-driven techniques) [58]. Receiver-driven techniques can further be classified into prob-based [59] and model-based [60]. Receiver-driven congestion control is typically applied to layered multicast video. Congestion control techniques used for unicast video, on the other hand, are usually sender-driven with two general categories: sender-based [58] and model-based [8]. We next discuss in detail sender-driven congestion control techniques used for unicast video, which is our focus in this dissertation.

Among the two approaches for sender-driven congestion control, the drawback of sender-based approach is its non-smooth transmission pattern. This is because it performs alternatively between additive rate increase and multiplicative rate decrease (AIMD), which may not be suitable to transmit continuous media. Thus, model-based TCP-friendly congestion control is usually recommended for video transmission over VBR (Variable Bit Rate) channels. In addition, since the Internet today is dominated by TCP traffic, it is very important for multimedia streaming to be "TCP-friendly", meaning that a media flow must generate similar flow throughput as TCP traffic under the same condition with lower latency.

One example of the stochastic TCP model can be used to estimate the network throughput, which represents the throughput of a TCP sender as a function of steady packet loss probability and RTT, as shown below [8, 61, 62].

$$R_T = \frac{\text{PacketSize}}{\mu_R\sqrt{2\varepsilon_R/3} + 3(\mu_R + 4\sigma_R)\epsilon_R(1 + 32\epsilon_R^2)\sqrt{3\epsilon_R/8}} \tag{2.8}$$

Where ϵ_R, μ_R, and σ_R^2 are short-term estimates of the packet loss probability, the mean RTT, and the variance of RTT, respectively. The formula (2.8) gives the upper bound of the sending rate R_T in bits per sec. The advantage of this approach is its TCP-friendliness and its simplicity. Certainly, there are other forms of model-based congestion controls. But any congestion control scheme will include some kind of channel estimation, e.g., estimation of channel parameters such as ϵ_R, μ_R, and σ_R^2.

To further smooth out the sending rate fluctuation to facilitate video applications, Wu *et al.* proposed a heuristic method adjusting the sending rate based on the estimated bandwidth, network congestion degree, etc. [8] By using that method, the sending rate can be increased or decreased very smoothly according to the network-related information. Similar methods of smooth and fast rate adaptation congestion control can be found in [63, 64].

In a DiffServ network, different QoS channel has different associated estimated parameters, thus different sending rates are available for different classes. Consequently, based on throughput estimation models, that will result in different transmission rate for different QoS classes.

As for wireless networks, one model is presented in [46] to estimate the UDP throughput R_T by assuming two-state Markov chain behavior of the success and failure of link-layer packets. In this model, R_T is dependent on transport-channel bit rate and the probability of successful UDP packet given that the previous packet was successful or failed, respectively. The relationship between packet transition probabilities and link-layer frame transition probabilities can be found in [46].

IP-Based Wireless Network Interface

Wireless IP networks are usually divided into two categories: indoor systems based on IEEE 801.11 Wireless LANs (WLANs) and outdoor systems based on the emerging 3G and 4G wireless networks [65]. Here, we highlight some major differences of wireless IP networks from the Internet, in terms of the functionalities provided to support real-time video communications.

It is well known that the application-layer or transport-layer ARQ is usually not suitable for real-time video transmissions. But the MAC (Media Access Control) layer retransmission can react to changing channels faster and lead to smaller delay than its upper layers. Therefore, the MAC layer retransmission might be applicable for real-time video transmission. In IEEE 802.11 WLAN, the maximum number of MAC layer retransmissions, i.e., the retransmission limit, can be changed adaptively per packet to provide throughput, reliability, and delay tradeoffs. Note that FEC is not employed in the MAC layer of the current IEEE 802.11 WLAN. However, there are eight modulation and channel coding modes defined in the physical layer.

Retransmissions in the 3G and 4G systems, such as CDMA2000, are more complicated. In such systems, after an IP packet is generated by passing through the transport layer and network layer, it is fragmented into PDUs (Packet Data Unit) in the link layer. The link layer protocol is called RLC (Radio Link Control), which is a kind of Selective Repeat ARQ. RLC defines the frame length of PDU as 336 bits, which is the unit for retransmission. The key component here affecting performance is the behavior of the layer 2 ARQ protocol. In addition, FEC and interleaving (e.g., RCPC/CRC is used to provide error protection and check) may be provided in layer 1. The FEC rate and interleaving length may vary depending on the CSI, the available bandwidth, and the delay constraint. The PDU loss becomes visible to the upper layers after interleaving and FEC are processed by layer 1 [6]. In addition, similar to IEEE 802.11, retransmission can be realized in the MAC layer to better react to the changing channels. Thus, besides TCP, there are two levels of retransmission implemented in the link layer, which makes the performance evaluation of such systems very challenging [66].

Since the focus of this work is on the application layer, we will not go to details of the link-layer retransmission including MAC layer retransmission. Instead, we assume that the function of link-layer retransmissions are disabled to avoid introducing extra latency. We further assume that physical-layer or link-layer FEC parameters can be accessed by the application layer so that they can be jointly decided with application layer and transport layer.

2.2.3 Network Channel

In wired networks, channel errors are usually in two forms: packet loss and packet truncation. In wireless networks, besides packet loss and packet truncation, bit error is another common source of error. Packet loss and truncation usually come from network traffic and clock drift. Bit corruption is due to the noisy air channel [67]. This dissertation focuses on packet loss in the wired link and bit error in the wireless channel.

The network is modeled as an independent time-invariant packet erasure channel with random delays, as in [9,62]. As discussed in [67,68], the wireless channel can also be treated as a packet erasure channel at the IP level, as it is "seen" by the applications. This is because IP-based wireless networks typically operate using a 32-bit Ethernet (802.2) CRC, and all packets failing that CRC check are rejected [67, 68]. Thus, we assume that packets with errors are not forwarded to the multimedia application. In real-time video applications, a packet is also considered lost if it does not arrive at the decoder on time. Thus the packet loss probability is made up of two components: the packet loss probability in the network and the probability that the packet experiences excessive delay. Combining these two factors, the overall probability of loss for packet k is

$$\rho_k = \epsilon_k + (1 - \epsilon_k) P\{\Delta T_n(k) > \tau\}, \tag{2.9}$$

where ϵ_k is the probability of packet loss in the network, ΔT_n is the network delay for the packet k, and τ is the maximum allowable network delay for this packet.

Packet losses in the network can be modeled in various ways, e.g., a Bernoulli process, a 2-state or k-th order Markov chain, etc. [69] The delay at network links is randomly varying. There is considerable experimental evidence that the time distribution of packet arrivals follows a self-similar law where the underlying distributions are heavily-tailed rather than following a Poisson distribution [15, 70]. The network delay could be modeled, for example, by a shifted Gamma distribution with heavy tail [9, 62, 71]. In addition, we assume that the probability of packet loss due to congestion (but not due to delay) does not change within the transmission of one packet, and that the delay of each channel is independent identically distributed (i.i.d.) for each transmitted packet.

Internet

In our simulations, packet loss in the Internet is modeled by a Bernoulli process, i.e., each packet is independently lost with probability ϵ. For simplicity, by ignoring the heavy tail, the network delay in our simulations is modeled as a *shifted Gamma distribution* with rightward shift γ and parameters n and α, defined as

$$f(\tau|\text{received}) = \frac{\alpha}{\Gamma(n)} (\alpha(\tau - \gamma))^{(n-1)} e^{(-\alpha(\tau-\gamma))} \quad \text{for } \tau \geq \gamma, \tag{2.10}$$

where n is the number of routers, γ is the total end-to-end processing time, and α is the parameter of exponentially distributed waiting time in each router that is modeled as an M/M/1 queue. The short-term estimates of n, γ and α can be obtained by periodically estimating the mean and variance of the forward trip time with the use of a TCP-friendly protocol. For more details, see [8, 61, 62].

Wireless Channel

Compared to their wire-line counterparts, wireless channels exhibit higher bit error rates, typically have a smaller bandwidth, and experience multi-path fading and shadowing effects. At the IP level, by assuming constant network delay, we only consider packet loss due to unrecoverable bit error. In this setting, one network parameter that can be specified is the transmission power used in sending each packet. For a fixed transmission rate, increasing the transmission power will increase the received SNR and result in a smaller probability of packet loss. This relationship could be determined empirically or modeled analytically. For example, in [72], an analytical model based on the notion of outage capacity [73] is used. In this model, a packet is lost whenever the fading realization results in the channel having a capacity less than the transmission rate. Assuming a Rayleigh fading channel, the resulting probability of packet loss is given by

$$\epsilon_k = 1 - exp\left(\frac{1}{P_k S(\theta_k)}(2^{R/W} - 1)\right),$$

where R is the transmission rate (in source bits per sec), W is the bandwidth, and $S(\theta_k)$ is the normalized expected SNR given the fading level, θ_k. Another way to characterize channel state is to use bounds for the bit error rate with regard to a given modulation and coding scheme; for example, in [74, 75], a model based on the error probability of BPSK (Binary Phase Shift Keying) in a Rayleigh fading channel is used. In this dissertation, we employ the latter due to practical considerations.

Consider using uncoded BPSK modulation scheme over a flat Rayleigh fading channel plus an Additive White Gaussian Noise (AWGN) process. The BER, p_e, assuming ideal interleaving, can be expressed as

$$p_e = \frac{1}{2}(1 - \sqrt{\frac{aE_b}{N_0 + aE_b}}), \tag{2.11}$$

where E_b is the bit energy, N_0 is the noise power spectrum density, and a is the expected value of the square of the Rayleigh distributed channel gain [76]. Likewise, for an AWGN channel, the BER can be written as

$$p_e = Q(\sqrt{\frac{2E_b}{N_0}}), \tag{2.12}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{\frac{-x^2}{2}} dx$ [76]. Letting m be the symbol length in bits (for example, m=4 or 8) and independent bit errors, the symbol error probability can be written as $p_s = 1 - (1 - p_e)^m$.

Usually in wireless channel, video packets are protected by channel codes through redundant symbols within packets. A packet will be treated as lost if the corrupted symbol in this packet cannot be recovered. Assuming independent bit errors (i.e., the additive noise and fading are each i.i.d. and independent of each other), the loss probability for a transport packet in the wireless channel can be calculated as

$$\beta_k = 1 - (1 - p_b)^{B_k}, \tag{2.13}$$

where p_b is the BER after channel decoding. In our work on wireless video, we employ RCPC (Rate-Compatible Punctured Convolutional) codes to perform link-layer protection, since they are widely used in providing intra-packet FEC due to its great performance and flexible implementation of the RCPC encoder and decoder. Both the theoretical bounds and simulation method of BER for RCPC codes can be found in [77, 78]. Note that the probability of packet loss β_k is a function of transmission power level, source coding parameter, and the channel coding rate selected for this packet [since p_b is a function of the channel BER p_e and channel coding rate r_k, where p_e is calculated from (2.11) or (2.12), depending on which channel model is used].

Hybrid Channel

Now we consider calculating the probability of loss for a source packet in a hybrid wireless-wired network, which consist of both wired link and wireless link, as shown in Fig. 2.6. At the IP level, as in [79], the network can be modeled as the combination of two independent packet erasure channels: the wired part with loss rate α_k and the wireless part with loss rate β_k. The overall loss probability of transport packet k in the network is then equal to

$$\epsilon_k = \alpha_k + (1 - \alpha_k)\beta_k. \tag{2.14}$$

2.3 Error Control Techniques

Video applications typically exhibit much higher latencies than that of voice. This is due to the strong inter-dependencies among video streams introduced by motion compensation. Predictive video encoding algorithm is employed in motion compensation to achieve high compression by reducing temporal redundancies between successive frames. Furthermore, predictive coding is also employed to reduce

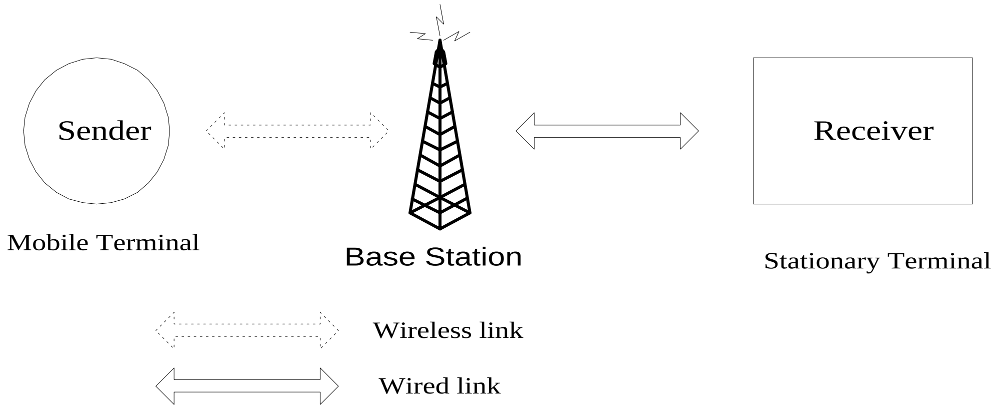


Figure 2.6: Black diagram of a hybrid wireless/wired network.

spatial statistical redundancies, as exampled by differential encoding of DC values of DCT coefficients, quantizer parameters, and motion vectors. Generally speaking, the higher the compression ratio, the greater the sensitivity of video stream to channel errors. Errors caused by motion information loss can propagate temporally and spatially due to the use of predictive coding, until prediction loop is restarted or synchronization is recovered [80]. For this reason, compressed video packets are sensitive to packet loss and are usually of different importance; thus, error control is critical in design consideration.

Error control generally includes error resilient resource coding, FEC, retransmission, and error concealment [13,80]. The structure of error control components in a video transmission system is illustrated in Fig. 2.7. Each of the above error control approaches is designed to deal with a lossy packet channel. Next we discuss each approach in detail.

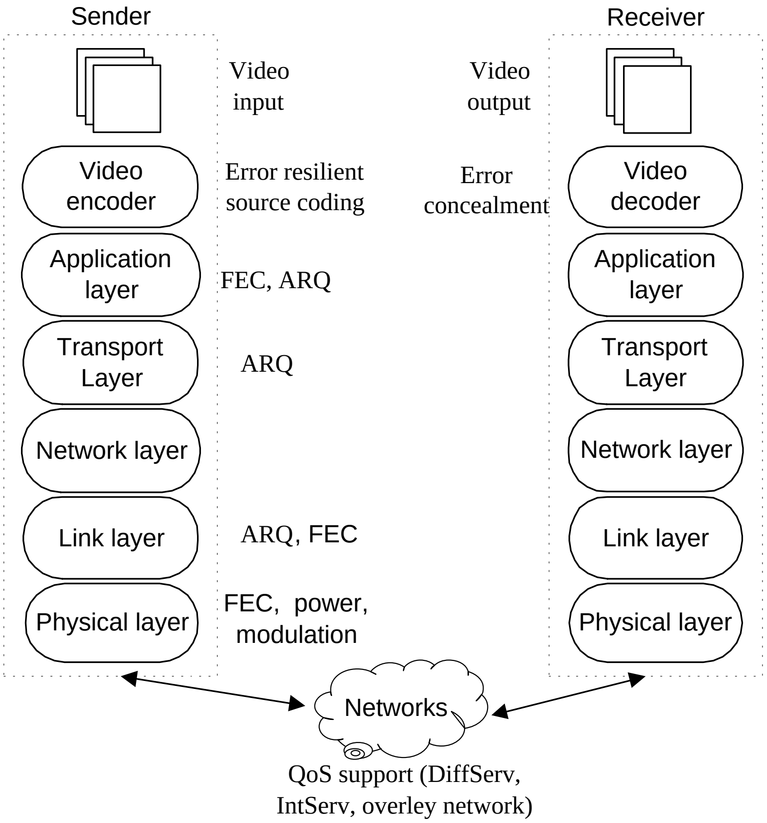


Figure 2.7: Illustration of error control components in video transmission system

2.3.1 Error Resilient Source Coding

Error resilient source coding refers to the technique of adding redundancy at the source coding level to prevent error propagation and limit the distortion caused by packet losses. This technique is usually composed of resynchronization marking, data partitioning and reversible variable-length coding (RVLC) for wireless video [56, 81, 82]. For packet-switched networks, error resilient source coding may include the encoding mode selection for each packet [81, 83–85], the use of scalable video coding [74, 86], or multiple description coding (MDC) [87, 88]. In addition, packet dependency control has been recognized as a powerful tool to increase error robustness. The common methods of packet dependency control are long-term memory (LTM) prediction for MBs, the reference picture selection (RPS), the intra-MB insertion, and video redundancy coding (VRC) [1].

Layered video coding produces a hierarchy of bitstreams, where the different parts of an encoded stream have unequal contributions to the overall quality. Layered coding has inherent error resilience benefits, especially if the layered property can be exploited in transmission, where, for example, available bandwidth is partitioned to provide UEP for different layers with different importance. This approach is commonly referred to as *layered coding with transport prioritization* [89]. In addition to the obvious benefits of scalability, layered coding with transport prioritization is one of the most popular and effective schemes for facilitating error resilience in a video transport system [55, 80].

MDC, another approach to improve error resilience [87], refers to a form of compression where a signal is coded into a number of separate bitstreams, each of which is referred to as a description. MDC has two important characteristics. First, each description can be decoded independently to give a usable reconstruction of the original signal. Second, combining more descriptions that are correctly received improves the decoded signal quality. A point worth mentioning is that each description is independent of each other and is typically of roughly equal importance. Thus, MDC does not require prioritized transmission. MDC will be beneficial if uncorrelated multiple paths are employed, since the use of multiple paths increases the chance of receiving at least a video of usable quality. The disadvantage of MDC is its low compression efficiency compared to conventional single description coding (SDC).

In contrast to the above work, we study the error resilient source coding problem by using optimal mode selection within a rate-distortion (R-D) optimization framework, as in [81, 84, 85, 90]. The gist of optimal mode selection method is to find the trade-off between coding efficiency and error robustness, since different prediction modes typically result in different levels of coding efficiency and robustness.

2.3.2 Forward Error Correction

Although error resilient source coding is a powerful tool to achieve robustness against packet loss, adaptation at the source cannot always overcome the large variations in channel condition and is also limited by the delay in the feedback as well as low level of accuracy in estimating the bottleneck bandwidth. Another way to deal with packet loss is to use error correction techniques by adding channel coding redundancy. Two basic techniques are used: FEC and Automatic Repeat reQuest (ARQ). Each has its own benefits in error robustness and network traffic load [17, 91]. Of the two error correction techniques, FEC is usually preferred for real-time video applications due to the strict delay requirements and semi-reliable nature of video streams [13, 56]. For this reason, FEC-based techniques are currently being considered by the Internet Engineering Task Force (IETF) as a proposed standard in supporting error resilience [92].

The FEC method used depends on the requirement of the system and the nature of the channel. FEC can usually be applied across packets (in the application or transport layer) and within packets (in the link layer) [93]. In applying inter-packet FEC, parity packets are usually generated in addition to source packets to perform cross-packet FEC, which is usually achieved by erasure codes. In the link layer, redundant bits are added within packet to perform intra-packet prediction from bit errors.

For Internet applications, many researchers have considered using erasure codes to recover packet losses [55, 94, 95]. In such approaches, a video stream is first partitioned into segments; each segment is packetized into a group of m packets. A block code is then applied to the m packets to generate additional l redundant packets (also called parity packets) resulting in a n-packet block, where $n = m + l$. With such a code, the receiver can recover the original m packets if a sufficient number of packets in the block are received. The most commonly studied erasure codes are Reed-Solomon (RS) codes, which have good erasure correcting properties and are widely used in practice [55, 94, 95]. Another class of erasure codes that have recently been considered for network applications are Tornado codes, which have slightly worse erasure protecting properties, but can be encoded and decoded much more efficiently than RS codes [89]. In this dissertation, we consider systematic RS codes, but the basic framework could easily be applied to other codes.

An RS code is represented as RS(n, m), where m is the number of source symbols and $(n - m)$ is the number of parity symbols. An RS code can be used to correct both errors and erasures, if an erasure occurs where the position of an error symbol is known. An RS(n, m) decoder can correct up to $(n - m)/2$ errors or up to $(n - m)$ erasures, regardless of which symbols are lost. The code rate of an RS(n, m) code is defined as m/n. For Internet applications, the channel errors are typically in the form of packet erasure, so an RS(n, m) code applied across packets can recover up to $(n - m)$ lost packets. Thus, with packet losses modeled by a Bernoulli random process, the block failure probability (i.e., the probability that at least one of the original m packets is in error) is

$$P_b(n,m) = 1 - \sum_{j=0}^{n-m} P(n,j) = 1 - \sum_{j=0}^{n-m} \binom{n}{j} \epsilon^j (1-\epsilon)^{n-j},$$

where ϵ is the probability of packet loss before error recovery, and $P(n,j)$ represents the probability of j errors out of n transmissions. The protection capability of an RS code depends on the block size and the code rate. These are limited by the extra delay introduced by FEC. The block length, n, can be determined based on the end-to-end system delay constraints [96].

As for wireless networks, channel coding is applied within each packet to provide protection. Source bits in a packet are first partitioned into m symbols, and then $(n-m)$ parity symbols are generated and added to the source bits to form a block. In this case, the noisy wireless channel causes symbol error within packets (but not erasure). As a result, the block error probability for an RS (n, m) code can be expressed as

$$P_b(n,m) = 1 - \sum_{j=0}^{(n-m)/2} P(n,j) = 1 - \sum_{j=0}^{(n-m)/2} \binom{n}{j} p_s^j (1-p_s)^{n-j},$$

where p_s is the symbol error rate. The packet loss probability is then $\epsilon = P_b(n,m)$. Note that ϵ is a function of the chosen quantizer and channel coding protection parameters for a packet, since the number of source symbols, m, depends on the source coding parameters selected for this packet.

Another popular type of code used to perform link-layer FEC is RCPC codes [93]. RCPC codes, first introduced in [77], are adopted in the level 3 of H.223 and H.324 annex C (mobile multiplexer), as a part of the mobile version of H.324 [97].

A family of RCPC codes is described by the mother code of rate $1/N$ and memory M with generator tap matrix of dimension N by $(M+1)$. Together with N, the puncturing period G determines the range of code rates as $R = G/(G+l)$ where l can vary between 1 and $(N-1)G$. The RCPC codes are punctured codes of the mother code with puncturing matrices $\boldsymbol{a}(l) = (a_{ij}(l))$ (of dimension $N \times G$) with $a_{ij}(l) \in (0,1)$ where 0 denotes puncturing.

The decoding of convolutional codes is most commonly achieved through the Viterbi algorithm, which is a maximum-likelihood sequence estimation algorithm. The Viterbi upper bound for the bit error probability is

$$p_b \leq \frac{1}{G} \sum_{d=d_{free}}^{\infty} c_d p_d$$

where d_{free} is the free distance of the convolutional code, p_d is the probability that the wrong path at distance d is selected, and c_d is the number of paths at Hamming distance d from the all-zero path. d_{free} and c_d are parameters of the convolutional code, while p_d depends on the type of decoding (soft or hard) and the channel. The theoretical bounds of BER for RCPC codes can be found in [77,78]. In the work of wireless video, we use the simulated BER. The method for simulation can be found in [74,77,78]. We consider using RCPC codes to perform link-layer FEC, but the proposed framework could easily be applied to RS or other codes as well.

2.3.3 Retransmission

For error correction, FEC is usually preferred for real-time video applications due to the strict delay requirements and semi-reliable nature of video streams [13,56]. However, FEC cannot completely avoid packet loss due to limits on the block-size dictated by the application's delay constraints. FEC also incurs constant overhead even when there are no losses in the channel. In addition, the appropriate level of FEC heavily depends on the accurate estimation of the channel's behavior. On the other hand, ARQ can automatically adapt to the channel loss characteristics by transmitting only as many redundant packets as are lost. Thus, if the application has a relatively loose end-to-end delay constraint (e.g., on-demand video streaming which can tolerate relatively large delay due to a large receiver buffer and long delay for playback), ARQ may be more applicable. Even for real-time applications, delay constrained application-layer ARQ has been shown to be useful for some situations such as in LAN, where RTT is relatively small [62, 91, 98, 99]. Various delay-constrained retransmission schemes for unicast and multicast have been discussed in [13].

2.3.4 Transmission Power Control

In wireless channels, besides FEC, through the use of transmitter power, the characteristics of the wireless channel as seen by the video encoder can be changed accordingly. Thus, prioritized transmission can be achieved through adjusting transmitter power for each packet. Specifically, for a fixed transmission rate[5], increasing the transmission power will increase bit energy and consequently decrease BER, as shown in (2.11) and (2.12). Conversely, for a fixed level of energy, increasing the transmission rate leads to higher BER but allows more data to be sent in a given time period. In addition to BER, the transmission rate affects transmission delay incurred by each packet. Furthermore, allocating different transmission power and transmission rate to the transmission of different packets results in different levels of

[5] Here we denote the transmission rate for the source in source bits per second, thus the corresponding BER is the source BER.

loss probability for different packets. Thus, in an energy-efficient wireless video transmission system, transmission power may need to be balanced against video delivery quality and delay to achieve the best video quality [75, 100, 101]. For example, in [75] a model was considered where the transmission power and transmission rate[6] were both adapted.

2.3.5 Network QoS Support

Error control can also be achieved through the QoS support from the network, since video packets with different importance levels can be transmitted with different QoS guarantees. Architectures supporting QoS have been under discussion for over a decade. Recently, two representative approaches have been proposed in IETF: the integrated service (IntServ) with the resource reservation protocol (RSVP) [102, 103] and the differentiated services (DiffServ or DS) [14, 104]. IntServ supports QoS by reserving resources for individual flow in the network. The main disadvantage of IntServ is that it does not scale well to large networks with thousands of reserved flows, where each router must maintain per-flow state information.

In contrast, DiffServ supports QoS by allocating resources discriminatorily to aggregated traffic flows based on multiple service classes [14, 15, 104]. Basically, the sender assigns a priority tag (a DS byte) to each packet, indicating the QoS class to which the packet belongs. Upon arriving at a router, a packet is queued and routed based on its assigned class. Consequently, the DiffServ approach allows different QoS to cater to different classes of aggregated traffic flows. Typically, a packet assigned to a high QoS class is less likely to be dropped or delayed at a router than a packet

[6] The source transmission rate was adapted by changing the amount of FEC applied to each packet using RCPC.

assigned to a low QoS class. These per hop behaviors lead to an end-to-end statistical differentiation between QoS classes [14, 15].

Aside from the approaches using QoS support to provide error control, path diversity is another architecture proposed to overcome network congestion and server overload [105]. When MDC is combined with path diversity, where the different descriptions are explicitly sent over different routes to a client, robustness can be achieved. Path diversity exploits the fact that losses on the two paths are likely to be uncorrelated. In another word, while any network link may suffer from packet loss, there is a much smaller chance that two network paths may simultaneously suffer from losses. Path diversity can be achieved using a relay infrastructure, a source-based routing, or content delivery network (CND) [105].

2.3.6 Error Concealment

Error concealment refers to post-processing technique employed by the decoder. Since human eyes can tolerate a certain degree of distortion in video signals, error concealment is a viable technique in handling packet loss. These methods can be broadly classified into spatial and temporal domain approaches [106]. In spatial approaches, missing data is reconstructed using neighboring spatial information, whereas in temporal approaches, the lost pixel is reconstructed from that in the previous frame. It was shown in [83] that temporal replacement usually results in lower perceptual distortion than spatial interpolation. Most temporal concealment techniques use temporal replacement based on the motion information of neighboring MBs [56]. These techniques attempt to estimate the missing motion information from neighboring spatial regions in order to perform motion compensation to conceal errors. Several estimates have been studied, e.g., using the average, median, and Maximum A Posteriori (MAP) estimates, or a side-matching criterion [106]. In [107], it was found that using the median estimate for motion compensation results in better subjective quality than the averaging technique in most cases. This is also the technique employed in the H.263 Test Model [108].

We consider a simple but efficient error concealment scheme similar to the ones in [55, 72]. In our simulations, we use temporal replacement error concealment strategy. The motion vector is spatially causal, i.e., the decoder will only use the information from previously received packets in concealing a lost packet. When a packet is lost, the concealment motion vector for an MB in the lost packet will be the median value of the three motion vectors of its top-left, top, and top-right MBs. If the previous packet is also lost, then the concealment motion vector is zero, i.e., the MB in the same spatial location in the previously reconstructed frame is used to conceal the current loss. For this concealment scheme, the expected distortion for the k-th packet, $E[D_k]$, can be described as

$$E[D_k] = (1-\rho_k)E[D_{R,k}] + \rho_k(1-\rho_{k-1})E[D_{C,k}] + \rho_k\rho_{k-1}E[D_{Z,k}] \tag{2.15}$$

where $E[D_{C,k}]$ and $E[D_{Z,k}]$ are the expected distortions after concealment when the previous packet is either received correctly or lost, respectively. In Chapter 3, we discuss how $E[D_{R,k}]$, $E[D_{L,k}]$, and $E[D_{Z,k}]$ are calculated in detail.

Chapter 3

Optimal Cross-Layer Resource Allocation

In this chapter, we present a resource-distortion optimization framework, which jointly considers source coding and various error control components at multiple network layers to achieve the best video quality. This framework forms the basis for the entire dissertation, i.e., the studies in the following chapters are various applications of this framework in different network infrastructures.

3.1 Introduction

An important aspect of communication networks is their dynamic behavior. In order to efficiently utilize limited network resources such as buffer, bandwidth, spectrum and energy, the end system needs to be adaptive to the changing network conditions. Adaptation represents the ability of network protocols and applications to observe and respond to the channel variations. Central to adaptation is the concept of cross-layer design [10]. The conventional layered protocol stack, where various protocol layers can only communicate with each other in a restricted manner, has proved to be inefficient and inflexible in adapting to the constantly changing network conditions. Cross-layer design of multimedia transmission aims to improve the system's overall performance by jointly considering multiple protocol layers.

For real-time video applications, cross-layer design may involve the video encoder and all the underlying network layers. The encoder can adjust its behavior, e.g., its flow rate or the amount of overhead devoted to error resilience, by selecting the source coding parameter for each video packet based on the changing network conditions. This technique is usually referred to as *error resilient source coding*. Adaptation can also take place in the underlying layers, such as the application layer and transport layer, e.g., by adding redundancy for forward error correction (FEC) or employing Automatic Repeat reQurest (ARQ) to retransmit lost packets [109]. For wireless networks, ARQ can also operate in the link layer, and FEC and modulation modes in the physical layer, e.g., the pan-European GSM system [110] and IEEE 802.11a [5, 111]. Information derived from the application, such as its QoS requirements and the priorities of the packets it produces, can be used in coordinating the behavior of the lower layers to increase resource utilization efficiency. Specifically, the amount of overhead devoted to FEC, the persistence level of the link-layer or MAC-layer ARQ mechanism, the transmitter power, transmission rates, and modulation modes should be adapted according to each application's latency and reliability requirements, as well as the traffic load. Note that the efficiency of adaptation heavily depends on the accuracy of CSI estimate.

Rather than designing source and channel coding separately, a recent trend in video coding research is to do joint source-channel coding. The main reason behind it is that Shannon's separation theorem does not strictly hold due to the delay and complexity constraints [1, 12, 13]. Joint source-channel coding has been extensively studied in the literature [13, 46, 74, 81, 86, 95, 112–118]. A more general framework of JSCC would be joint source-network coding (JSNC) in the study of video transmission over networks. JSNC requires the source coding to be adapted to the complicated network conditions through the interaction of network layers.

This dissertation is devoted to the field of cross-layer resource allocation for real-time video transmission applications, where the emphasis is on the interactions between different network layers, so as to improve the performance of video delivery given the resource constraints. Our focus is on the end-system design. We assume that the lower layer provides a set of given adaptation components. Our goal is to specify how the end-system should use these components to jointly allocate resources. What types of components are available depends on the specific application and network infrastructure. For example, for video transmission over wireless channels, transmission power, modulation or transmission rate in the physical layer may be specified in the application layer. For Internet-based video transmission, FEC priority could be decided in the application layer with regard to which source packet should be given higher protection. For video transmission over a DiffServ network, the packet transport priority could be adapted in the application to achieve priority QoS for video packets that are of different importance. All those cross-layer resource allocation problems can be accommodated in the proposed resource-distortion optimization framework.

The rest of this chapter is organized as follows. In Sect. 3.2 we present the resource-distortion optimization framework. We then briefly discuss related work of the area of cross-layer resource allocation for video transmission systems in Sect. 3.3. A key component in the optimization framework is the metric used for evaluating video quality. In Sect. 3.4 we review approaches for distortion estimation and characterization in packet-based video communication systems. In order to show the advantage of the framework, in Sect. 3.5, we give an example of joint source-channel coding for real-time Internet video transmission, namely, integrated joint source-channel coding (IJSCC) framework. We show by both analysis and simulations the advantage of the IJSCC framework in comparison with a sequential JSCC approach.

3.2 Resource-Distortion Optimization Framework

Let $\mathcal{Q}$ be the set of the source coding parameters, which include the prediction mode and quantization step size. The network resource parameter set is defined as $\mathcal{R}$. Let $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ denote the vector of source coding parameters and network resource parameters for one frame, respectively. The formulation of resource-distortion optimized joint source-network coding can be formally written as,

$$\begin{aligned} \min_{\{\boldsymbol{\mu}\in\mathcal{Q},\boldsymbol{\nu}\in\mathcal{R}\}} \quad & E[D(\boldsymbol{\mu},\boldsymbol{\nu})] \\ \text{s.t.} \quad & C(\boldsymbol{\mu},\boldsymbol{\nu}) \le C_0 \\ & T(\boldsymbol{\mu},\boldsymbol{\nu}) \le T_0, \end{aligned} \tag{3.1}$$

where $E[D]$ is the total expected distortion for one frame, and C_0 and T_0 are the cost constraint and transmission delay constraint, respectively, for that frame. The cost constraint C_0 is usually explicitly determined by specific application. For example, for the application of transmitting video from a mobile device to the base station, energy constraint comes from the battery life of the mobile device. For the application of DiffServ-based video transmission, the cost constraint comes from the negotiation of the user and the ISP (Internet Service Provider) through the SLA (Service Level Agreement). Unlike the cost constraint, the transmission delay constraint T_0 is more implicitly determined by the applications. For the applications with very short end-to-end delay constraints such as videoconferencing, the initial setup time T_{max} is usually very small (i.e., there is little additional buffering time in the receiver). For such applications, T_0 is very strict and usually around one frame's time, $1/F$, where F is the frame rate. However, for the applications that have relatively loose end-to-end delay such as on-demand video streaming, T_{max} is generally much longer than one frame's time, thank to the additional buffering at the receiver. In this case, T_0 is not that strict and usually varies around $1/F$ according to the video content (e.g., complicated frames usually need more bits to encode, or, alternatively, more transmission delay to transmit), the dynamics of the encoder buffer and the playback buffer at the receiver. The determination of T_0 for a video group is achieved by rate control. Because rate control is usually separately designed from error control, we skip this component for simplicity in this dissertation. However, we recognize that rate control is a very important component in the overall end-system design.

The proposed resource-distortion optimization framework is general and can be used for the joint consideration of error resilient source coding, error concealment and cross-layer resource allocations[1]. We emphasize that in this framework, all the

[1] These cross-layer resource allocations are limited to those layers whose resource allocation parameters can be specified and controlled in the application layer. This usually requires new protocols that enable the application layer to specify those adaptation components at the lower layers.

available error control components are fully integrated. In this chapter, we focus on the discussion of the framework itself. The next three chapters are devoted to three special cases of cross-layer resource allocations in different network infrastructures.

3.3 Related Work

Cross-layer resource allocation for video transmission systems is an active and growing field of research. In order to give the readers a big picture of the work in this area, we briefly highlight the related work of each adaptation scenario for real-time video applications herein. More detailed discussion of the development in each direction is given in the ensuing chapters.

Error resilient source coding: Error resilient source coding has been studied for a long time. Earlier work on this packet dependency control regime includes long-term memory (LTM) prediction for MBs for increased error-resilience [119], the reference picture selection (RPS) mode in H.263+ standard [21] and the emerging H.264/AVC standard [26], the redundancy coding (VRC) technique (intra-MB insertion) in H.263+ standard [108, 120] and MPEG-4 standard [35], and the Intra refreshment and synchronization algorithm [121]. In [38,81,83–85,90,122] this problem has been studied in a rate-distortion optimization framework, where the optimization is achieved through source coding mode selection for each MB. In the above work, the goal is to select the source coding mode for each MB by taking into account the probability of packet loss in the channel and the error concealment technique used by the decoder in order to reduce the expected distortion at the receiver. Another pool of literature targeting at bit-based channels, such as resynchronization marking, data partitioning, and RVLC [81], is beyond the scope the discussion here. For a review of such work, readers can refer to [56, 82]

Joint source-channel coding: Similar to error resilient source coding, joint source-channel coding has been extensively studied in literature [55, 74, 112–114, 123]. In general, JSCC is accomplished by designing the quantizer and entropy coder for given channel errors, such as the work in [112]. For image and video signals, JSCC focuses on the optimal bit allocation between source coding and channel coding given channel loss characteristics [13]. In [113, 116], JSCC is studied for wavelet image transmission over the Internet, where the source and channel coding bits are allocated to minimize the expected end-to-end distortion. In [55], JSCC for Internet scalable video is studied, where error resilient source coding and FEC are jointly considered. The similar problem of scalable video transmission over bit-based channels is studied in [123] based on a 3D scalable codec and [74] based on the H.263+ codec. Our recent work in [86] studied this problem by jointly considering error resilient source coding and FEC based on scalable video. We also take ARQ into account in the study in [124] for non-scalable video.

Joint source coding and packet scheduling: The problem of packet scheduling for video transmission has been studied in [18, 62, 101, 125–127]. The goal of [125, 126] is to minimize the total amount of transmission energy while meeting the delay constraints. In [101, 127], this problem is studied by jointly considering both the physical layer power control and scheduling along the adaptation of source coding parameters. Recent work by Chou *et al.* provides a flexible framework to allow rate-distortion control of packet transmissions [18, 62]. In that work, the video streaming system can allocate time and bandwidth resources among packets in a way that minimizes a Lagrangian cost function of expected rate and distortion based on the packets' deadlines, transmission histories, the channel characteristics, the packets' interdependencies, and the each packet's associated distortion reduction.

Joint source coding and packet classification: With the emerging architecture of DiffServ employed in the Internet to support QoS, the DiffServ-aware video streaming system has been of interest recently [9, 29, 128–131]. In [128], an adaptive packet forwarding mechanism is proposed for a DiffServ network with QoS accommodation by mapping video packets onto different DiffServ service levels. This framework does not incorporate video source coding. The authors in [129] proposed a rate-distortion optimized packet marking technique to deliver MPEG-2 video sequences (only INTRA frames were used in this work) in a DiffServ IP network. Their goal was to minimize the bandwidth consumption in the premium class while achieving nearly constant perceptual quality. This work was extended by taking into account inter-frame motion compensation in [130]. Neither [129] nor [130] considers the selection of source coding parameters. In [62] and [9], cost-distortion optimized multimedia streaming over DiffServ networks is studied. Although the proposed framework in [62] and [9] is very general, it is based on pre-encoded media. Thus the selection of encoding parameters is not considered, and neither is error concealment included. In [29], a similar problem is studied based on an assumption that the channel is modeled by a Bernoulli random process, in which packets are lost with some known constant probabilities. Distinguishable from the above studies, in our work [131], we have proposed a novel framework, which incorporates the packet delay management into the calculation of packet loss probability, with joint selection of encoding parameters and packet scheduling priority. Our study aims at minimizing the end-to-end distortion at given total cost and transmission delay constraints.

Joint source-channel coding and power adaptation: Besides channel coding, transmission power is another component that has been widely studied in the joint design with source coding to adapt to the changing wireless channel [72, 117, 127, 132–134]. Energy efficient wireless video transmission has been studied in [72]. The goal is to adjust the source coding parameters and the allocation of transmitter power in order to spend the minimal amount of energy necessary in transmitting a video sequence subject to an expected distortion and delay constraint. In [127], the selection of source coding parameters is jointly considered with transmitter power and rate adaptation, as well as packet transmission scheduling for energy efficient wireless video streaming. A joint source coding and power control approach is presented in [134] for optimally allocating source coding rate and bit energy normalized with respect to the multiple-access interference noise density in the context of 3G CDMA networks. The work in [134] did not address error resilient source coding and channel coding. Joint source-channel coding and transmission power allocation has been studied in [117] for progressive image transmission. A joint FEC and transmission power allocation scheme for layered video transmission over a multiple user CDMA network is proposed in [135] based on the 3D-SPIHT codec. In [46, 132] a regime of joint source channel coding with optimal power consumption is studied to transmit scalable video over a 3G wireless network, where the channel-adaptive hybrid scheme of UEP and delay constrained ARQ is proposed to achieve the minimal power assumption with distortion and bit rate constraint in [132]. An adaptive cross-layer protection scheme is presented in [111] for robust scalable video transmission over 802.11 wireless LANs, where application-layer FEC, the MAC (media access control) retransmission limit, and packet sizes are jointly considered.

3.4 End-to-End Distortion

In our simulations, the distortion measurement is based on per-pixel accurate distortion calculations, which ensure accurate estimation of the overall end-to-end distortion [83, 84, 90]. The expected distortion for the k-th packet can be calculated as the average of the expected distortions of all the pixels in this packet, as follows,

$$E[D_k] = \frac{1}{K} \sum_{i=1}^{K} E[d_i] \tag{3.2}$$

where d_i is the distortion of the i-th pixel, and K is the number of pixels of packet k.

3.4.1 ROPE Algorithm

Assuming the MSE criterion, the distortion measurement based on the ROPE algorithm (Recursive Optimal Per-pixel Estimate) [84] is used to calculate the overall expected distortion level of pixel i in frame n

$$E[d_i^{(n)}] = E[(f_i^{(n)} - \tilde{f}_i^{(n)})^2] = (f_i^{(n)})^2 - 2f_i^{(n)} E[\tilde{f}_i^{(n)}] + E[(\tilde{f}_i^{(n)})^2] \tag{3.3}$$

The parameters used in this subsection are defined as in Table 3.1.

The first and second order expected values of one pixel are recursively calculated using the ROPE algorithm. Their calculations depend on the specific packetization scheme, error concealment method used, and the pixel's prediction mode. For example, in using the packetization scheme and error concealment scheme descried in Chapter 2, $E[\tilde{f}_i^{(n)}]$ with INTRA, SKIP and INTER mode are defined as below. The

$f_i^{(n)}$	:	i-th pixel of the n-th original frame
$\tilde{f}_i^{(n)}$	:	i-th pixel of the n-th expected reconstructed frame
$\hat{f}_i^{(n)}$	:	i-th pixel of the n-th reconstructed frame
$\tilde{f}_l^{(n-1)}$	:	l-th pixel of the (n-1)-th expected reconstructed frame, which is appointed by the decoded motion vector
$\tilde{f}_j^{(n-1)}$	:	j-th pixel of the (n-1)-th expected reconstructed frame, which is appointed by the concealment motion vector
$d_i^{(n)}$	:	distortion of the i-th pixel of the n-th frame
$\hat{e}_i^{(n)}$	:	i-th quantized residue of the n-th frame

Table 3.1: Notations used in the ROPE algorithm.

second-order expected value, $E[(\tilde{f}_i^{(n)})^2]$, can be defined in the similar fashion.

$$\text{INTRA: } E[\tilde{f}_i^{(n)}] = (1-\rho_k)E[\hat{f}_i^{(n)}] + \rho_k(1-\rho_{k-1})E[\tilde{f}_j^{(n-1)}] + \rho_k\rho_{k-1}E[\tilde{f}_i^{(n-1)}] \tag{3.4}$$

$$\text{SKIP: } E[\tilde{f}_i^{(n)}] = (1-\rho_k)E[\tilde{f}_i^{(n-1)}] + \rho_k(1-\rho_{k-1})E[\tilde{f}_j^{(n-1)}] + \rho_k\rho_{k-1}E[\tilde{f}_i^{(n-1)}] \tag{3.5}$$

$$\text{INTER: } E[\tilde{f}_i^{(n)}] = (1-\rho_k)(\hat{e}_i^{(n)} + E[\tilde{f}_l^{(n-1)}]) + \rho_k(1-\rho_{k-1})E[\tilde{f}_j^{(n-1)}] + \rho_k\rho_{k-1}E[\tilde{f}_i^{(n-1)}] \tag{3.6}$$

Here we emphasize that the inter-frame error propagation due to channel errors has been captured in this distortion characterization. This is because in order to calculate the expected distortion of the current frame, all that needs to be determined is the first order and second order information of each pixel's values in the previous frame due to the nature of motion compensation. For the study of the accuracy of this distortion measurement, readers can refer to the original paper [84].

We next consider an extension of the original ROPE algorithm to the scenario where channel feedback is available such that which packet is lost or received is known to the encoder.

3.4.2 Distortion Estimation Based on Feedbacks

Although the distortion for packet k, D_k, is usually a random variable, based on the given feedback information, it could be deterministic. This occurs when packet k and all the associated packets in the previous frames serving as its prediction are acknowledged and have not been further retransmitted. Thus, the expectations in (2.7) need to be re-calculated based on the updated probability distribution of channel losses given the available feedback. For example, if one packet is known to have been received, its probability of loss becomes 0; if one is lost, its loss probability becomes 1 if no further retransmission for this packet has been initiated. Based on the updated probabilities of packet loss, the expected distortion of all packets in the encoder buffer is recursively re-calculated as in (2.7). For simplicity, assume that the RTT is constant. In using the extended ROPE model like this, the error propagation due to packet loss (after 1 RTT) can be fully captured and, as a consequence, the effect of previously lost packets on the future frames is taken into account. This can be shown in Fig. 3.1, where frame $n-2$ has been acknowledged so that the distortions up to frame $n-2$ are all deterministic and those in the following frames are still random variables. If retransmission is allowed, distortion estimation will become more complicated because the retransmission should be accounted for in the calculation of the updated probability distribution of packet losses. This is discussed in more detail in the next chapter.

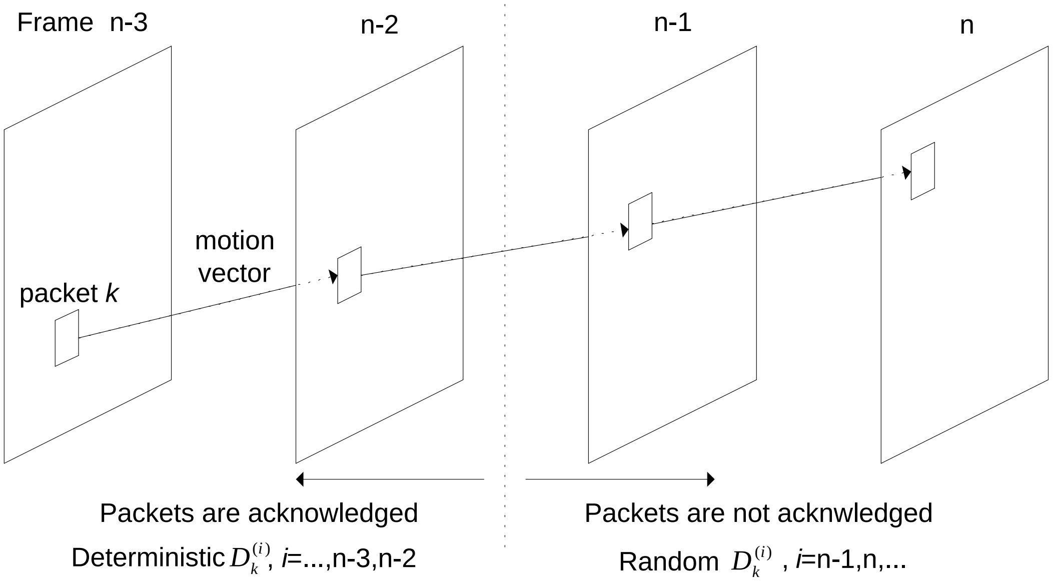

Frame n-3
n-2
n-1
n
motion vector
packet k
Packets are acknowledged
Deterministic $D_k^{(i)}$, i=...,n-3,n-2
Packets are not acknwledged
Random $D_k^{(i)}$, i=n-1,n,...

Figure 3.1: Illustration of feedback effect on the distortion calculation.

3.5 Joint Source-Channel Coding

In this section, we analyze in detail the motivation of the proposed resource-distortion optimization framework (3.1) and the reason why we claim that it is a fully integrated framework for the study of joint source-network coding. As a special case of joint source-network coding, we use joint source-channel coding as an example to study the key properties of this framework. We name our framework for this special case as integrated joint source-channel coding (IJSCC).

The basic idea of JSCC is illustrated in Fig. 3.2. When the channel is error free, increasing the bit rate leads to decreasing distortion, as in standard R-D theory [11,136]; this is shown by the lowest curve in Fig. 3.2. However, when channel errors are present, this trend may not hold, since the overall distortion contains both source distortion and channel distortion. As more bits are allocated to source coding, fewer will be left for channel coding, which leads to less protection and higher channel distortion. As shown in Fig. 3.2, an optimal bit allocation exists between source and channel coding. Note that different channel error rates result in different optimal allocations. This is indicated by the points (R2, D2) and (R3, D3) on the two curves with different channel error rates.

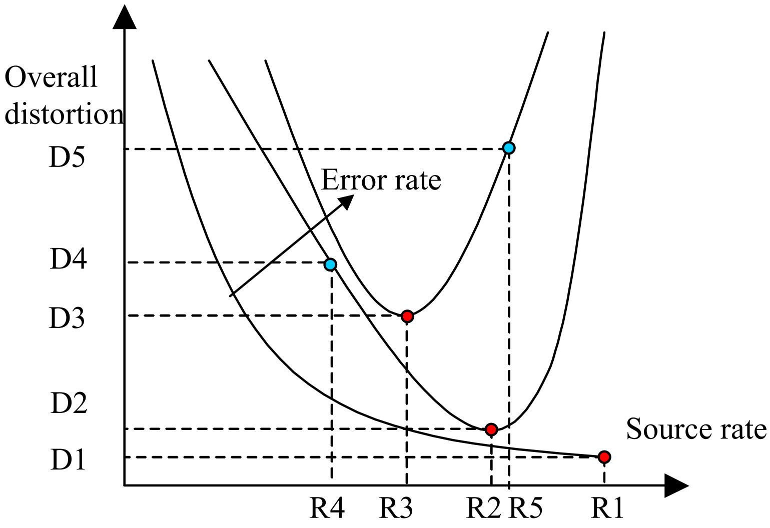


Figure 3.2: Illustration of joint source-channel coding.

For image and video applications, JSCC has three tasks: finding an optimal bit allocation between source coding and channel coding at given channel loss characteristics; designing the source coding to achieve the target source rate; and designing the channel coding to achieve the required robustness [85].

Most of the JSCC work to date has focused on the bit allocation between source and channel coding, such as in [46,74,94,115,117]. Source coding is performed based on the given bit budget, after the bit allocation between source and channel is completed. The optimization of source coding can be achieved in the form of mode selection by taking into account the residual packet loss rate after channel coding, such as in [55, 83, 84]. We argue that the above studies, however, do not fully consider the interaction between source coding and channel coding. More specifically, they do not take into account how error resilient source coding affects the bit allocation between source and channel. In this work, we consider the IJSCC framework, where error resilient source coding, channel coding, and error concealment are jointly considered in a tractable optimization setting.

In this section, we discuss approaches for jointly optimizing error resilient source coding and channel coding. First, we discuss approaches that sequentially achieve optimal bit allocation between source and channel coding, and then optimize the source coding given the resulting bit budget. Next we present our IJSCC framework which optimizes both the bit allocation and source coding in a single step.

3.5.1 Sequential Joint Source-Channel Coding

Let $\mathcal{Q}$ be the set of source coding parameters, and the FEC parameter set is defined as $\mathcal{R} = \{(N_1, M), ..., (N_q, M)\}$, where q is the number of available code options. Let the superscript (n) denote the frame index, and the subscripts s and c stand for source and channel, respectively. The sequential two-step JSCC can then be formally presented as

$$\begin{aligned} &\min_{\{\boldsymbol{\nu}\in\mathcal{R}\}} E[D^{(n)}(\boldsymbol{\nu})] \\ \text{s.t. } T^{(n)}(\boldsymbol{\nu}) =& B_s^{(n)}(\boldsymbol{\mu}(\boldsymbol{\nu}))/R_T + B_c^{(n)}(\boldsymbol{\nu})/R_T \leq T_0^{(n)}, \end{aligned} \tag{3.7}$$

and

$$\begin{aligned} &\min_{\{\boldsymbol{\mu}\in\mathcal{Q}\}} E[D^{(n)}(\boldsymbol{\mu})] \\ &\text{s.t. } T_s^{(n)}(\boldsymbol{\mu}) = B_s^{(n)}(\boldsymbol{\mu})/R_T \leq T_{s,0}^{(n)}, \end{aligned} \tag{3.8}$$

where $E[D]$ is the expected distortion; R_T is the transmission rate; B_s and B_c are the source bits and channel bits, respectively; T is the associated transmission delay; T_0 and $T_{s,0}$ are the transmission delay constraint for the whole frame (including both source and channel bits) and the source bits, respectively. In (3.7), the constraint is on the total transmission delay for the n-th frame, $T^{(n)}$; in (3.8), the constraint is on the source transmission delay[2], $T_s^{(n)}$. Several channel coding techniques have been considered for solving (3.7). For work utilizing pre-encoded video, such as [94, 95], source coding is fixed. Thus, the objective in these works is to minimize the channel induced distortion, and the second step (3.8) is not necessary. For work on coding the source on the fly, one way to characterize the distortion in (3.7) is to use a source R-D model, as in [74, 115]. For example, a universal R-D model is used in [74]. In [115], the distortion is expressed as the sum of source and channel distortion, both of which are model-based. By assuming uncorrelated source and channel distortion, the first-step of the minimization in [115] aims at minimizing the channel distortion, while the second-step minimizes the source distortion. There has also been considerable work in the area of JSCC for wavelet-based scalable video coders, such as [113, 116, 135]. The inherent prioritization of information in a wavelet-based video bitstream makes the implementation of JSCC more straightforward. For block-based motion compensated video coding, JSCC is more challenging because the relative importance of packets is not explicitly available.

The above studies, however, do not optimally account for how error resilient source coding affects the bit allocation between source and channel. The goal of JSCC is to optimally add redundant bits in the source (error resilience source coding) and the

[2] Note both of these constraints can also be interpreted as specifying bit budgets of $T_0^{(n)} R_T$ and $T_{s,0}^{(n)} R_T$.

channel (channel coding) to achieve the best trade-off between error robustness and compression efficiency. The optimal way to achieve this requires jointly considering error resilient source coding and channel coding. It is clear that such an integrated approach should be superior to the sequential approach in (3.7) and (3.8).

3.5.2 Integrated Joint Source-Channel Coding

Next we present our IJSCC framework for jointly optimizing error resilient source coding and channel coding. That is, instead of separating the overall expected distortion into source distortion and channel distortion, as in (3.7) and (3.8), we consider the interaction between these components.

A related framework is presented in [55] for jointly considering error resilient source coding and channel coding. In that work, the distortion measurement is model-based, where the concealment distortion for each block is calculated by weighting the distortions of the surrounding blocks in the previous frame(s) that overlap with the motion-compensated block (with the weights proportional to the overlap area). Here, we recursively calculate packet distortion based on (2.7), which takes into account both source distortion and channel distortion, as well as error propagation due to channel errors[3]. Our objective is to minimize the total expected distortion for the n-th frame, given a transmission delay constraint, i.e.,

$$\begin{aligned} &\min_{\{\boldsymbol{\mu}\in\mathcal{Q},\boldsymbol{\nu}\in\mathcal{R}\}} E[D^{(n)}(\boldsymbol{\mu},\boldsymbol{\nu})] \\ &\text{s.t.} \quad T^{(n)}(\boldsymbol{\mu},\boldsymbol{\nu}) = B^{(n)}(\boldsymbol{\mu},\boldsymbol{\nu})/R_T \leq T_0^{(n)}, \end{aligned} \tag{3.9}$$

where $B^{(n)}$ represents the total bits used for both source coding and channel coding,

[3] The effect of error propagation can be fully captured based on the acknowledgement information after 1 RTT's delay.

and $T_0^{(n)}$ is the transmission delay constraint for this frame.

In calculating the expected distortion for each source packet, the loss probability for the source packet ρ needs to be determined. The relationship between the source packet loss probability and transport packet loss probability depends on the specific transport packetization scheme and channel coding chosen [118].

3.5.3 Solution Algorithm

By using a Lagrange multiplier $\lambda \geq 0$, (3.9) can be converted into an unconstrained problem as,

$$\min_{\{\boldsymbol{\mu}\in\mathcal{Q},\boldsymbol{\nu}\in\mathcal{R}\}} \sum_{k=1}^{M} J_k^{(n)} = \sum_{k=1}^{M} E[D_k^{(n)}] + \lambda \sum_{k=1}^{M} T_k^{(n)}, \tag{3.10}$$

where M is the number of packets in the frame. Note that the transmission delay for the k-th packet, $T_k^{(n)} = B_k^{(n)}/R_T$, takes into account the associated channel bits used to protect this packet. The convex hull solution of this relaxed problem can be found by choosing an appropriate λ to satisfy the transmission delay constraint. This can be done using standard techniques such as a bisection search [136]. We can write the problem as:

$$\min_{\{\boldsymbol{\mu}\in\mathcal{Q},\boldsymbol{\nu}\in\mathcal{R}\}} \sum_{k=1}^{M} J_k^{(n)} = \min_{\{\boldsymbol{\nu}\in\mathcal{R}\}} \{ \min_{\{\boldsymbol{\mu}\in\mathcal{Q}\}} \sum_{k=1}^{M} J_k^{(n)}(\boldsymbol{\mu},\boldsymbol{\nu}) \}. \tag{3.11}$$

Given a specific λ, minimization of (3.11) can be divided into two steps: bit allocation for FEC and the optimal mode selection based on the remaining delay. Note that this differs from solving (3.7) and (3.8) in that the bit allocation for FEC takes into account the effect of this choice on source coding. The optimal mode selection can be found using a dynamic programming (DP) approach. The DP can be viewed as a shortest path problem in a trellis, where each stage corresponds to the mode selection for a given packet [131, 136]. Note that by using the error concealment strategy described in Sect. 2.3.6, the distortion $E[D_k]$ depends on the encoding modes selected for the previous source packet. Thus, the Lagrangian $\sum_{k=1}^{M} J_k^{(n)}(\boldsymbol{\mu}, \boldsymbol{\nu})$ in (3.11) is not separable. In this case, the time complexity is $O(|M \times |\mathcal{R}| \times |\mathcal{Q}|^2)$, where $|\cdot|$ denotes the cardinality of the set inside [136].

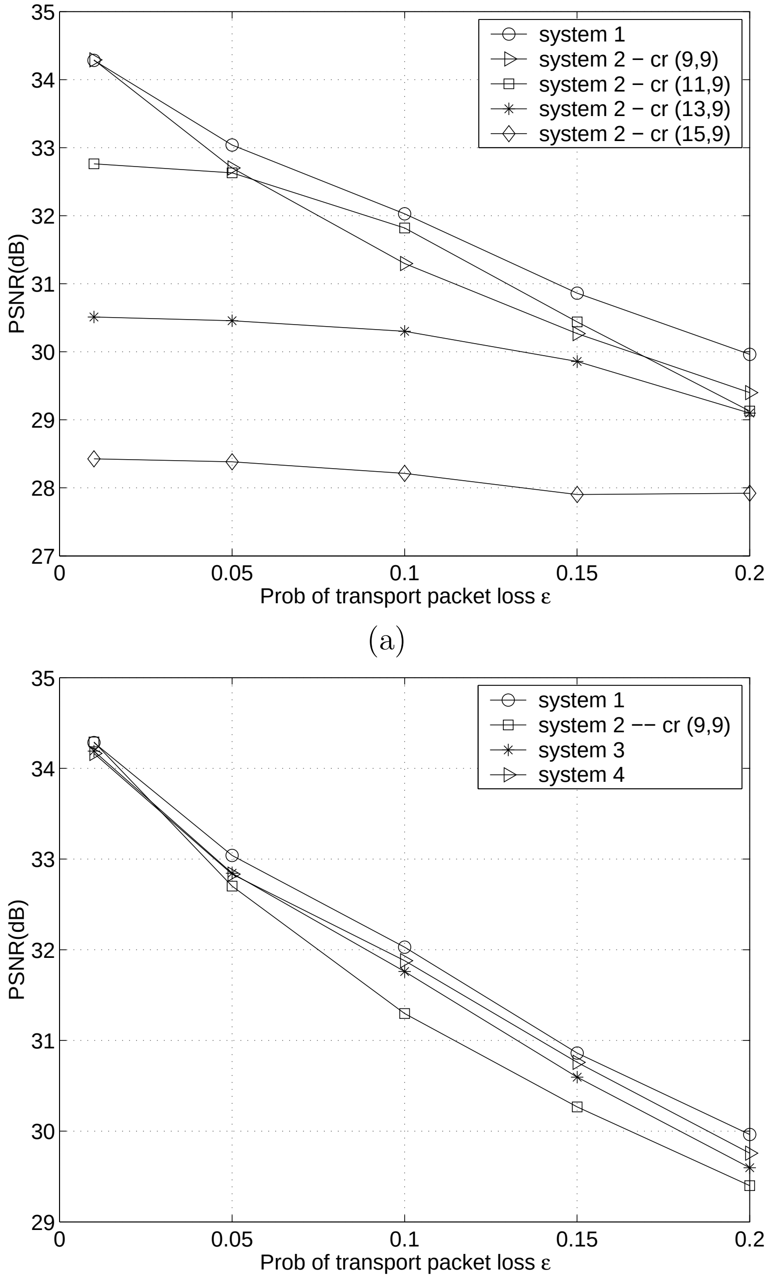
3.5.4 Experimental Results

In the simulations, we use an H.263+ codec [21] to perform source coding, and we consider the QCIF format (176×144) Foreman sequence. Rate control is not implemented in the video streaming system. Thus, every frame has the same transmission delay constraint of one frame's duration, i.e., $T_0^{(n)} = T_F$. We assume that after 1 RTT, channel feedback is available to the encoder in the form of which packets are received or lost. We consider applications that require a short end-to-end delay, T_{max}, and the RTT is set equal to two frames. Under that situation, the feedback delay is long enough to preclude retransmissions.

Four systems are compared: i) system 1, which uses the proposed framework to jointly consider error resilient source coding and channel coding; ii) system 2, which performs error resilient source coding, but with fixed rate channel coding; iii) system 3, which performs only channel coding, but no error resilient source coding (i.e., source coding is not adapted to the modified channel characteristics after error recovery); and iv) system 4, which performs sequential JSCC. All four systems are optimized in the following manners. System 2 performs optimal error resilient source coding to adapt to the channel errors (with fixed rate channel coding). System 3 selects the optimal channel coding rate to perform FEC and does optimal source coding (without considering residue packet loss after channel coding) at the given bit budget. In the sequential JSCC, channel coding and error resilient source coding are performed sequentially, i.e., bit allocation between source and channel is performed with no awareness of error resilient source coding as in (3.7), and error resilient source coding is performed thereafter given the bit budget as in (3.8).

We illustrate the performance of the four systems in Fig. 3.3 at $R_T = 480$ kbps and the frame rate $F = 30$ fps. Here, we plot the average PSNR against different packet loss rates. All four systems have the same transmission delay constraints and transmission rate. It can be seen in Fig. 3.3(a) that system 1 outperforms system 2 with different pre-selected channel coding rates. In addition, system 1 outperforms the optimized system 2 (the upper bound of system 2 with different pre-defined channel rates) with different channel coding rates by up to 0.4 dB. This is due to the flexibility of system 1 in varying the channel coding rate in response to the channel conditions.

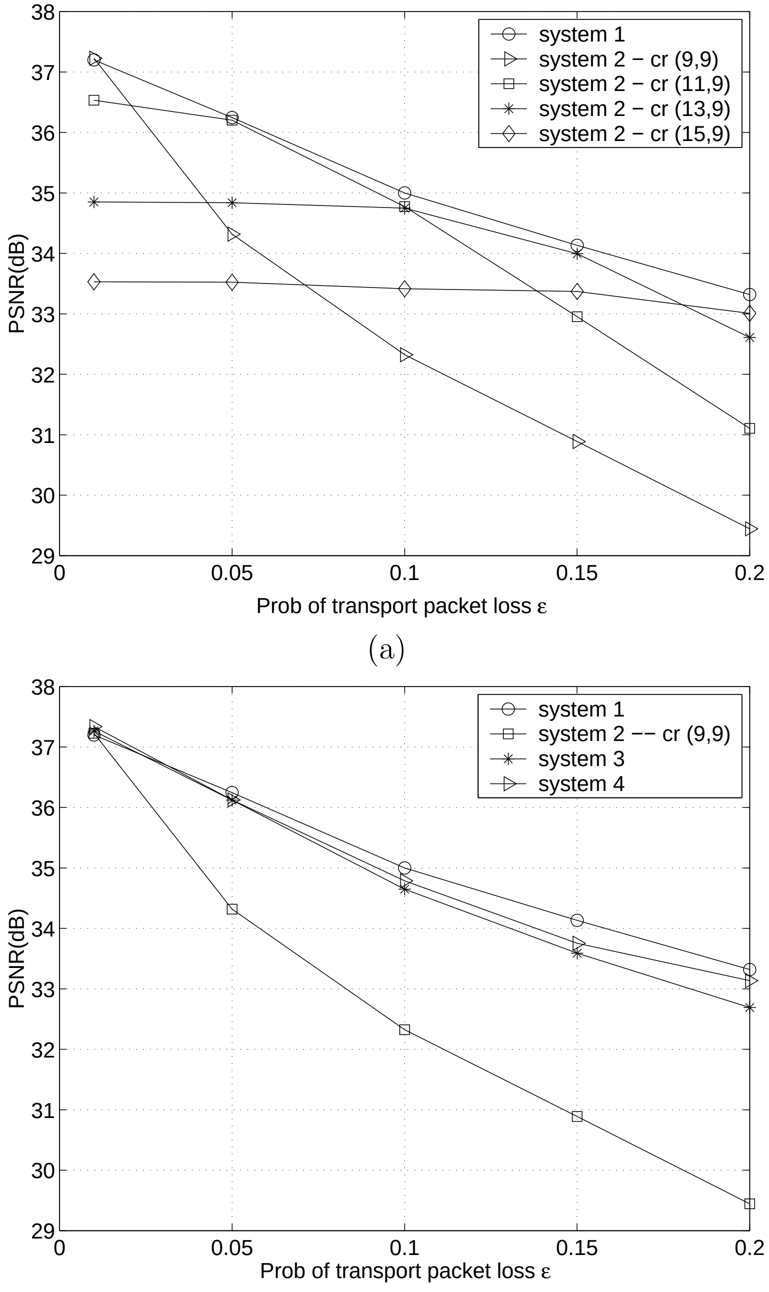
In Fig. 3.3(b), we can see that system 3 has higher average PSNR than system 2 without channel coding. Such a result is expected because FEC can change the channel characteristics to a greater extent (e.g., an RS(7, 5) code can change the packet loss probability from 10% to 1.1%) compared to error resilient source coding, which can only adapt to the channel characteristics to a limited degree. Also, as shown in Fig. 3.3(b), system 1 outperforms systems 3 and 4 with up to around 0.4 dB and 0.3 dB, respectively. The gain in system 1 compared to system 4 is due to the joint consideration of source coding and channel coding. The gain in system 4 in comparison to system 3 comes from the adaptation of source coding to the modified channel characteristics after error recovery (system 3 does not employ error resilient source coding).



(a)

(b)

Figure 3.3: Average PSNR vs. transport packet loss probability (a) System 1 vs. System 2 with indicated channel rates (b) System 1 vs. System 2, 3 and 4 ($R_T = 480$ kbps, $F = 30$ fps, cr in the legend denotes channel rates).



(a)

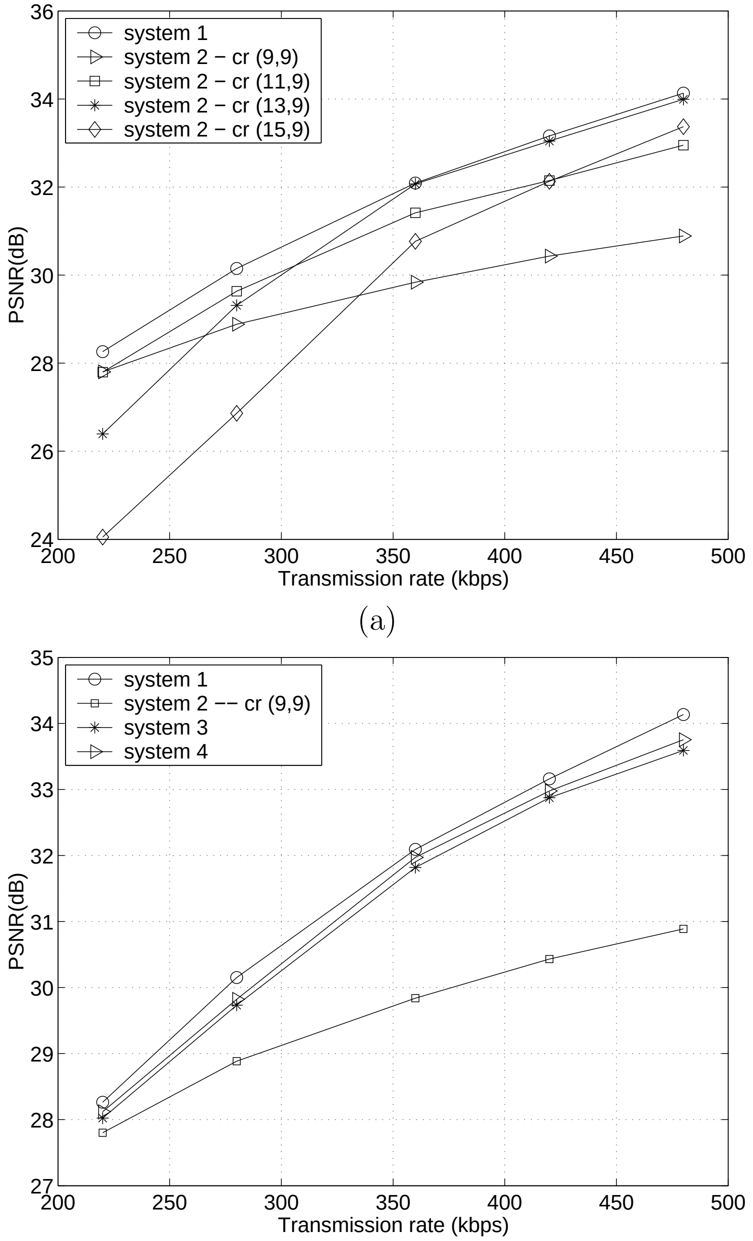
(b)

Figure 3.4: Average PSNR vs. transport packet loss probability (a) System 1 vs. System 2 with indicated channel rates (b) System 1 vs. System 3 and 4 ($R_T = 480$ kbps, $F = 15$ fps, cr in the legend denotes channel rates).

Figure 3.4 shows the same performance comparisons at a lower frame rate of $F = 15$ fps. Since the average bit budget per frame is given by R_T/F, this results in greater bit budget per frame. In this case, when the channel loss rate increases, the PSNR curve for system 2 without channel coding deviates from those with channel coding at a much higher rate compared to the situation in Fig. 3.3. The low bit budget in Fig. 3.3 restricts the use of channel coding, because a majority of the bits are needed for source coding. When the bit budget gets larger, the system becomes more flexible in its ability to allocate bits to the channel to improve the overall system performance.

The effect of bit budget is better illustrated in Fig. 3.5, where the PSNR is plotted against the transmission rate. It can be clearly seen that as the transmission rate increases (i.e., the bit budget per frame increases), the gap between the curves of systems 2 (without channel coding) and the other systems (with channel coding) also increases. Again, as shown Fig. 3.5(a), system 1 outperforms system 2 with different pre-selected channel coding rates and system 3 and 4 at various transmission rates.

The gain of the IJSCC system (system 1) compared with system 3 (without performing error resilient source coding) or system 4 (the sequential JSCC) may not be very significant. This is because in all systems, we perform the optimization by jointly considering several available error control components such as error concealment. Thus, absence of one of the error control components, as in system 3, or lack of the joint consideration of source and channel coding, as in system 4, may not have a very significant effect due to mitigation of other error control components in the system. Another observation is that in practical situations where computation resources are constrained, application of the integrated system may not be necessary



(a)

(b)

Figure 3.5: Average PSNR vs. transmission rate (a) System 1 vs. System 2 with indicated channel rates (b) System 1 vs. System 3 and 4 ($\epsilon = 0.15$, $F = 15$ fps, cr in the legend denotes channel rates).

if the additional gain does not outweigh the additional computational complexity. Nevertheless, the integrated system can still be useful in practical situations in that it provides an optimization benchmark against which the performances of other sub-optimal systems can be evaluated.

3.6 Conclusions

This chapter presented a general resource-distortion optimization framework for studying joint source-network coding. By assuming that the encoder can access and specify the resource allocation parameters in the underlying layers, the cross-layer resources are optimally allocated according to the optimization framework so as to achieve the best video quality. This is an application-layer based approach that achieves cross-layer design for real-time video communications.

The proposed framework not only provides an optimization benchmark against which the performances of other sub-optimal systems can be evaluated, but also gives rise to a useful tool in assessing the effectiveness of different adaptation components in practical system design. Specifically, we took the IJSCC framework, which is a special case of the resource-distortion optimization framework for Internet video transmission, as an example. Through both analysis and simulations, we showed the advantage of the IJSCC framework over the traditional sequential approaches. Depending on the available network adaptation parameters in different applications, we next study the problems of real-time video transmission over the Internet, wireless networks, and DiffServ networks.

Chapter 4

Joint Source-Channel Coding for Internet Video Transmission

In this chapter, we study cross-layer resource allocation problem for Internet video transmission, where the available error control components we consider are error resilient source coding, channel coding, and error concealment. We focus on channel coding in this work. In particular, we study two problems. The first one is application-layer packetization schemes that provide FEC. The second is hybrid error control that consists of FEC and retransmission. The study is carried out in the resource-distortion optimization framework presented in Chapter 3.

4.1 Introduction

Real-time video applications, such as on-demand video streaming, videophony and videoconferencing, have gained increased popularity. However, it is well known that the best effort design of the current Internet makes it difficult for the network to provide the QoS needed by these applications. A direct approach dealing with the lack of QoS is to use error control, where different error control components can be implemented in different network layers. Error control techniques for Internet video transmission in general include error resilient source coding at the encoder, channel coding (FEC and ARQ) at the application layer, and error concealment at the receiver. In this chapter, we jointly consider a combination of these error control approaches. Our study, however, focuses on the channel coding.

With FEC, the packet loss probability can be modeled as specifying a loss probability per packet as a function of the FEC choice. The details of this model will depend on how transport packets are formed from the available video packets. In the first part of this chapter, we study the performance of two application-layer packetization schemes that provide FEC. Note that the packetization we discuss here is transport packetization, whose functionality is to convert source packets to transport packets. Source packetization, which is about how source packets are formed at the encoder, is discussed in Sect. 2.2.2.

In addition to FEC, retransmission-based error control may also be used in the form of ARQ protocols. Such protocols are only useful if the application can tolerate sufficient delay, such as in the case of on-demanding streaming. When ARQ protocols are used, the decision whether to retransmit a packet or send a new one forms another channel coding parameter, which also affects the probability of loss as well as the transmission delay. In the second part of this chapter, we study the performance of pure FEC, pure retransmission, and hybrid FEC and retransmission in achieving protection.

The rest of this chapter is organized as follows. We discuss application-layer packetization in Sect. 4.2 and hybrid application-layer error control in Sect. 4.3. Section 4.4 summarizes this chapter.

4.2 Application-Layer Packetization

For Internet video, in performing FEC, parity packets are usually generated in addition to source packets to perform cross-packet FEC, such as in [55, 95]. We refer to this approach as scheme 1. Alternatively, a source packet can be partitioned into different transport packets to achieve prioritized protection. This scheme has been studied for scalable video such as MPEG-4 FGS and 3D wavelet-based video [68, 94, 116]. Unlike scheme 1, this scheme is seldom used for non-scalable video. In this section, we consider this approach as a means to flexibly provide prioritized FEC for non-scalable video, which is henceforth referred to as scheme 2.

In terms of application-layer packetization, there are other dimensions such as packet size as in [55, 111, 114] that may be taken into account in optimization. In this study, we are mainly interested in the performance comparison of the above mentioned two type of schemes in providing FEC for non-scalable video. In using the resource-distortion optimization framework, we will show that either packetization scheme may be optimal depending on the packet loss rate.

4.2.1 Packetization Schemes

Next we discuss the packetization schemes.

Scheme 1: Figure 4.1(a) illustrates packetization scheme 1 for a frame, where one row corresponds to one GOB. In this packetization scheme, one GOB is directly packetized into one transport packet by the attachment of a transport packet header. Since the source packet sizes $B_{s,k}$ (shown by the shaded area in Fig. 4.1(a)) are usually different, the maximum packet size of a block (a group of packets protected by one RS code) is determined first, then all the packets are padded with stuffing bits in the tail parts to make the size equal. The stuffing bits are removed after the parity codes are generated and thus are not transmitted [55, 96]. The resulting parity packets are all of the same size (maximum packet size mentioned above). Each source packet in Fig. 4.1(a) is protected by an RS(N, M) code.

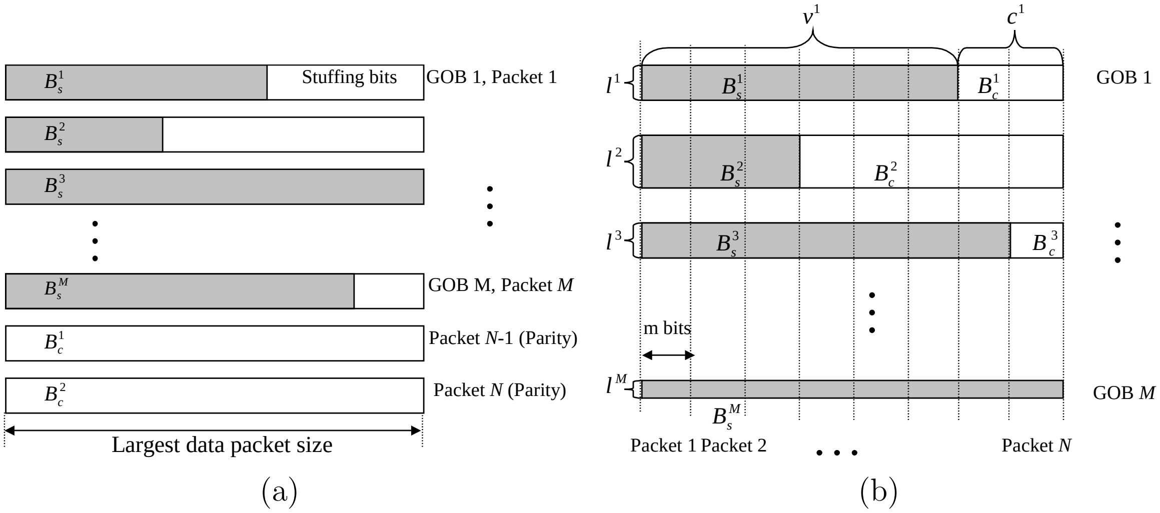


Figure 4.1: Packetization schemes: (a) scheme 1: one row corresponds to a GOB and one transport packet. (b) scheme 2: one row corresponds to a GOB, and one column corresponds to a transport packet.

In this scheme, a source packet is regarded as lost after error recovery at the receiver only when the corresponding transport packet is lost and the block containing the lost transport packet cannot be recovered. Therefore, the probability of source packet loss ρ after error recovery is defined as

$$\rho = \epsilon(1 - \sum_{i=0}^{N-1-M} \binom{N-1}{i} \epsilon^i (1-\epsilon)^{N-1-i}) = \sum_{i=N-M+1}^{N} \frac{i}{N} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}, \qquad (4.1)$$

where ϵ is the probability of transport packet loss. Note that in scheme 1, all source packets in a given frame have the same probability of loss, ρ. Next, we consider a packetization scheme that can easily provide UEP for any packet.

Scheme 2: In this scheme, as shown in Fig. 4.1(b), one row corresponds to one GOB (source packet), and one column corresponds to one transport packet. The source bits and parity bits for the k-th source packet are denoted by $B_{s,k}$ and $B_{c,k}$, respectively. The source bits, $B_{s,k}$, are distributed into v_k transport packets and the redundancy bits, $B_{c,k}$, are distributed into the remaining c_k transport packets, as shown in Fig. 4.1(b). Assuming the symbol length is m bits, $l^k = \frac{B_{s,k}}{mv_k}$ is defined as the length that the k-th source packet occupies in each transport packet, as shown in Fig. 4.1(b).

When the k-th source packet is protected by an RS(N, v_k) code, the probability of losing this source packet is given by

$$\rho_k = \sum_{i=N-v_k+1}^{N} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}. \qquad (4.2)$$

Notice that (4.2) differs from (4.1) in that in this case a particular source packet can not be recovered if fewer than v_k transport packets from a block are received. However, with packetization scheme 1, when fewer than M transport packets arrive, a particular source packet may still be received if its corresponding transport packet is not lost. The difference between these approaches is illustrated in Fig. 4.2, which shows the residual packet loss probability versus the probability of transport packet

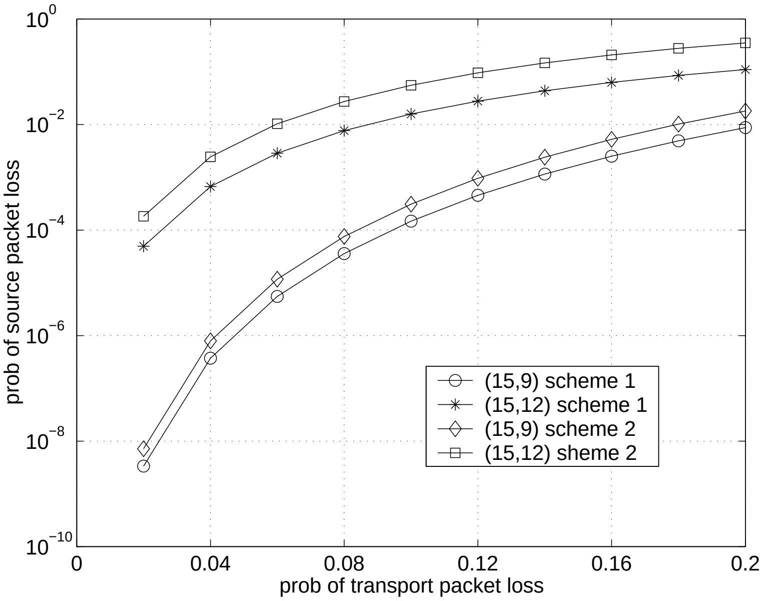


Figure 4.2: Residual packet loss probability of scheme 1 and 2 at different transport packet loss rate and channel rate.

loss with different channel codes for packetization scheme 1 and 2. It is not surprising to see that at any given transport packet loss probability, scheme 1 always produces a lower residual packet loss probability than scheme 2. In addition, as shown in Fig. 4.2, when the channel code rate gets lower, the difference in residual error between scheme 1 and 2 becomes smaller.

However, this does not necessarily mean that packetization scheme 1 outperforms scheme 2. In applying scheme 2, each source packet is protected differently, resulting in UEP for the source packets in a frame. This is a major difference from scheme 1. Due to its flexibility in supporting prioritized protection for different source packets, scheme 2 is widely used in layered or progressive video transmission [68,116]. In Sect. 4.2.3 we demonstrate the advantage of using scheme 2 for single layer video communications when the probability of packet loss is large.

We next study the performance of these two packetization schemes within the IJSCC framework (3.9) presented in Sect. 3.5.2.

4.2.2 Solution Algorithm

As discussed in Sect. 3.5.3, the convex hull solution of (3.9) can be found by solving (3.10) with the appropriate Lagrange multiplier that satisfies the required transmission delay constraint. Given a Lagrange multiplier, the relaxed minimization problem (3.10) itself can be solved using DP as in (3.11). Due to the different structures of packetization, the DP takes different forms.

For simplicity, we omit the frame index in the following equations. In packetization scheme 1, the FEC parameter for one frame is a scalar. Thus, the DP can be solved in two steps: optimal channel coding rate selection resulting in bit allocation between the source and channel coding given the total bit rate constraint, and optimal mode selection given the source bit rate constraint, shown as below.

$$\min_{\{\boldsymbol{\mu}\in\mathcal{Q},\nu\in\mathcal{R}\}} \sum_{k=1}^{M} J_k = \min_{\{\nu\in\mathcal{R}\}} \sum_{k=1}^{M} J_k(\boldsymbol{\mu}^*|\nu) = \min_{\{\nu\in\mathcal{R}\}} \left\{ \min_{\{\boldsymbol{\mu}\in\mathcal{Q}\}} \sum_{k=1}^{M} J_k(\mu_{k-1},\mu_k) \right\}. \tag{4.3}$$

While in using packetization scheme 2, the FEC parameter for a frame is a vector, thus, we have

$$\min_{\{\boldsymbol{\mu}\in\mathcal{Q},\boldsymbol{\nu}\in\mathcal{R}\}} \sum_{k=1}^{M} J_k = \min_{\{\boldsymbol{\mu}\in\mathcal{Q},\boldsymbol{\nu}\in\mathcal{R}\}} \sum_{k=1}^{M} J_k(\mu_{k-1},\mu_k,\nu_{k-1},\nu_k), \tag{4.4}$$

where μ_0 and ν_0 are defined as any constants since they do not play any role in the simulations. Note that the cost function J_k for packet k in (4.3) and (4.3) are dependent on the parameter(s) selected for packet k and $k-1$. This is due to the use of error concealment described in Sect. 2.3.6, based on which the expected distortion for one packet depends on the probability of loss for this packet and the previous one, as shown in (2.15). The resultant time complexity for (4.3) and (4.4) are $O(M \cdot |\mathcal{R}| \times |\mathcal{Q}|^2)$ and $O(M \cdot |\mathcal{R} \times \mathcal{Q}|^2)$, respectively [136].

4.2.3 Experimental Results

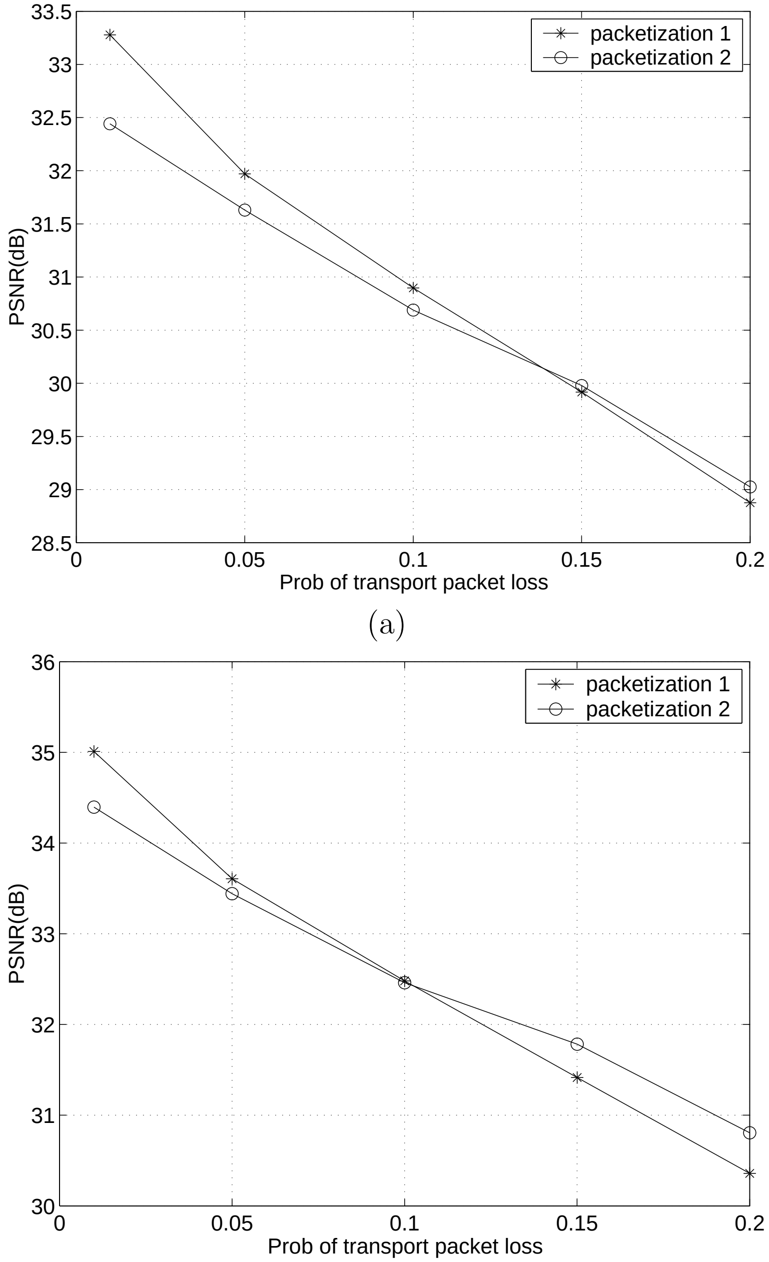
We consider the QCIF format Foreman sequence with at frame rate of 30 fps. Rate controller is not implemented in the video streaming system. Thus, every frame has the same transmission delay constraint of one frame's duration, i.e., $T_0^{(n)} = T_F$. We assume that channel feedback is available to the encoder in the form of which packets are received or lost. An RTP/UDP/IP header of 40 bytes is included for each transport packet. In this work, we assume that the receiver responds to a lost or corrupt packet with a negative acknowledgement (NAK), and responds to a correctly received packet with a positive acknowledgement (ACK). All acknowledgements are assumed to arrive correctly after one round-trip-time (RTT), i.e., the feedback delay is a constant and the feedback channel is error free. Note that this assumption also holds for Sect. 4.3. For simplicity, we set RTT$= 2T_F$ in this work, which is long enough to preclude retransmissions.

Under both schemes, error resilient source coding and FEC are jointly considered within the resource-distortion optimization framework, i.e., they both perform optimal UEP to minimize the expected distortion at given transmission delay constraints. The difference is that in scheme 2, UEP can be achieved from GOB to GOB, whereas in scheme 1, the protection ratio is the same for all GOBs within one frame, although it can vary from frame to frame. We illustrate the performance of the two systems in Fig. 4.3, where we plot the average decoded PSNR of the Foreman sequence against transport packet loss rates at channel transmission rates 360 kbps and 480 kbps, respectively.

It can be seen that at low loss rates, packetization scheme 1 performs better. This is because scheme 1 results in a lower residual packet loss probability than scheme 2 at various channel code rates. However, when the packet loss probability increases, scheme 2 starts to outperform scheme 1. The gain comes from two sources: 1) scheme 2 is more flexible in performing UEP than scheme 1; 2) in scheme 1, the size of the parity packets has to be the maximum size of the source packets. A substantial number of bits are wasted in the application of scheme 1, because some part of the parity packets are used to protect the stuffing bits. Another observation is that the cross point of packet loss probability at higher transmission rates is smaller. This is because at higher transmission rate, more bits are available for each packet, thus the effect of overhead limiting the flexibility of packetization scheme 2 becomes less significant. Table 4.1 shows the protection ratio averaged over all frames from the optimization using the two packetization schemes at R_T=360 kbps.

Prob of transport packet loss	0	0.01	0.05	0.1	0.15	0.2
Protection ratio using scheme 1	0	0.02	0.12	0.21	0.27	0.30
Protection ratio using scheme 2	0	0.10	0.20	0.26	0.31	0.35

Table 4.1: Protection ratios in using packetization scheme 1 and 2 (R_T=360 kbps).



(a)

(b)

Figure 4.3: Performance comparison of packetization scheme 1 and 2: (a) R_T=360 kbps (b) R_T=480 kbps.

4.3 Hybrid FEC and Selective Retransmission

In this section, we focus on channel coding. As discussed in Chapter 2, each of the two error correction mechanisms, FEC and ARQ, has its own benefits in error robustness and network traffic load [17, 91]. In this work, we consider hybrid FEC and ARQ. More specifically, we consider the hybrid of FEC and application-layer selective retransmission in performing optimal error control. Our goal is to study the efficiency and effectiveness of these two error correction techniques in different network situations (such as packet loss probability and network round trip time) and application requirements (such as end-to-end delay).

4.3.1 Related Work

With regard to related work on hybrid FEC/retransmission, in [62], a general cost-distortion framework is proposed to study several scenarios such as DiffServ, sender-driven retransmission and receiver-driven retransmission. Here, we take into account source coding and error concealment, which are not considered in [62]. For wireless IP networks, a link-layer hybrid FEC/ARQ scheme is considered in [137] and an application-layer hybrid FEC/ARQ technique based on heuristics is presented for video transmission in [91], which is based on heuristic methods. A receiver-driven hybrid FEC/Pseudo-ARQ mechanism is proposed for Internet multimedia multicast in [17]. Another related work is [99], which considers scalable video; pure ARQ is used for the base layer and pure FEC is used to protect the enhancement layer. Our work differs from the above in that we consider application-layer sender-driven retransmission, where lost packets are selectively retransmitted according to a rate-distortion optimized policy.

4.3.2 Problem Formulation

Assume that there are up to A frames in the sender's buffer that are eligible for retransmission. Let $\sigma_k^{(n)} \in \{0, 1\}$ denote the retransmission parameter for the k-th source packet in frame n, where 0 denotes no retransmission and 1 denotes retransmission. Let $\boldsymbol{\sigma}^{\boldsymbol{(n)}} = \{\sigma_1^{(n)}, ..., \sigma_M^{(n)}\}$ denote the retransmission parameter vector for frame n, and $\boldsymbol{\sigma} = \{\boldsymbol{\sigma}^{(n-A)}, ..., \boldsymbol{\sigma}^{(n-1)}\}$ the vector for the A frames. For video transmission applications, usually a higher-level rate controller is used to constrain the bits, or equivalently the transmission delay for each frame. For simplicity, let $T_0^{(n)}$ be the transmission delay for the n-th frame obtained from the rate controller. Following the structure of the IJSCC framework in (3.9), we consider the following problem formulation

$$\begin{aligned} &\min_{\{\boldsymbol{\mu},\boldsymbol{\nu},\boldsymbol{\sigma}\}} \sum_{i=0}^{A} E[D^{(n-i)}] = \sum_{i=1}^{A} E[D^{(n-i)}(\boldsymbol{\sigma}^{\boldsymbol{(n-i)}})] + \sum_{k=1}^{M} E[D_k^{(n)}(\boldsymbol{\mu},\boldsymbol{\nu})] \\ &\text{s.t.} \;\; \sum_{i=1}^{A}\sum_{k=1}^{M} \sigma_k^{(n-i)} T_k^{(n-i)} + \sum_{k=1}^{M} T_k^{(n)} \le T_0^{(n)}. \end{aligned} \tag{4.5}$$

Gains might be obtained by grouping the retransmitted packets and the packets in the current frame together to perform FEC. However, this introduces additional delay for the retransmitted packets and considerably complicates the solution of the problem. In this work, we only consider FEC for the current frame.

The above formulation is for an optimization scheme with a sliding window of size $A+1$ frames. The optimization window shifts at the frame level instead of at the packet level, since the latter usually leads to much higher computation complexity.

In addition, the packets in one frame typically have the same deadline for playback. In this formulation, when processing each frame, we assume that all the raw data for the frame is available in a buffer, and the optimization (retransmission policy for the first A frames based on feedback, and source coding and FEC for the current frame) is performed on the $A+1$ frames in the window. After optimization is done, the retransmitted packets and the transport packets in the current frame (including source packets and parity packets) are transmitted over the network. After the transmission of these packets, the window shifts forward by one frame, and the optimization is solved again based on the updated feedback.

When each frame is encoded, the probability of packet loss for all the past A frames is updated based on the received feedback. For example, if one packet is known to be received, its probability of loss becomes 0; if one is lost, its loss probability becomes 1 if no further retransmission for this packet has been initiated. Based on the updated probabilities of packet loss, the expected distortion of all packets in the encoder buffer is recursively re-calculated as in (2.7). In using this model, the error propagation due to packet loss (after 1 RTT) can be fully captured and consequently the effect of previously lost packets on the future frames is taken into account. Since each time we do not consider re-encoding the past A frames, the complexity in updating the expected distortion is not significant.

Additional gain may be obtained by considering the future frames when the current frame is encoded. For example, by doing so, the effect of the parameter decisions in the optimization window on future frames can be taken into account. This will generally result in better performance due to the motion-compensation dependencies of video frames. However, this leads to a very complicated and usually intractable problem. In addition, for a real-time application, future frames may not be available when the current frame is encoded.

4.3.3 Calculation of Packet Loss Probability

We discuss next how to calculate the probability of packet loss ρ_k in order to find the expected distortion in (2.7). For a packet in the current frame, the probability of packet loss can be defined as $\rho_k^{(n)} = \rho_{k,FEC}^{(n)}\rho_{k,RET}^{(n)}$, where $\rho_{k,FEC}^{(n)}$ and $\rho_{k,RET}^{(n)}$ denote respectively the probability of packet loss due to FEC and retransmission. $\rho_{k,FEC}^{(n)}$ is defined in (4.1). The probability of loss in future retransmissions can only be estimated since the acknowledgement information and retransmission decisions (note that lost packets are selectively retransmitted) are not available in the encoding of the current frame. In this work, we give an approximate formula to estimate it, i.e., $\rho_{k,RET}^{(n)} = \epsilon^{\tilde{m}}$, where $\tilde{m}$ denotes the estimate of the total number of retransmissions for the k-th packet, m. Note that m is not a constant and is dependent on how $\rho_{k,RET}^{(n)}$ itself is calculated and the future video content. In addition, the effect of packet recovery due to other packets' retransmissions should also be taken into account when calculating $\rho_{k,RET}^{(n)}$. However, it is almost impossible to accurately estimate this factor due to the use of block code. In our estimation formula, although this factor is not explicitly indicated, it has been taken into account by the estimate of $\tilde{m}$. In this work, we use an estimate of $\tilde{m}$ developed from simulations. Figure 4.4 shows the performance of the hybrid FEC/retransmission system versus m for the QCIF format Foreman sequence and Akiyo sequence (here m is fixed for the entire sequence). In both tests, the number of frames that are eligible for retransmissions is $A = 4$ and the frame rate is 15 fps. The transmission rate is 480 kbps and 360 kbps, respectively.

Based on these results, we use $\tilde{m}$, calculated by $\tilde{m} = \frac{A}{(1+\text{RTT})^2}$, where RTT is the round-trip-time in the units of one frame's duration T_F; this appears to provide good performance and is used subsequently. Note that the maximum number of available retransmission opportunities is $\lfloor A/(1+\text{RTT})\rfloor$. In addition, from Fig. 4.4, we can see that the system performance is not very sensitive to the choice of $\tilde{m}$.

In considering the possible retransmission of packets in the current frame, the expected additional transmission delay used for retransmission in the future should be taken into account; it is calculated by $E[\Delta T_k^{(n)}] = \sum_{k=1}^{M} \tilde{m}\rho_{k,FEC}^{(n)} T_k^{(n)}$. The delay constraint in (4.5) is modified accordingly.

For a lost packet in the past frames, we let $\rho_k^{(n-i)} = \rho_{k,UPD}^{(n-i)}\rho_{k,RET}^{(n-i)}$ for $i = 1, ..., A$, where $\rho_{k,UPD}^{(n-i)}$ is the updated probability of packet loss based on feedback and $\rho_{k,RET}^{(n-i)}$ is the probability of packet loss due to retransmissions. Assume that one past frame is protected by an RS(N, M), and L packets are lost. Let $J = L + M - N$ and V be the number of retransmitted packets in that frame. Taking into account the RS codes, the calculation of $\rho_{k,RET}^{(n-i)}$ is different for the lost packets that are either retransmitted or those that are not. If $V < J$, we have

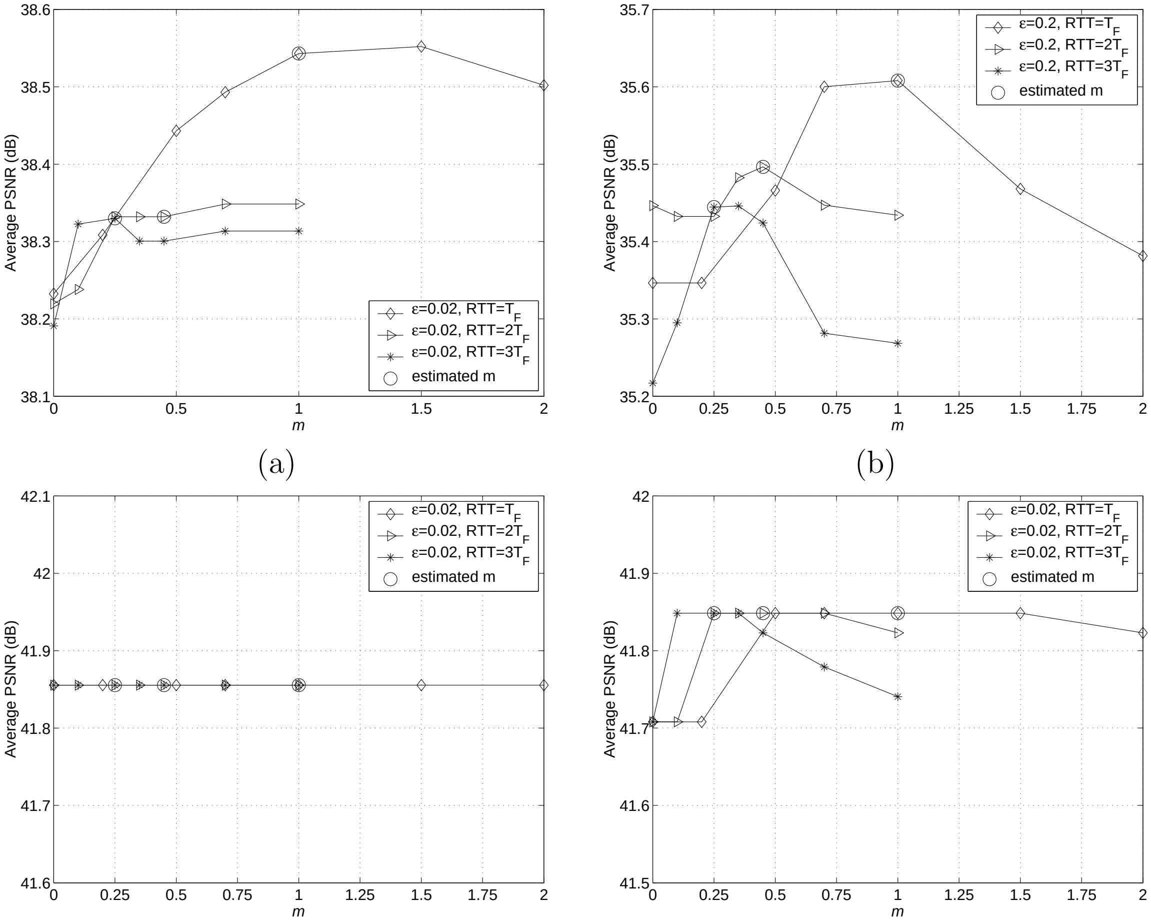
$$\rho_{k,RET}^{(n-i)} = \epsilon^{\sigma_k^{(n-i)}};$$

if $V = J$, we have

$$\rho_{k,RET}^{(n-i)} = \begin{cases} \epsilon & \text{if } \sigma_k^{(n-i)} = 1 \\ 1-(1-\epsilon)^J & \text{if } \sigma_k^{(n-i)} = 0; \end{cases}$$

and if $V > J$ we have

$$\rho_{k,RET}^{(n-i)} = \begin{cases} \sum_{j=V-J+1}^{V} \frac{j}{V}\binom{V}{j}\epsilon^j(1-\epsilon)^{V-j} & \text{if } \sigma_k^{(n-i)} = 1 \\ \sum_{j=V-J+1}^{V} \binom{V}{j}\epsilon^j(1-\epsilon)^{V-j} & \text{if } \sigma_k^{(n-i)} = 0. \end{cases}$$



(a) (b)

(c) (d)

Figure 4.4: Average PSNR vs. m in the hybrid FEC/retransmission system; (a) and (b) QCIF Foreman sequence at $F = 15$ fps, $R_T = 480$ kbps and $A = 4$, (c) and (d) QCIF Akiyo sequence at $F = 15$ fps, $R_T = 360$ kbps and $A = 4$.

Note that when the above formulas are derived, the effect of future possible retransmission is ignored. The effect of future retransmissions could be included by replacing ϵ with $\epsilon^{\tilde{m}}$, as in the estimation of $\rho_{k,RET}^{(n)}$ for the current frame. However, we noticed that due to the limited window size, the maximum number of possible retransmissions is relatively small (3 in our simulations). In that case, the possibility that one packet is retransmitted twice is very small. Furthermore, by ignoring this factor in the calculation of $\rho_{k,RET}^{(n-i)}$, the system tends to retransmit the lost packets rather than waiting for a second chance of retransmission, which usually leads to better system performance. We next present our solution to this formulation.

4.3.4 Solution Algorithm

By using a Lagrange multiplier $\lambda \geq 0$, (4.5) can be converted into an unconstrained problem as,

$$\begin{aligned} & \min_{\{\boldsymbol{\mu}\in\mathcal{Q},\boldsymbol{\nu}\in\mathcal{R},\boldsymbol{\sigma}\in\mathcal{P}\}} \sum_{i=0}^{A} J^{(n-i)} \\ & = \sum_{i=1}^{A} E[D_k^{(n-i)}(\boldsymbol{\sigma}^{(n-i)})] + \sum_{k=1}^{M} E[D_k^{(n)}(\boldsymbol{\mu},\boldsymbol{\nu})] \\ & + \lambda \left\{ \sum_{i=1}^{A}\sum_{k=1}^{M} \sigma_k^{(n-i)} T_k^{(n-i)} + \sum_{k=1}^{M} T_k^{(n)} \right\} \end{aligned} \tag{4.6}$$

Given an appropriate λ, we can write the problem as:

$$\begin{aligned} & \min_{\{\boldsymbol{\mu}\in\mathcal{Q},\boldsymbol{\nu}\in\mathcal{R},\boldsymbol{\sigma}\in\mathcal{P}\}} \sum_{i=0}^{A} J^{(n-i)} \\ & = \min_{\{\boldsymbol{\sigma}\in\mathcal{P}\}} \sum_{i=1}^{A} J^{(n-i)}(\boldsymbol{\sigma}^{(n-i)}) + \min_{\{\boldsymbol{\nu}\in\mathcal{R}\}} \left\{ \min_{\{\boldsymbol{\mu}\in\mathcal{Q}\}} \sum_{k=1}^{M} J_k^{(n)}(\boldsymbol{\mu},\boldsymbol{\nu}) \right\}, \end{aligned} \tag{4.7}$$

where $J^{(n-i)} = E[D_k^{(n-i)}] + \lambda \sum_{k=1}^{M} \sigma_k^{(n-i)} T_k^{(n-i)}$ and $J_k^{(n)} = E[D_k^{(n)}(\boldsymbol{\mu}, \boldsymbol{\nu})] + \lambda T_k^{(n)}$. There are three minimizations in (4.7). They correspond to the bit allocation for retransmission, bit allocation for FEC, and the optimal mode selection for the current frame based on the remaining delay. The first and second steps can be solved by using exhaustive search, and the optimal mode selection can be found using a DP approach. Note that by using the error concealment strategy described in Sect. 2.3.6, the time complexity is $O(|2^L \times |\mathcal{R}| \times M \times |\mathcal{Q}|^2)$, where L is the number of lost packets in the optimization window. If the error concealment strategy does not introduce dependency across source packets, the time complexity would be $O(|2^L \times |\mathcal{R}| \times M \times |\mathcal{Q}|)$ [136].

4.3.5 Experimental Results

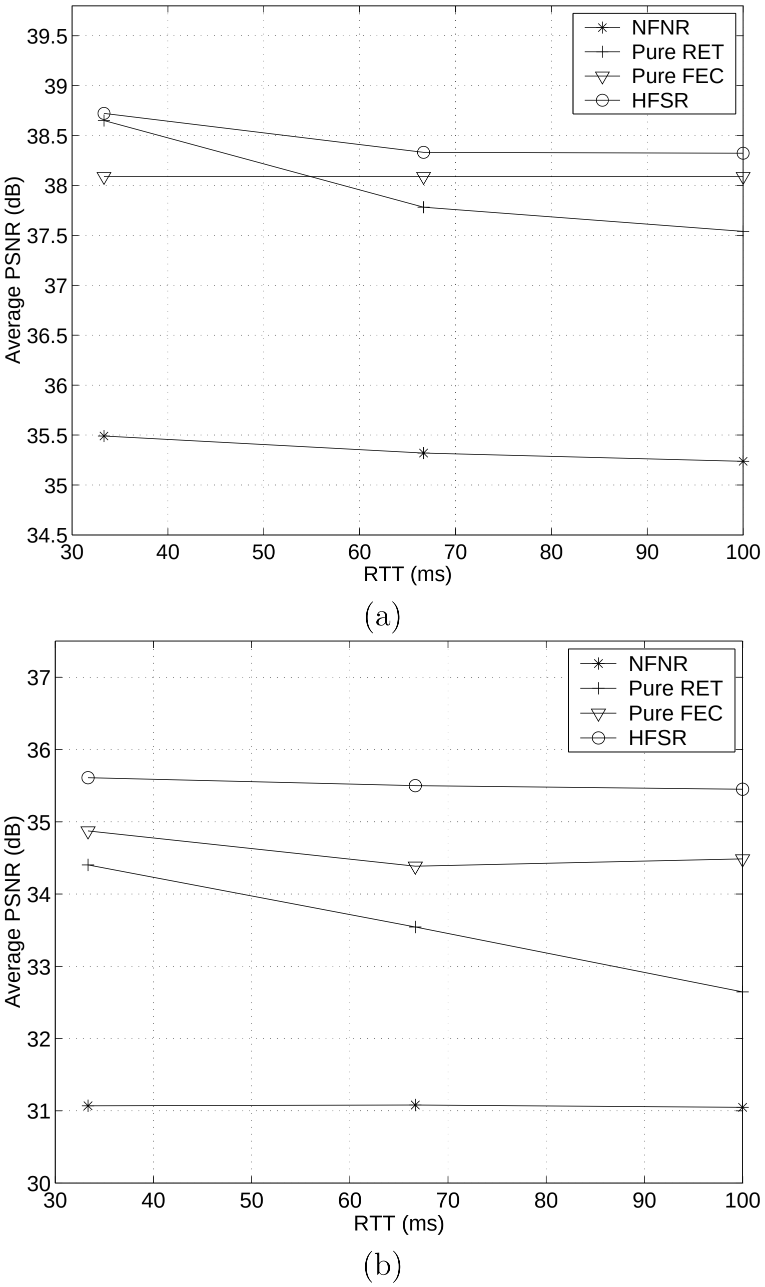
Four schemes are compared: i) neither FEC nor retransmission (NFNR), ii) pure retransmission, iii) pure FEC, and iv) Hybrid FEC and selective Retransmission (HFSR). All four systems are optimized using the IJSCC framework. Although the IJSCC framework in (4.5) is general, in our simulations, we restrict a packet's retransmission only when its NAK has been received. In all experiments of this subsection, we consider the QCIF Foreman sequence and set $A = 4$.

Sensitivity to RTT

Figure 4.5 shows the performance of the four systems in terms of PSNR versus RTT, with different levels of channel loss rate. We set $R_T = 480$ kbps and $F = 15$ fps. As expected, the HFSR system offers the best overall performance. It can also be seen that the pure retransmission approach is much more sensitive to variations in the RTT than FEC. In addition, at low ϵ and low RTT, the pure retransmission approach outperforms the pure FEC system, as shown in Fig. 4.5(a). However, when the channel gets worse and the RTT becomes larger, the pure FEC system starts to outperform the pure retransmission system, as shown in Fig. 4.5(b). This means that retransmission is suitable for those applications where the RTT is short and channel loss rate is low, which confirms the observation in [91]. The disadvantage of retransmission when the RTT gets longer stems from two factors: 1) Given the same value of A, which is decided by the initial setup time T_{max}, the number of retransmission opportunities becomes smaller; 2) Errors accumulated due to error propagation from the motion compensation become larger, and consequently retransmission of lost packets becomes less efficient.

Sensitivity to packet loss rate

In Fig. 4.6, we plot the performance of the four systems in terms of PSNR versus probability of transport packet loss for different values of RTT when $R_T = 480$ kbps and $F = 15$ fps. The RTT is set equal to T_F and $3T_F$ in Fig. 4.6(a) and (b), respectively. It can be seen that the HFSR system achieves the best overall performance of the four. The resulted PSNR in the pure retransmission system drops faster than the pure FEC system, which implies retransmission is more sensitive to packet loss rate. In addition, the pure retransmission system only outperforms the pure FEC system at low ϵ. When the channel loss rate is high, FEC is more efficient since retransmission techniques require frequent retransmissions to recover from packet loss, which results in high bandwidth consumption and is also limited by the delay constraint. For example, when RTT$= 3T_F$ and $A = 4$, each lost packet



(a)

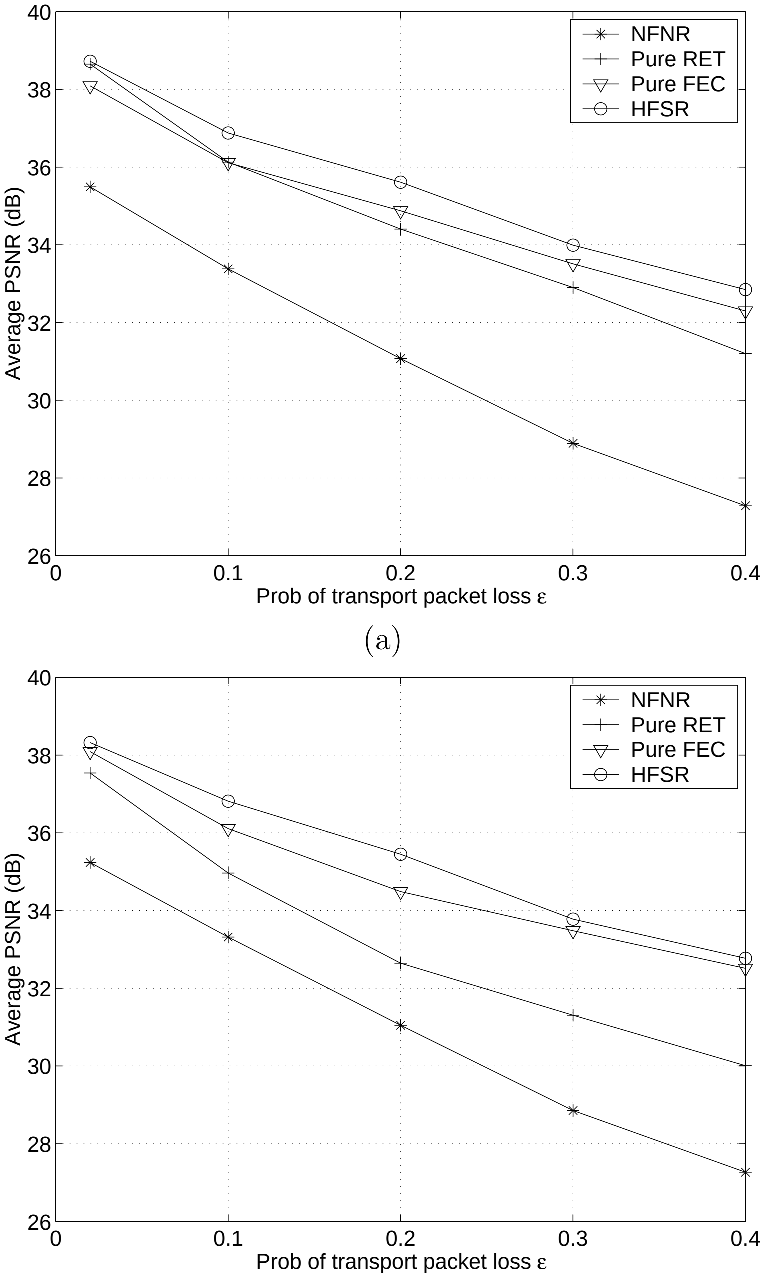
(b)

Figure 4.5: Average PSNR vs. RTT, $R_T = 480$ kbps, $F = 15$ fps (a) ϵ=0.02 (b) $\epsilon = 0.2$.

has only one chance for retransmission, which is not enough to recover many losses when $\epsilon = 0.3$. However, when the channel loss rate is small and the RTT is small, retransmission becomes more efficient, since FEC typically requires a fixed amount of bandwidth overhead. Consequently, the pure retransmission system performs close to the HFSR system, as shown in Fig. 4.6(a).

Sensitivity to transmission rate

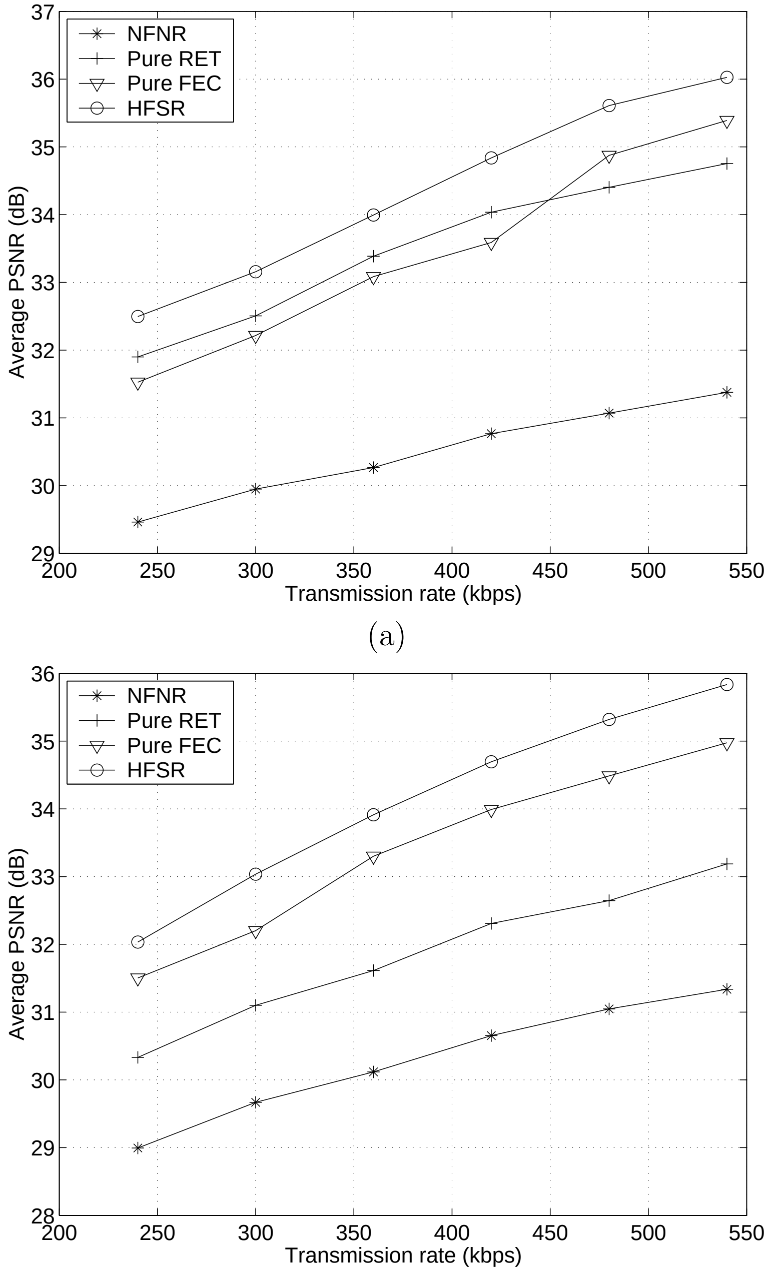
Figure 4.7 shows the performance of the four systems in terms of PSNR versus transmission rate when $\epsilon = 0.2$ and $F = 15$ fps. The RTT is set equal to T_F and $3T_F$ in Fig. 4.7(a) and (b), respectively. As shown in Fig. 4.7(a), when RTT$= T_F$, the pure retransmission system outperforms the pure FEC system by up to 0.4 dB when the transmission rate is less than 450 kbps. When the transmission rate is greater than 450 kbps, the pure FEC system starts to outperform the pure retransmission system by up to 0.5 dB. When the RTT becomes longer, as shown in Fig. 4.7(b), although the pure FEC system always outperforms the pure retransmission system, the difference between the two systems increases from 1.2 dB to 1.8 dB when the transmission rate increases from 240 kbps to 540 kbps, which means that FEC is more sensitive to variations in the transmission rate. These observations imply that FEC is more efficient than retransmission when the transmission rate becomes greater (resulting in a higher bit budget per frame). As discussed in Sect. 3.5.4, this is due to the constant overhead introduced by FEC, which restricts the use of FEC. When the bit budget gets larger, however, the system becomes more flexible in its ability to allocate bits to the channel to improve the overall performance. In addition, it can be seen that the HFSR system achieves the best overall performance of the four.



(a)

(b)

Figure 4.6: Average PSNR vs. probability of transport packet loss ϵ, $R_T = 480$ kbps, $F = 15$ fps (a) RTT=T_F (b) RTT=$3T_F$.



(a)

(b)

Figure 4.7: Average PSNR vs. channel transmission rate R_T, $\epsilon = 0.2$, $F = 15$ fps (a) RTT=T_F (b) RTT=$3T_F$.

Although we only showed simulation results based on the QCIF Foreman sequence, extensive experiments have been carried out and similar results were obtained using other test sequences such as Akiyo, Container, and Carphone.

In summary, given our simulation settings, retransmission is suitable for short network RTT, low probability of packet loss, and low transmission rate, while FEC is more suitable otherwise. In general, our proposed hybrid FEC and selective retransmission scheme is able to find the best combination of the two.

4.4 Conclusions

This chapter addressed the problem of optimal application-layer error control for real-time video transmission over the Internet. We jointly considered error resilient source coding at the encoder, FEC and retransmission at the application layer, and error concealment at the receiver, to achieve the best video quality. In particular, we have compared the efficiencies of two packetization schemes in providing FEC for packetized video applications. Simulation results showed that packetization scheme 2, which is widely used for scalable video coding, is also suitable for non-scalable video, especially at high packet loss rates, because of its efficiency and flexibility in providing UEP. In addition, we have studied the performance of different application-layer error control scenarios such as pure FEC, pure ARQ, and hybrid FEC/selective retransmission. Through simulations, we have illustrated how sensitive each error correction technique is to the variations of network RTT, packet loss rate, and transmission rate. Simulation results also showed that the proposed hybrid FEC/retransmission system achieved better performance than pure FEC and pure retransmission systems.

Chapter 5

Joint Source-Channel Coding and Power Adaptation for Energy Efficient Wireless Video Communications

In this chapter, we study efficiently transmitting video over a hybrid wireless/wire-line network by optimally allocating resources across multiple protocol layers. The resource-distortion optimization framework here is in the form of joint source-channel coding and power adaptation (JSCCPA), where error resilient source coding, channel coding, and transmission power allocation are jointly designed to compensate for channel errors. The focus of this work is on the channel coding and transmission power adaptation. In particular, we consider the combination of two types of channel coding – inter-packet coding (at the transport layer) to provide protection against packet dropping in the wire-line network and intra-packet coding (at the link layer) to provide protection against bit errors in the wireless link. In both cases, we allow the coding rate to be adaptive to provide unequal error protection at both the packet and frame level. In addition to both types of channel coding, we also compensate for channel errors by adapting the transmission power used to send each packet. An efficient algorithm based on Lagrangian relaxation and the method of alternating variables is proposed to solve the resulting optimization problem. Simulation results are shown to illustrate the advantages of joint optimization across multiple layers.

5.1 Introduction

We consider streaming applications with relatively strict delay constraints; for such applications, FEC is the preferred channel coding technique to recover packet losses. The type of FEC method used depends on the requirements of the application and the nature of the channel. Packet loss in an IP-based hybrid wireless/wire-line network typically has two components: packet loss due to congestion in the wired channel and unrecoverable bit errors due to fading in the wireless channel [67, 79]. One way to combat these two types of packet loss is to use both inter-packet and intra-packet protection. To protect against packet loss in the wired link, inter-packet FEC is performed at the transport layer through the generation of parity packets in addition to source packets. This is usually achieved by using erasure codes. In the link layer, redundant bits are added within a packet to provide intra-packet protection from bit errors in the wireless link [93]. The combination of the above two techniques, i.e., intra and inter-packet FEC, is termed as product code FEC (PFEC). A PFEC scheme is proposed in [138] to combat channel variations in progressive image transmission. In that work, intra-packet FEC is achieved through a concatenated CRC/RCPC code, and inter-packet FEC through a systematic Reed-Solomon (RS) code. In [139], an efficient algorithm is provided for finding an optimal equal error protection (EEP) solution for packetized progressive image transmission. Our work considers the use of PFEC in video coding and transmission. An important contribution here is the consideration of UEP for video packets in using PFEC.

Besides FEC, transmitter power adjustment also affects the characteristics of the wireless channel as seen by the video encoder. Allocating different transmitter power levels to different packets results in different probabilities of loss or delay for these packets. For video transmission over wireless networks, the efficient utilization of transmission energy is a critical design consideration [140]. Thus, transmission power needs to be balanced against delay to achieve the best video delivery quality [100,101]. If the encoder can specify the transmission power (at the physical layer) for each transmission bit or packet, the question is how to minimize the end-to-end distortion by optimally allocating bandwidth (bits) between source coding and channel coding, and optimally allocating energy (power) to each packet.

With regard to related work of cross-layer design for energy efficient wireless multimedia communications, joint error resilient source coding (quantization parameter and mode selection) and power management for energy efficient wireless video transmission has been studied in [72], premised on a perfect channel coding mechanism. In [127], the selection of source coding parameters is jointly considered with transmitter power and rate adaptation, and packet transmission scheduling for energy efficient wireless video streaming. A joint source coding and power control approach is presented in [134] for optimally allocating source coding rate and bit energy normalized with respect to the multiple-access interference noise density in the context of 3G CDMA networks. The work in [134] did not address error resilient source coding and channel coding. Joint source-channel coding and transmission power allocation has been studied in [117] for progressive image transmission. A joint FEC and transmission power allocation scheme for layered video transmission over a multiple user CDMA network is proposed in [135] based on the 3D-SPIHT codec. Source coding and error concealment are not considered in that work. Joint source-channel coding and processing power control for transmitting layered video over a 3G wireless network is studied in [46]. An adaptive cross-layer protection scheme is presented in [111] for robust scalable video transmission over 802.11 wireless LANs, where application-layer FEC, the MAC (media access control) retransmission limit, and packet sizes are jointly considered.

We jointly consider cross-layer error control components, including error resilient source coding, channel coding, transmitter adaptation, and error concealment in the JSCCPA framework [75]. In this framework, we consider how to optimally allocate bits between source coding, transport-layer FEC and link-layer FEC, together with power adaptation in the physical layer, to achieve the best video quality at the receiver end given an energy and transmission delay constraint. To tackle the resulting optimization problem with two constraints, an efficient algorithm based on Lagrangian relaxation is proposed.

The rest of this chapter is organized as follows. We first describe product FEC in Sect. 5.2. In Sect. 5.3 the problem formulation of JSCCPA is presented. Section 5.4 presents the proposed solution algorithm. Experimental results are discussed in Sect. 5.5. Finally, we summarize this chapter in Sect. 5.6.

5.2 Product Code FEC

In the transport layer, we consider a systematic RS code to provide inter-packet protection. A popular family of codes used to perform link-layer FEC with variable code rates are RCPC codes [77], which are used in this work.

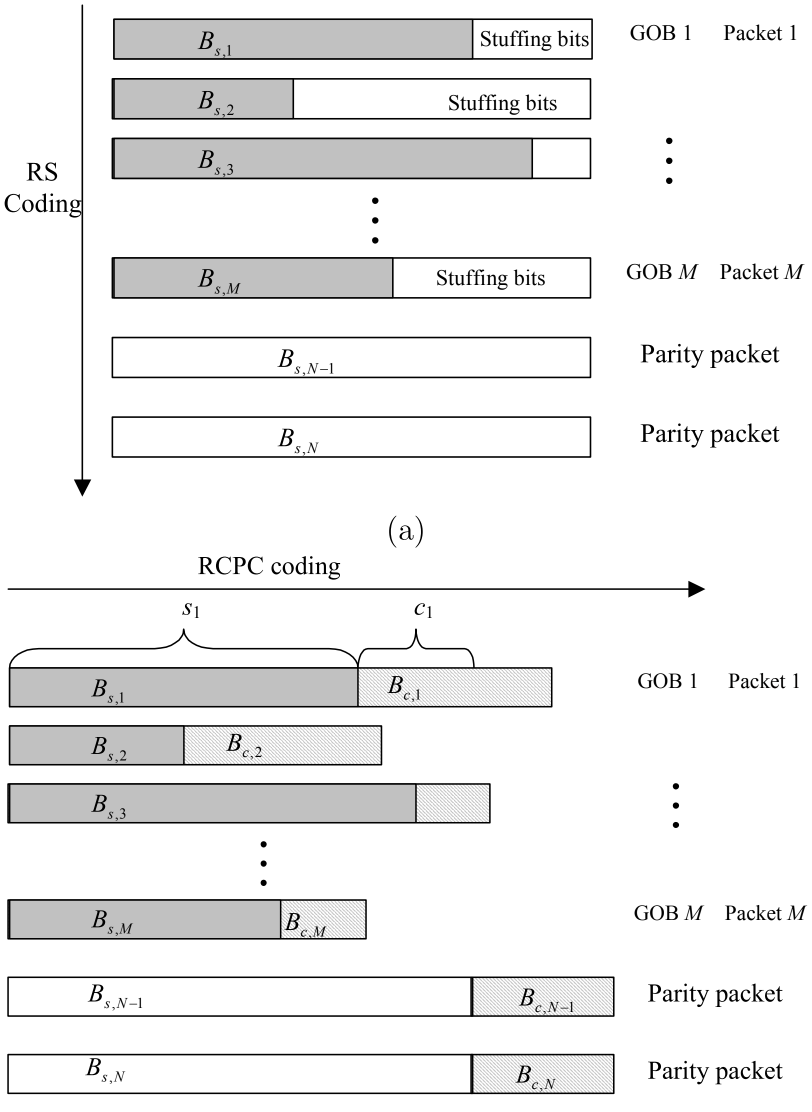
In the product FEC scheme considered here, the first step is to perform RS coding at the transport layer. The same as scheme 1 described in Sect. 4.2.1, after packetization at this stage, each source packet is protected by an RS(N, M) code, as shown in Fig. 5.1(a). In the link layer, each packet (including the parity packets) is padded with parity bits. As shown in Fig. 5.1(b), by using a particular RCPC code with rate r_k, the length of packet k is then $B_k = B_{s,k} + B_{c,k} = B_{s,k}/r_k$.

5.2.1 Calculation of Transport Packet Loss Probability

Next, we discuss how to calculate the probability of loss for each transport packet. Let p_b be the bit error rate (BER) after link-layer channel decoding, i.e., p_b is the BER as seen by the application. Assuming independent bit errors (i.e., the additive noise and fading are each i.i.d. and independent of each other), the loss probability for a transport packet in the wireless channel can be calculated as

$$\beta_k(\mu_k, \nu_k, \eta_k) = 1 - (1 - p_b)^{B_{s,k}}, \tag{5.1}$$

where $B_{s,k}$ is the number of source bits in this transport packet. Note that the probability of packet loss β_k is a function of source coding parameter, the intra-packet channel coding rate, and transmission power level selected for this packet (since p_b is



(a)

(b)

Figure 5.1: (a) Step 1: Transport layer RS coding; (b) Step 2: Link layer RCPC coding.

a function of the channel BER before channel decoding p_e and channel coding rate r_k, where p_e is calculated from (2.11) or (2.12), depending on which channel model is used). Let α denote the packet loss rate in the wired part. At the IP level, as in [79], the network can be modeled as the combination of two independent packet erasure channels: the wired part with loss rate α and the wireless part with loss rate β. Thus, the overall loss rate of a transport packet is then equal to

$$\epsilon_k(\mu_k, \nu_k, \eta_k) = \alpha + (1-\alpha)\beta_k(\mu_k, \nu_k, \eta_k). \tag{5.2}$$

5.2.2 Calculation of Source Packet Loss Probability

We next discuss how to translate the transport packet loss probability ϵ_k into the source packet loss probability ρ_k. Let $\mathcal{Q}$, Γ, $\mathcal{R}$, and $\mathcal{P}$ be the sets of allowable source coding parameters, RS coding parameter, RCPC coding parameters, and transmission power levels, respectively. Let $\mu_k \in \mathcal{Q}$, $\gamma \in \Gamma$, $\nu_k \in \mathcal{R}$, and $\eta_k \in \mathcal{P}$ represent the corresponding parameters selected for the k-th packet. Following the same notation used in [139], let $Q_j^t(N), j = 1, ..., N,\ t = 1, ..., \binom{N}{j}$ denote the t-th subset with j elements of $Q(N) = \{1, ..., N\}$, and $\overline{Q_j^t}$ its complement. For example, if $N = 3$, then $Q(3) = \{1, 2, 3\}$, $Q_1^1(3) = \{1\}$, $Q_1^2(3) = \{2\}$, $Q_1^3(3) = \{3\}$, $Q_2^1(3) = \{1, 2\}$, $Q_2^2(3) = \{1, 3\}$, $Q_2^3(3) = \{2, 3\}$, $Q_3^1(3) = \{1, 2, 3\}$. Let $I_j(N, k) = \{Q_j^t \subseteq Q(N) | k \in Q_j^t(N), |Q_j^t| = j\}$, then the loss probability of a source packet can be written as

$$\begin{aligned}\rho_k(\boldsymbol{\mu}, \gamma, \boldsymbol{\nu}, \boldsymbol{\eta}) &= \sum_{j=N(\gamma)-M+1}^{N(\gamma)} P_{loss,k}(N(\gamma), j) \\ &= \sum_{j=N-M+1}^{N} \sum_{Q_j^t \in I_j(N,k)} \left(\prod_{i \in Q_j^t} \epsilon_i \prod_{l \in \overline{Q_j^t}} (1-\epsilon_l) \right),\end{aligned} \tag{5.3}$$

where $P_{loss,k}(N, j)$ is the probability that the k-th packet is not correctly decoded by the RCPC decoder and the total number of transport packets that are not correctly received from the group of N packets is j. Let $\boldsymbol{\mu} = \{\mu_1, \mu_2, ..., \mu_M\}$ denote the vector of source coding parameters for the M source packets, and $\boldsymbol{\nu} = \{\nu_1, \nu_2, ..., \nu_N\}$ and $\boldsymbol{\eta} = \{\eta_1, \eta_2, ..., \eta_N\}$ the vectors of RCPC coding rates and power levels for the N transport packets in a frame, respectively. Note that the calculation of $\rho_k(\boldsymbol{\mu}, \gamma, \boldsymbol{\nu}, \boldsymbol{\eta})$ itself is rather complicated, as shown in (5.3). In addition, ρ_k not only depends on the source coding parameter, intra-packet FEC parameter, and power level parameter selected for that packet, but also on the parameters chosen for all the other packets in the frame. This complicated inter-packet dependency stems from two factors. The first complication is due to the fact that the loss probability of a transport packet $\epsilon_k(\mu_k, \nu_k, \eta_k) = \alpha + (1 - \alpha)\beta_k(\mu_k, \nu_k, \eta_k)$ differs from packet to packet[1]. The second complication is due to the inter-packet dependency introduced by inter-packet FEC. Together, these make the expected distortion for one packet depend on the parameters selected for all the packets in the same frame.

5.3 Problem Formulation

We first consider real-time video transmission from a mobile device to a receiver through a heterogeneous wireless network, as shown in Fig. 2.6. In this case, the efficient utilization of transmission energy is a critical design consideration [140]. In addition, each video packet should meet a delay constraint in order to reach the

[1] As shown in (5.1), the loss probability for a transport packet in the wireless channel $\beta_k(\mu_k, \nu_k, \eta_k)$ is a function of the source coding parameter, the link-layer FEC parameter, and the transmission power level selected for this packet [since p_b is a function of the channel BER p_e and the link-layer channel rate r_k, where p_e is calculated from (2.11)].

receiver in time for playback. In an energy-efficient wireless video transmission system, transmission power needs to be balanced against delay to achieve the best video quality [100]. Specifically, for a given transmission rate, increasing the transmission power will increase the energy per bit and consequently decrease BER, as shown in (2.11), resulting in a smaller probability of packet loss. On the other hand, for a fixed transmission power, increasing the transmission rate will increase the BER but decrease the transmission delay needed for a given amount of data (or allow more data to be sent within a given time-period). Therefore, in order to efficiently utilize resources such as energy and bandwidth, those two adaptation components should be jointly designed.

By jointly considering error resilient source coding $\boldsymbol{\mu}$, transport-layer FEC γ, link-layer FEC $\boldsymbol{\nu}$, power adaptation $\boldsymbol{\eta}$, and error concealment, the JSCCPA problem is formulated as:

$$\begin{aligned} \min_{\{\boldsymbol{\mu}\in\mathcal{Q},\gamma\in\Gamma,\boldsymbol{\nu}\in\mathcal{R},\boldsymbol{\eta}\in\mathcal{P}\}} E[D] &= \sum_{k=1}^{M} E[D_k(\boldsymbol{\mu},\gamma,\boldsymbol{\nu},\boldsymbol{\eta})] \\ \text{s.t.} \quad C &= \sum_{k=1}^{N(\gamma)} B_k P_k(\eta_k)/R_T \le C_0 \\ T &= \sum_{k=1}^{N(\gamma)} B_k/R_T \le T_0, \end{aligned} \tag{5.4}$$

where B_k and P_k are, respectively, the number of bits (including both source bits and channel bits) and the power level for the k-th packet; M and N are, respectively, the number of source packets and the total number of transport packets in one frame; R_T is the transmission rate; and C_0 and T_0 are the energy and transmission delay constraint for the frame, respectively.

As a special case, we also focus on the last hop of a wireless network and consider transmitting real-time video from a mobile device to the base station over a single wireless link; this is likely to be the bottleneck of the whole video transmission system. In Sect. 5.5.1, we show through simulations that inter-packet FEC is not helpful in this case (at least for the cases we have simulated). Thus, in this special case, we do not use transport-layer FEC. By jointly considering error resilient source coding, link-layer FEC, and power adaptation, we formulate a JSCCPA problem given below for video transmission over a wireless link,

$$\begin{aligned} \min_{\{\boldsymbol{\mu}\in\mathcal{Q},\boldsymbol{\nu}\in\mathcal{R},\boldsymbol{\eta}\in\mathcal{P}\}} E[D] &= \sum_{k=1}^{M} E[D_k(\boldsymbol{\mu},\boldsymbol{\nu},\boldsymbol{\eta})] \\ \text{s.t.} \quad C &= \sum_{k=1}^{M} B_k(\mu_k,\nu_k)P_k(\eta_k)/R_T \le C_0 \\ T &= \sum_{k=1}^{M} B_k(\mu_k,\nu_k)/R_T \le T_0. \end{aligned} \tag{5.5}$$

Note that in this case, we have $\rho_k(\mu_k,\nu_k,\eta_k) = \beta_k(\mu_k,\nu_k,\eta_k)$, which means that the probability of loss for one packet depends only on the parameters selected for this packet. Consequently, the expected distortion for one packet depends only on the parameters selected for this packet and its previous packet, based on (2.15).

5.4 Solution Algorithm

In this section, we present solutions for (5.4) and (5.5) based on Lagrangian relaxation. Depending on the complexities of the inter-packet dependencies, the resulting two minimization problems can be efficiently solved using an iterative descent algorithm that is based on the method of alternating variables for multivariate minimization [141] and deterministic dynamic programming, respectively.

5.4.1 Lagrangian Relaxation

First, we formulate a Lagrangian dual for (5.4) by introducing Lagrange multipliers, $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$, for the transmission energy and delay constraints, respectively. The resulting Lagrangian is

$$L(\boldsymbol{\mu}, \gamma, \boldsymbol{\nu}, \boldsymbol{\eta}, \lambda_1, \lambda_2) = E[D] + \lambda_1(C - C_0) + \lambda_2(T - T_0) \tag{5.6}$$

and the corresponding dual function is

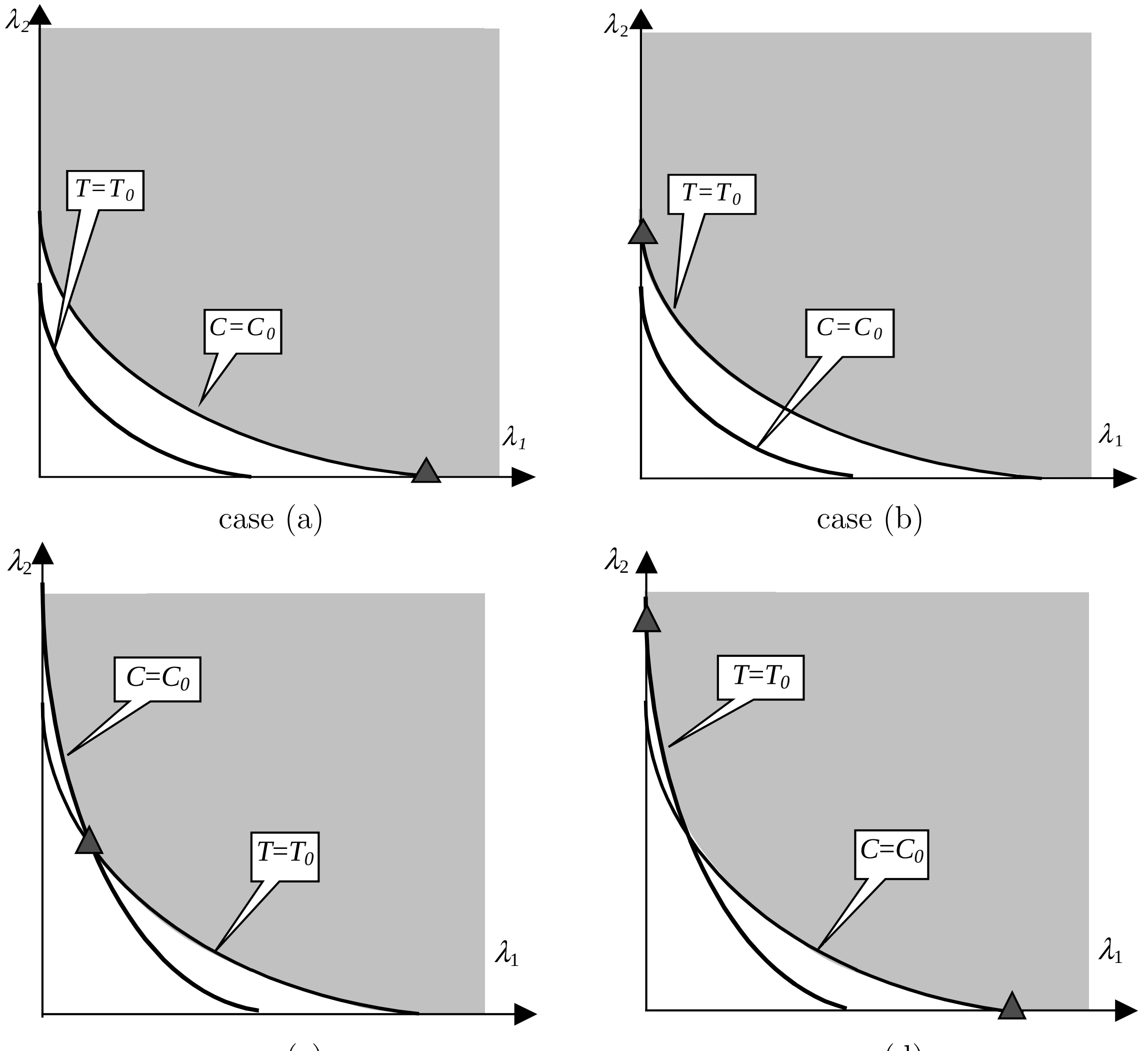
$$g(\lambda_1, \lambda_2) = \min_{\{\boldsymbol{\mu}\in\mathcal{Q}, \gamma\in\Gamma, \boldsymbol{\nu}\in\mathcal{R}, \boldsymbol{\eta}\in\mathcal{P}\}} L(\boldsymbol{\mu}, \gamma, \boldsymbol{\nu}, \boldsymbol{\eta}, \lambda_1, \lambda_2). \tag{5.7}$$

Note that the Lagrangian may not be separable because the distortion for the k-th packet, $E[D_k]$, may depend on the parameters chosen for the other packets. The dual problem to (5.4) is then given by

$$\max_{\lambda_1\geq 0, \lambda_2\geq 0} g(\lambda_1, \lambda_2). \tag{5.8}$$

Solving (5.8) will provide a solution to (5.4) within a convex hull approximation. Assuming we can evaluate the dual function for a given choice of λ_1 and λ_2, a solution to (5.8) can be found by choosing the correct Lagrange multipliers. This can be accomplished by using a variety of methods such as cutting-plane or sub-gradient methods [142]. Alternatively, based on the observed structure of this problem, we propose the following heuristic approach, which is considerably more efficient than the above-mentioned methods.

Figure 5.2 illustrates four possible cases of the energy contour $C = C_0$ and the delay contour $T = T_0$ in the $\lambda_1 - \lambda_2$ plane, where C_0 and T_0 are the transmission energy and transmission delay constraints for one frame, respectively. The shaded



case (a)

case (b)

case (c)

case (d)

Figure 5.2: Four cases of cost and delay contours.

area indicates the valid choices of (λ_1, λ_2) that satisfy both constraints. The triangle point in each figure represents the location of the optimal solution. To derive our heuristic, let us assume that this problem has no duality gap. In this case, from complementary slackness, the optimal solution must lie at one of the points where the contours intersect the axis or at the intersection of the contours[2]. In Fig. 5.2, we show the contours intersecting at only one point; this is the only case we have observed in practice, and we assume that it is true in the following. Let $H \in \mathcal{Q} \times \Gamma \times \mathcal{R} \times \mathcal{P}$, then the appropriate (λ_1, λ_2) can be obtained using the algorithm described below.

Step 1 (*case a, d*): Let $\lambda_2 = 0$, find the largest λ_1^* such that $C(H(\lambda_1^*, 0)) \leq C_0$. If $T(H(\lambda_1^*, 0)) \leq T_0$, $H(\lambda_1^*, 0)$ is the solution. Otherwise,

Step 2 (*case b, d*): Let $\lambda_1 = 0$, find the largest λ_2^* such that $T(H(0, \lambda_2^*)) \leq T_0$. If $C(H(0, \lambda_2^*)) \leq C_0$, $H(0, \lambda_2^*)$ is the solution. Otherwise,

Step 3 (*case c*):

i. Let $\lambda_1^l = 0$, $\lambda_1^r = \lambda_1^*$, $\lambda_2^b = 0$, $\lambda_2^t = \lambda_2^*$ (where λ_1^* and λ_2^* are given in steps 1 and 2).

ii. Let $\lambda_1^m = (\lambda_1^l + \lambda_1^r)/2$, find λ_2^* within $[\lambda_2^b, \lambda_2^t]$ to satisfy $T(H(\lambda_1^m, \lambda_2^*)) \leq T_0$.

iii. If $C(H(\lambda_1^m, \lambda_2^*)) > C_0$, then let $\lambda_1^l = \lambda_1^m$, $\lambda_2^t = \lambda_2^*$, and go to step 3ii. Otherwise,

iv. If $C(H(\lambda_1^m, \lambda_2^*)) < C_0 - \delta$ (δ is a relatively small number), then let $\lambda_1^r = \lambda_1^m$, $\lambda_2^b = \lambda_2^*$, and go to step 3ii. Otherwise,

v. Let the solution be $H(\lambda_1^m, \lambda_2^*)$.

In the proposed solution, when one Lagrange multiplier is fixed, the dual problem becomes a one-dimensional problem, which can be easily solved by standard

[2] If we consider the convex hull of the primal problem (i.e. we take the convex hull of the constraint set); then if Slater's condition is satisfied, this problem will have no duality gap.

convex search techniques, such as the bisection method [136]. Note that in cases (a) and (b), one of the constraints is inactive. Case (d) indicates that different combinations of (λ_1, λ_2) may result in the same minimum distortion. Next, we consider evaluating the dual function in (5.7), given appropriate λ_1 and λ_2.

5.4.2 Minimization of Lagrangian

In (5.4), for given Lagrange multipliers, minimizing the resulting Lagrangian itself is still complicated due to the fact that the loss probability of one source packet depends on the operational parameters chosen for all the other packets. Hence, we solve the minimization problem by an iterative descent algorithm that is based on the method of alternating variables for multivariate minimization [141]. To be precise, for each RS code $\gamma \in \Gamma$ (i.e., we do an exhaustive search for γ), the RS block size is $N(\gamma)$, which is also the number of total transport packets in a frame. Then by adjusting one set of operational parameters for one packet at a time, while keeping constant those for the other packets until convergence, we can minimize the Lagrangian, $L(\boldsymbol{\mu}, \gamma, \boldsymbol{\nu}, \boldsymbol{\eta}, \lambda_1, \lambda_2)$ in (5.6). In particular, let $x_k = \{\mu_k, \nu_k, \eta_k\}$ denote the vector of the source coding, intra-FEC channel coding, and power level selected for the k-th packet, and $\boldsymbol{x} = \{x_1, x_2, ..., x_N\}$ denote the parameters selected for the N packets[3]. Let $\boldsymbol{x}^{(t)} = \{x_1^{(t)}, x_2^{(t)}, ..., x_N^{(t)}\}$, for $t = 0, 1, 2, ...$, be the parameter vector selected by optimization at step t, where $\boldsymbol{x}^{(\mathbf{0})}$ corresponds to any initial parameter vector selected for the N packets. This can be done in a round-robin style, e.g., let $t_n = (t \bmod N)$. If $i \neq t_n$, let $x_i^{(n)} = x_i^{(n-1)}$. Otherwise, for $i = t_n$, the following

[3] Note that for $k > M$, there is no associated source coding parameter μ_k defined, because these packets are parity packets. However, the number of source bits in those parity packets is determined by the maximum of the source bits in the source packets.

optimization is carried out

$$x_i^{(t)} = arg \min_{\boldsymbol{x}^{(t)}} L(x_1^{(t)}, ..., x_{i-1}^{(t)}, x_i, x_{i+1}^{(t)}, ..., x_N^{(t)}, \lambda_1, \lambda_2). \tag{5.9}$$

The optimal operational parameter vector $\boldsymbol{x}^{(t)}$ is updated until the Lagrangian $L(\boldsymbol{x}^{(t)}, \gamma, \lambda_1, \lambda_2)$ converges. Convergence is guaranteed because the Lagrangian is non-increasing and bounded below [62,135]. In fact, in our simulations, we have observed that it only takes two to three iterations for the Lagrangian to converge. The computational complexity mainly comes from the calculation of (5.3), which depends on the block size of the RS code, $N(\gamma)$. Note that the above iterative algorithm generates a set of optimal parameters of $(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\eta})$ for a particular γ. The final optimal parameters $(\boldsymbol{\mu}, \gamma, \boldsymbol{\nu}, \boldsymbol{\eta})$ corresponds to the minimum Lagrangian with one particular γ and its corresponding optimal parameters of $(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\eta})$.

If the special case, i.e., formulation (5.5), is considered, we can accurately (since the global optimal solution is guaranteed) and efficiently minimize the resulting Lagrangian by using DP due to the limited inter-packet dependencies[4]. For simplicity, let C_k and T_k denote the transmission energy and transmission delay for packet k, respectively. The Lagrangian corresponding to formulation (5.5) can be expressed as $L(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\eta}, \lambda_1, \lambda_2) = \sum_{k=1}^{M} J(k)$, where

$$J(k) = E[D_k] + \lambda_1 C_k + \lambda_2 T_k.$$

From (5.1) and (2.7), the cost of each packet $J(k)$ is a function of μ_k, ν_k, η_k and $E[D_{L,k}]$. As shown in (2.15), if we employ the error concealment strategy described

[4] Note that due to the use of inter-packet FEC in (5.4), where the expected distortion for one packet depends on the parameters selected for all the packets in the same frame, DP is not applicable in (5.4).

in Sect. 2.3.6, the cost of the k-th packet can be described as

$$J(k) = J(\mu_{k-1}, \nu_{k-1}, \eta_{k-1}, \mu_k, \nu_k, \eta_k).$$

The dual can then be evaluated via dynamic programming. The time complexity[5] of this scheme is $O(M \cdot |\mathcal{Q} \times \mathcal{R} \times \mathcal{P}|^2)$ [136]. Note that if a simpler error concealment scheme is used, i.e., the lost MB is recovered from the MB with the same spatial location in the previously reconstructed frame, the cost for packet k is in the form of

$$J(k) = J(\mu_k, \nu_k, \eta_k),$$

resulting in a time complexity of $O(M \cdot |\mathcal{Q} \times \mathcal{R} \times \mathcal{P}|)$.

5.5 Experimental Results

In our simulations, we choose an H.263+ codec to perform source coding, and consider the QCIF Foreman test sequence at a frame rate of 30 fps. In all experiments, we set RTT$= 4T_F$, which is long enough to preclude retransmissions in this setting.

We use an RCPC code with generator polynomials (133, 171), mother code rate 1/2, and puncturing rate $G = 4$. This mother rate is punctured to achieve the 4/7, 2/3, and 4/5 rate codes. At the receiver, soft Viterbi decoding is used in conjunction with BPSK demodulation. We present experiments on Rayleigh fading channels, and the channel parameter is defined as SNR $= a\frac{E_b}{N_0}$, where a is the expected value of the square of the Rayleigh distributed channel gain. In the simulations, the bit error rates for the Rayleigh fading with the assumption of ideal interleaving were obtained

[5] Note that here we only discuss the time complexity for evaluating the dual given particular Lagrange multipliers. The total time complexity depends on the number of times needed to find the correct Lagrange multipliers.

experimentally using simulations, as shown in Table 5.1. The method for simulation can be found in [77, 78].

SNR (dB)	2	6	10	14	18	22
cr=1/2	1.4×10^{-3}	2.2×10^{-5}	2.1×10^{-6}	2.4×10^{-7}	6.4×10^{-8}	2.8×10^{-9}
cr=4/7	1.1×10^{-1}	5.3×10^{-4}	4.1×10^{-5}	1.1×10^{-5}	3.8×10^{-6}	1.3×10^{-6}
cr=2/3	3.2×10^{-1}	7.4×10^{-3}	1.7×10^{-4}	3.5×10^{-5}	1.2×10^{-5}	4.2×10^{-6}
cr=4/5	4.2×10^{-1}	4.0×10^{-2}	6.6×10^{-4}	1.1×10^{-4}	3.6×10^{-5}	1.2×10^{-5}

Table 5.1: Performance of RCPC over a Rayleigh fading channel with interleaving (cr denotes channel rate).

5.5.1 Video Transmission over Hybrid Wireless Networks

We first evaluate the performance of the proposed PFEC on a hybrid wireless network, which consists of both wired and wireless links. We fix the transmission power in this study. This is mainly due to the high computation complexity in calculating (5.4) if all those operational components are included. In addition, this simplified case allows us to better analyze the potential of the proposed PFEC approach in providing UEP. For the transport-layer inter-packet FEC, we choose $\Gamma = \{(9,9),(11,9),(13,9),(16,9)\}$ as the available RS coding set. Longer blocks not only complicate the computation of (5.3), but also introduce longer delays. The transmission rate is set as $R_T = 360$ kbps in all simulations of this subsection.

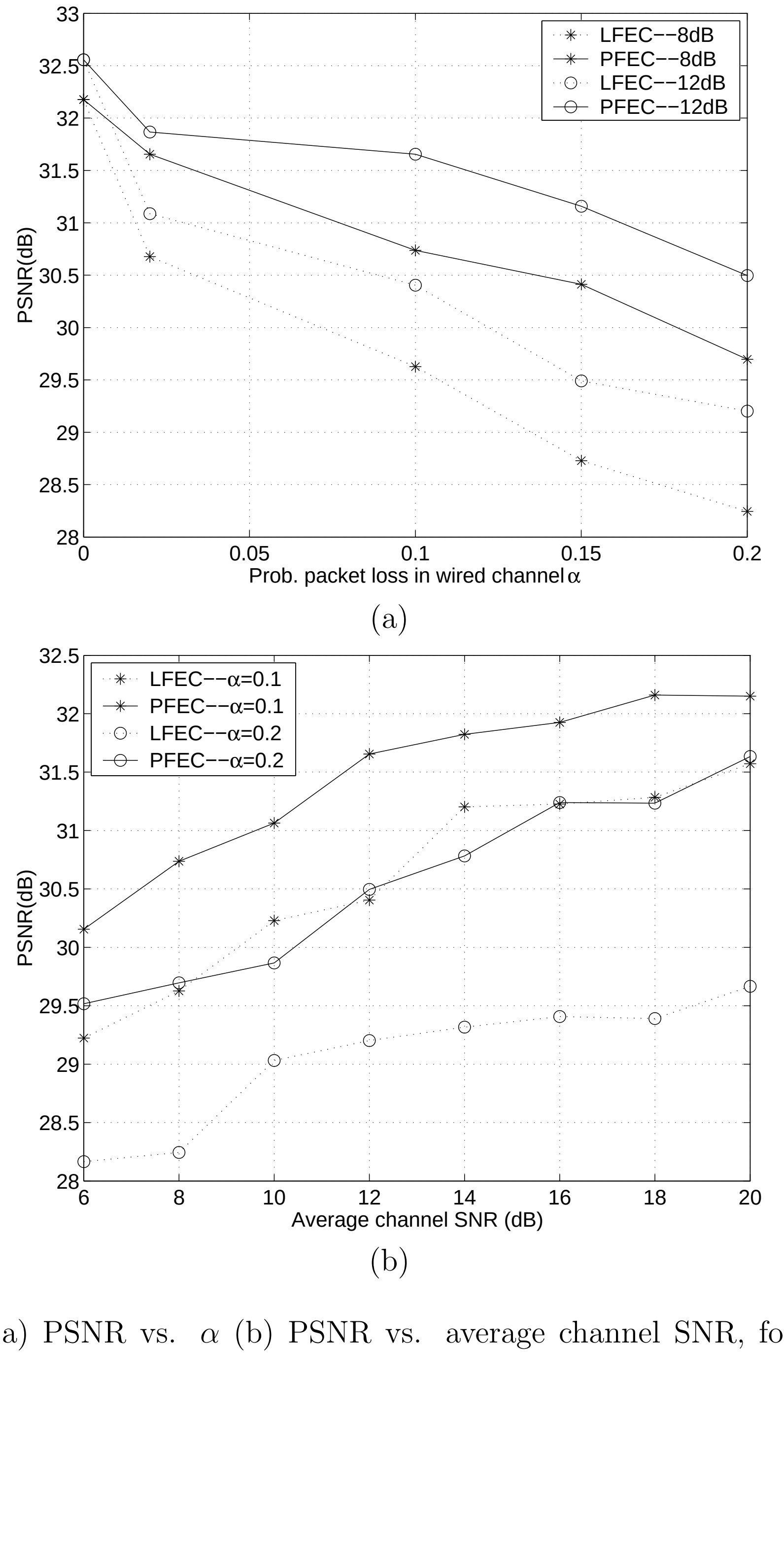
Product FEC vs. Link-Layer FEC

In this experiment, we compare the performances of two UEP systems: i) the UEP product code FEC (UEP-PFEC) and ii) pure link-layer FEC (UEP-LFEC). The goal is to illustrate the advantage of using PFEC. Both systems are UEP optimized using (5.4), where the PFEC system allows transport-layer RS coding but the LFEC system does not. The two systems have the same energy and transmission delay constraints.

We illustrate the performance of the two systems in Fig. 5.3, where we plot the average decoded PSNR for various average SNR values in the wireless link and packet loss rates α in the wired link. As shown in Fig. 5.3, with the above simulation setup, when α is small, LFEC is close to PFEC. However, as the wired link gets worse, PFEC starts to outperform LFEC by up to 2.5 dB. This improved performance is due to the use of cross-packet protection in the transport layer. Table 5.2 shows how link-layer FEC rates are selected in the two systems. As can be seen, as the channel SNR improves, less link-layer protection is needed, i.e., higher channel rates are used. Second, compared to the LFEC approach, the PFEC approach uses lower rate codes of link-layer FEC because of the overhead from the transport-layer FEC. In addition, we can see in the table that the link-layer FEC rates do not follow the change of α. This implies that the link-layer FEC is apparently less efficient than transport-layer FEC in reacting to packet losses in the wired link, because the link-layer FEC does not provide inter-packet protection. Another observation from Fig. 5.3(a) is that when $\alpha = 0$, which corresponds to the case where the wired link is error free, the inter-packet FEC in the transport layer becomes unnecessary and thus the optimized PFEC is equivalent to LFEC.

UEP vs. EEP

In the second experiment, we illustrate the advantage of UEP by comparing the performance of two PFEC systems: i) UEP product FEC (UEP-PFEC) and ii)



(a)

(b)

Figure 5.3: (a) PSNR vs. α (b) PSNR vs. average channel SNR, for PFEC and LFEC.

α	SNR (dB)	Link-layer FEC rates in percentage in UEP-PFEC				Link-layer FEC rates in percentage in UEP-LFEC			
		cr=1/2	cr=4/7	cr=2/3	cr=4/5	cr=1/2	cr=4/7	cr=2/3	cr=4/5
0.1	6	28.9	68.9	0.7	1.5	94.8	5.2	0	0
	8	26.7	5.2	68.1	0	63	8.9	28.1	0
	10	3	0	89.6	7.4	28.9	0	71.1	0
	12	0.7	0	80.8	18.5	7.4	0	91.1	1.5
	16	0.7	0	23.7	75.6	0	0	80	20
0.2	6	54.1	45.9	0	0	91.1	8.9	0	0
	8	17.8	10.4	71.8	0	47.4	11.9	40.7	0
	10	2.2	0	97.8	0	23.7	0	76.3	0
	12	2	0	98	0	3	0	94	3
	16	1.5	0	11.8	86.7	0	0	60	40

Table 5.2: Link-layer FEC rates in percentage in the UEP-PFEC and UEP-LFEC system (cr denotes channel rate).

EEP product FEC (EEP-PFEC). Both systems use a product FEC and are optimized within (5.4). The difference is that the EEP system has fixed link layer FEC, while the link layer FEC in the UEP system is optimally employed. For the two systems, we plot in Fig. 5.4 the average decoded PSNR under different average channel SNR. It can be seen that UEP-PFEC achieves the upper bound of all EEP-PFEC systems, and outperforms the best of all EEP systems by around 0.2 dB at all channel conditions. The gain comes from the higher flexibility of the UEP-PFEC approach, where link-layer coding parameters can be optimally assigned to different packets to achieve UEP for video packets that are of different importance.

5.5.2 Video Transmission over Wireless Links

In this study, we consider video transmission over a single wireless link. This can be regarded as a special case of hybrid wireless networks where the wired link is error and delay free. In the study above, we have shown that when the wired

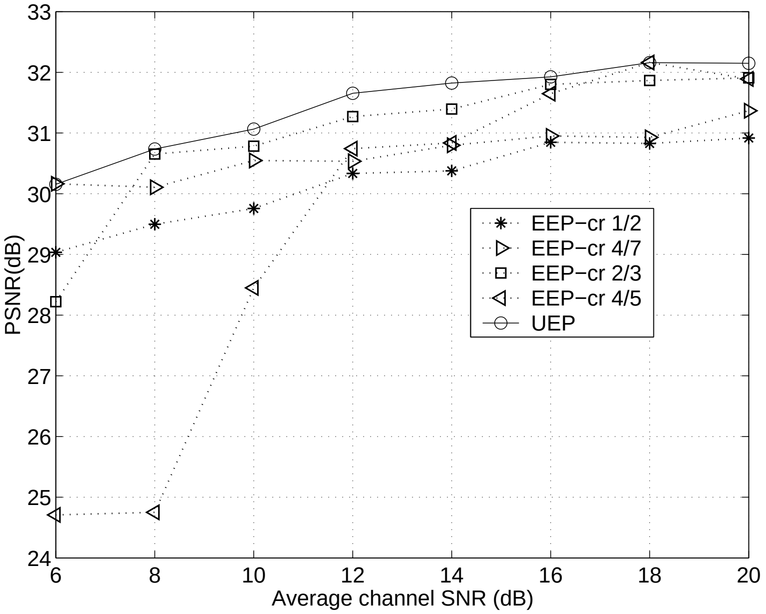


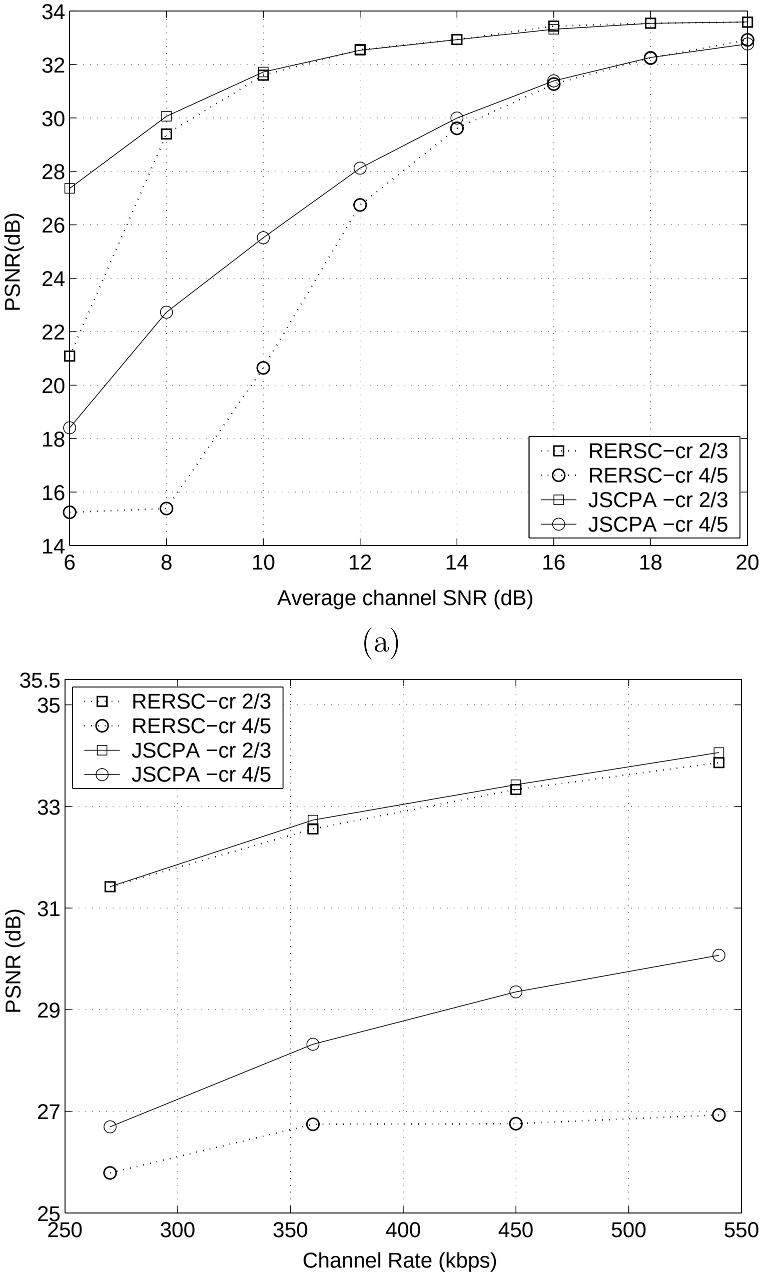
Figure 5.4: PSNR vs. average channel SNR (α=0.1), for UEP and EEP.

link has no errors ($\alpha = 0$), transport-layer inter-packet FEC is not necessary; thus it is omitted, which makes the computations much more efficient. The goal of this subsection is to study the effectiveness of channel coding (intra-packet FEC) and power adaptation in achieving the optimal UEP. We focus on the trade-off of the two adaptation components.

Performance Comparison of JSCPA and RERSC Systems

In this experiment, we compare the performance of two systems: the proposed framework in (5.5) with fixed channel coding rate, which is referred to as JSCPA (joint source coding and power adaptation) system, and an RERSC (reference error resilient source coding) system which uses a fixed channel coding rate and transmission power. The transmission power levels can be adapted depending on the CSI and source content in the JSCPA system, but is fixed in the RERSC system. We refer to the RERSC system as the reference system, and evaluate it under different channel SNR (referred to as reference channel SNR) to generate the energy constraints for the JSCPA system. Thus, the two systems have the same transmission delay constraints and use the same amount of transmission energy.

We illustrate the performance of the two systems in Fig. 5.5(a), which shows the average decoded PSNR for the Foreman sequence under different channel SNR when $R_T = 360$ kbps, and Fig. 5.5(b), where we plot the average decoded PSNR versus transmission rate when the reference channel SNR is 12 dB. As shown in Fig. 5.5, by adjusting the power levels, the JSCPA system achieves a significant gain over the RERSC system. When the channel SNR is small, e.g., 8 dB, the gain can be as large as 6 dB in PSNR. The gain comes from the higher flexibility of the JSCPA approach, where the power level can be optimally assigned to different packets to achieve UEP for video packets that are of different importance. In addition, from Fig. 5.5, we can see that under some settings, JSCPA achieves little gain over RERSC [e.g., when the channel SNR is 12 dB and the channel coding rate used is low, as shown in Fig. 5.5(b)]. This observation can help us assess the effective components in designing a practical video streaming system. Table 5.3 shows how transmission power is optimally selected for transmitting video packets in the proposed JSCPA system. The values inside the parentheses denote the percentage of packets with transmission power level 1, 2, 3, 4, 5, respectively. Note that the power level parameters 1, 2, 3, 4 and 5 are simplified substitutes for the actual transmission power values. The actual values are proportional to those parameters.



(a)

(b)

Figure 5.5: JSCPA vs. RERSC (a) PSNR vs. average channel SNR with $R_T = 360$ kbps (b) PSNR vs. transmission rate with reference channel SNR be 12 dB (cr denotes channel rate in the legend).

Reference SNR (dB)	6	12	20
Channel rate=1/2	(2.4,18.5,73.9,5.1,0)	(12.6,32.4,33.9,19.6,1.4)	(62.3,0,12.9,0,24.8)
Channel rate=4/7	(18,0,14.3,66.1,1.6)	(2.3,29.9,56.4,11.0,0.3)	(10,35,39.2,13.4,2.3)
Channel rate=2/3	(40,0,0,13,47)	(0.7,13.9,66.1,18.7.0.6)	(11.6,10.8,69.1,6.9,1.6)
Channel rate=4/5	(45.8,0,0,0,54.2)	(2,4.1,41.8,47.3,4.9)	(8.2,31.5,43.8,15.3,1.3)

Table 5.3: Power level allocation of power level (1,2,3,4,5) in percentage in the JSCPA system (the reference power level is 3).

Performance Comparison of JSCCPA and JSCPA Systems

In the second experiment, we compare the performance of the JSCCPA system as in (5.4) and the JSCPA (joint source coding and power adaptation) system, in which the channel coding parameter is fixed. Note that the two systems have the same transmission delay and energy constraints, which are obtained from the corresponding reference RERSC systems. For the two systems, we plot the average decoded PSNR for the Foreman sequence under different channel SNRs when $R_T = 360$ kbps in Fig. 5.6(a) and at different transmission rates when the reference channel SNR is 12 dB in Fig. 5.6(b). It can be seen that the JSCCPA approach achieves the upper bound of all JSCPA approaches. The gain comes from the higher flexibility of the JSCCPA approach, where channel coding parameters can be optimally assigned to different packets to achieve UEP. Table 5.4 shows how channel coding rates are selected by the JSCCPA system. As can be seen, as the channel SNR improves, less channel protection is needed.

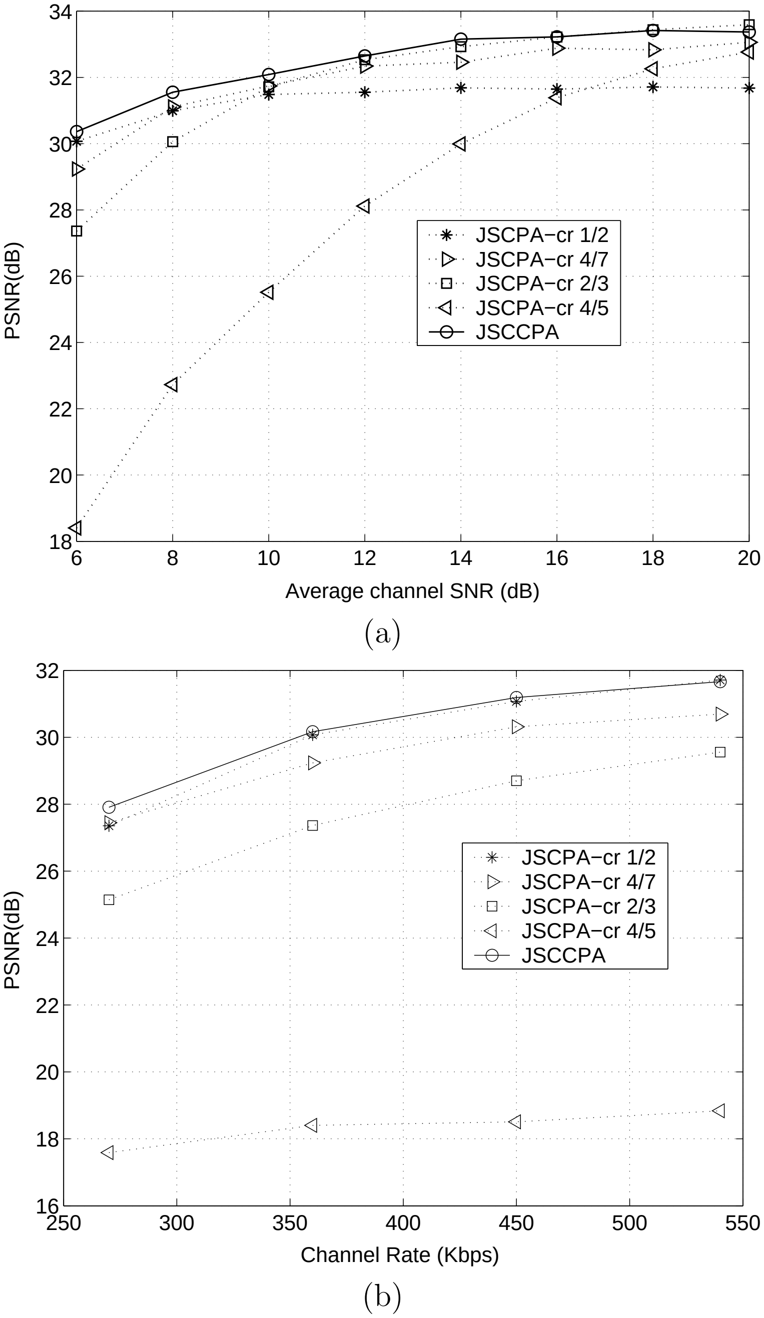


Figure 5.6: JSCCPA vs. JSCPA (a)PSNR vs. average channel SNR with $R_T = 360$ kbps (b) PSNR vs. channel transmission rate with the reference channel SNR be 12 dB (cr denotes channel rate in the legend).

Reference SNR (dB)	6	8	10	12	14	16	18	20
Channel rate=1/2	96.2	67.7	41.2	19.6	6.7	4.7	1.0	1.6
Channel rate=4/7	3.8	31.9	57.3	69.6	61.3	35.0	17.8	5.6
Channel rate=2/3	0	0.4	1.5	10.8	31.3	57.7	73.9	69.6
Channel rate=4/5	0	0	0	0	0.7	2.6	7.3	23.2

Table 5.4: Channel coding rates in percentage in JSCCPA system.

5.6 Conclusions

We have studied cross-layer resource allocation for energy efficient video communications over a hybrid wireless/wire-line network. The network resources are optimally allocated according to the joint source-channel coding and power adaptation framework, where various error control components, such as error resilient source coding, transport-layer FEC, link-layer FEC, power adaptation and error concealment, are jointly considered. Our focus is on the channel coding and power adaptation.

We first demonstrated the outstanding performance of the proposed PFEC which can provide optimal UEP for video streams. Next, through simulations on a hybrid wireless network, we showed that transport-layer FEC is not necessary if the wired link has no error, based on our simulation setups. We also demonstrated the advantage of jointly adapting the link-layer FEC and transmission power to the varying wireless channel conditions. In addition, as another contribution, the proposed algorithm, which is based on Lagrangian relaxation, can be used to tackle other discrete optimization problems with two constraints.

Chapter 6

Joint Source Coding and Packet Classification for Video Transmission over DiffServ Networks

Differentiated Services (DiffServ) is one of the leading architectures for providing quality of service in the Internet. In this chapter, we consider the problem of transmitting video over a DiffServ network, where the available error control components include error resilient source coding, channel coding, QoS support in the transport network and error concealment. We focus on the joint consideration of error resilient source coding at the video encoder and differentiated QoS for each packet through network-layer packet prioritized transmission. The encoding parameter selection and packet classification are both used to manage end-to-end delay variations and packet losses within the network. Our goal is to minimize the total end-to-end distortion while meeting cost and delay constraints, or, alternatively, to minimize the total cost given the distortion and delay constraints.

6.1 Introduction

DiffServ is a Layer 3 (IP layer) approach for supporting QoS by discriminately allocating resources to aggregated traffic flows according to service classes. Specifically, each packet is assigned a priority tag (a DS "code-point"), indicating a QoS class to which the packet belongs. Upon arriving at a router, a packet is queued and forwarded based on its assigned class. As a consequence, a router can provide different "per hop behaviors" to each aggregated traffic class. These per hop behaviors lead to an end-to-end statistical differentiation between the QoS of each class [14, 15, 104].

The DiffServ network considered here uses pricing to achieve efficient network resource utilization. This is based on the assumption that the service level can be prespecified in the service level agreement (SLA) between the ISP and the users [14, 15]. In this setting, a cost is associated with each service class as specified in the SLA. By adjusting the price for each service class, the network can influence the class that a user selects. Typically, transmitting a packet with a higher priority service class results in a higher cost but a better QoS (lower delay and loss probability). The sender has to classify each packet according to its importance in order to better utilize the available network resources.

Several related approaches have been discussed in the literature. In [128], an adaptive packet forwarding mechanism is proposed for a DiffServ network where video packets are mapped onto different DiffServ service levels. However, the framework in [128] does not incorporate video source coding decisions. The authors in [129] present a rate-distortion optimized packet marking technique to deliver MPEG2 video sequences (only INTRA frames were used in this work) in a DiffServ IP network. Their goal is to minimize the bandwidth consumption in the premium class while achieving nearly constant perceptual quality. This work is extended by taking into account inter-frame motion compensation in [130]. Neither [129] nor [130] considers the selection of source coding parameters. In [62] and [9], cost-distortion optimized multimedia streaming over DiffServ networks is studied. Although the proposed framework in [62] and [9] is very general, it is based on pre-encoded media. Thus, the selection of encoding parameters is not considered. In addition, error concealment is not included. The work presented in this chapter builds on the earlier work in [29], which considers cost-distortion optimized streaming in a simpler setting. The main results of this chapter have been published in [131, 143].

We study the problem of joint adaptation of source coding and packet priority assignment in the resource-distortion optimization framework. In this work, we consider the random network delay for each packet; this delay is managed through selecting the source coding parameters and packet priority. More specifically, finer quantizers lead to higher video reconstruction quality but longer delay (given the same QoS class), which results in higher loss probability. In addition, the packet QoS class also needs to be selected in a way to properly balance cost, delay, and video quality. For example, packets that are hard to conceal but can be easily encoded should use coarser quantizers and a higher QoS class. Packets that are easily concealable can be sent using a lower QoS class. For packets that are hard to encode, the best choice may be a finer quantizer and a higher QoS. The goal is to minimize the end-to-end distortion subject to cost and delay constraints, or alternatively, to minimize the overall cost given end-to-end distortion and delay constraints. Such a goal is achieved through joint selection of source coding parameters and QoS classes to optimally balance the received video quality and the overall cost.

The rest of this chapter is organized as follows. In Sect. 6.2, we describe the encoder buffer behavior model used in this work for DiffServ. In Sect. 6.3, we present problem formulations for both the minimum distortion approach and the minimum cost approach. Section 6.4 provides a detailed description of the algorithm used to solve the optimization problem. Simulation results and discussion are reported in Sect. 6.5. We draw conclusions in Sect. 6.6.

6.2 Preliminaries

6.2.1 DiffServ Traffic Classes

Each video packet is classified into one of the available DiffServ traffic classes before being transmitted over the network. There are several parameters associated with each class. First, each class has a specified packet loss probability and a packet delay distribution. These parameters may be specified in a service level agreement (SLA) between the Internet service provider (ISP) and the users [15,128] or estimated via feedback from the network (e.g., using RTCP). Also, each class is associated with a price per bit transmitted. These prices can also be prespecified in an SLA and are used by the service provider to achieve efficient network resource utilization.

Each class also has a “rate” constraint limiting the amount of traffic that can be sent in the class. This can again be specified in the SLA and enforced at the ingress routers. Alternatively, the rate constraint could be estimated via feedback from the network as in TCP-Friendly protocols [8,62]. In the experiments in Sect. 6.5 we use a model-based TCP-friendly congestion controller, where the throughput per class is estimated using a stochastic TCP model based on the steady-state packet loss probability and round-trip time [8,61,62]. Such rate constraints limit the throughput per class over a specified time-scale. A user can ensure that its traffic confirms to this constraint by using a traffic regulator, such as a Token Bucket algorithm.

In order to develop a tractable optimization model, we assume that all encoded video packets (for every class) are transmitted FIFO from a single buffer. To model the rate constraint in this setting, we assume that a higher level controller specifies an *effective rate*, $R(\pi)$ for each class π. When a packet containing B bits is transmitted, the next packet cannot be sent until $B/R(\pi)$ seconds later (independently of the class of the next packet). Thus, in this context $R(\pi)$ should not be interpreted as the rate constraint for class π, but as a parameter that is adjusted by a congestion controller to ensure that any rate constraint is satisfied over a longer horizon. The actual transmission time of the packet will be B/R_l, where R_l is the link transmission rate which is typically much greater than $R(\pi)$. Thus $B/R(\pi) - B/R_l$ is the time during which the transmitter is idle (this models, for example, the time the transmitter must wait for a token to be available with a Token Bucket regulator). During a given interval of time, this ensures that the average rate of packets in class π is upper bounded by $R(\pi)$, where this bound is approached if every packet is in class π. When packets are sent from multiple classes over a time-interval, then the upper bound $R(\pi)$ may not be met, due to the transmission of packets from the other classes. If the rate constraint for a given class is not met, then the higher level controller can adjust the effective rate for that class for future frames (depending on the time-scale over which the rate constraint is defined).

An alternative model would be to view the system as having one transmission buffer for each class, where each buffer is served at the effective rate for that class. Though, in principle, the following framework can be extended to this setting, there are two difficulties with this approach: First, the packets in the actual system are all transmitted over the same link, which a multiple buffer model does not capture. Second and more importantly, the complexity of the resulting optimization problem increases exponentially with the number of queues and quickly becomes intractable. The single buffer model greatly simplifies the problem and also has the benefit of ensuring that all packets are more evenly spaced out over time. Also, we emphasize that we are discussing the optimization model. The actual system need not transmit packets in this manner, as long as the model provides a bound on the resulting performance.

6.2.2 Encoder Buffer Behavior Model

As discussed in Sect. 6.2.1, when leaving the encoder buffer, packets are transmitted at the link transmission rate, R_l, but the transmitter is required to idle according to the effective transmission rate of the given service class. Thus, from the point of view of the controller, we model packets from each QoS class, π_k, as being actually transmitted at the effective transmission rates, $R(\pi_k)$. Thus, a packet containing B_k bits has an effective transmission time of $B_k/R(\pi_k)$, as shown in Fig. 6.1. [1]

[1] Note, in calculating the delay, we assume that the idle time occurs before the actual packet transmission. The model could easily be modified for the case where the idle time follows the packet

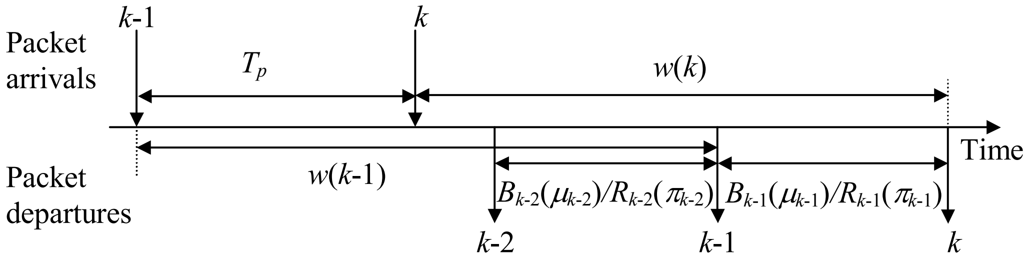


Figure 6.1: Model of packet transmission behavior in the encoder buffer. The length of each block corresponds to the transmission time of the packet.

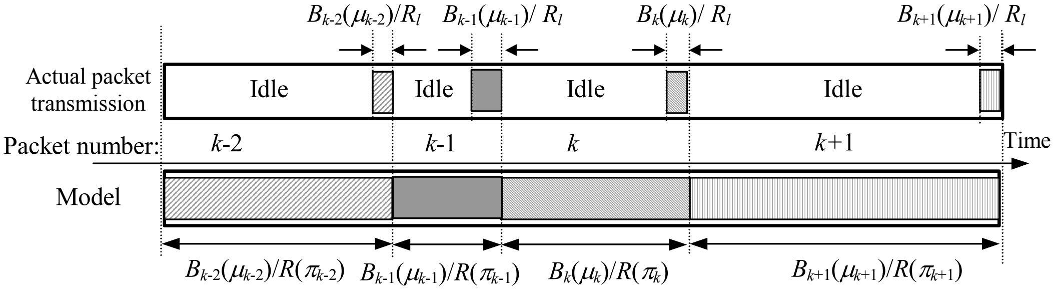


Figure 6.2: Illustration of the buffer delay calculation for each packet. The top arrows indicate the time at which a packet arrives at the encoder buffer, the lower arrows indicate the time at which the packet departs (including transmission time).

Next, based on this model, we specify the maximum allowable network delay for each packet, $\tau(k)$, i.e., if the network delay $\Delta T_n(k) > \tau(k)$, the packet will be lost due to excessive delay. Let $w(k)$ be the waiting time in the encoder buffer for the k-th packet before it is transmitted. The encoder buffer delay (waiting time plus transmission delay) for the k-th packet $\Delta T_{eb}(k)$ can be written as

$$\Delta T_{eb}(k) = w(k) + \frac{B_k(\mu_k)}{R_k(\pi_k)}, \tag{6.1}$$

where $B_k(\mu_k)$ and $R_k(\pi_k)$ are the packet length in bits and the transmission rate in bits per sec for packet k respectively. The k-th packet has a particular class $\pi_k \in \Pi$

transmission.

and coding parameter $\mu_k \in \mathcal{Q}$, where Π and $\mathcal{Q}$ are the set of available service classes and source coding parameters, respectively. Based on the discussion in Sect. 2.1.2, from (2.3), the maximum allowable network delay for packet k is then

$$\tau(k) = T_{max} - w(k) - \frac{B_k(\mu_k)}{R_k(\pi_k)}. \tag{6.2}$$

As shown in Fig. 6.2, the waiting time for the k-th packet can be recursively calculated as

$$w(k) = w(k-1) + \frac{B_{k-1}(\mu_{k-1})}{R_{k-1}(\pi_{k-1})} - T_p, \tag{6.3}$$

where the constant T_p is the processing time for both encoding and decoding a packet.

6.3 Problem Formulation

The first problem we consider is how to provide the best video quality (minimum end-to-end distortion) for given cost and delay constraints. We refer to this as the "minimum distortion problem". This problem can be formulated as

$$\begin{aligned} &\min_{\{\boldsymbol{\mu}\in\mathcal{Q},\boldsymbol{\pi}\in\Pi\}} \sum_{k=1}^{M} E[D_k(\boldsymbol{\mu},\boldsymbol{\pi})] \\ \text{s.t.} \quad &\sum_{k=1}^{M} c_k(\pi_k)B_k(\mu_k) \le C_0^{(i)} \\ &\sum_{k=1}^{M} B_k(\mu_k)/R_k(\pi_k) \le T_0^{(i)}, \end{aligned} \tag{6.4}$$

where $c_k(\pi_k)$ is the cost per bit of the k-th packet, M is the number of packets in a frame, $C_0^{(i)}$ and $T_0^{(i)}$ are the total cost constraint and transmission delay constraint for the i-th frame, respectively. We use $\boldsymbol{\mu} = \{\mu_1, \mu_2, ..., \mu_M\}$ and $\boldsymbol{\pi} = \{\pi_1, \pi_2, ..., \pi_M\}$ to denote the coding parameters and QoS classes for all the packets in a frame, respectively. The transmission delay constraint can be obtained from a higher-level rate controller, and may vary from frame to frame depending on how difficult it is to encode. Assuming $T_{rc}^{(i)}$ is the corresponding transmission delay budget for the i-th frame, $T_0^{(i)}$ can be recursively obtained as

$$T_0^{(i)} = \begin{cases} T_{rc}^{(0)} & \text{if } i = 0 \\ T_{rc}^{(i)} + T_0^{(i-1)} - \sum_{k=1}^{M} \frac{B_k^{(i-1)}(\mu_k)}{R_k^{(i-1)}(\pi_k)} & \text{if } i \geq 1 \end{cases}, \tag{6.5}$$

where $B_k^{(i-1)}$ and $R_k^{(i-1)}$ denote the bits and transmission rate for the k-th packet in frame $i-1$ (Notice that for simplicity, we have not been using the frame index for B_k and R_k in the remaining of the chapter.) Therefore, the smaller the transmission delay for a frame, the greater the time left for the next one to reach the receiver before the deadline set by the rate controller.

An alternative way to provide the most efficient service at a given cost affordable to the client is to minimize the expected end-to-end distortion with given cost constraint and transmission delay constraint. This problem is referred to as the "minimum cost problem". Formally we consider

$$\begin{aligned} & \min_{\{\boldsymbol{\mu}\in\mathcal{Q},\boldsymbol{\pi}\in\Pi\}} \sum_{k=1}^{M} E[c_k(\pi_k)B_k(\mu_k)] \\ \text{s.t.} \quad & \sum_{k=1}^{M} E[D_k(\boldsymbol{\mu},\boldsymbol{\pi})] \leq D_0^{(i)} \\ & \sum_{k=1}^{M} B_k(\mu_k)/R_k(\pi_k) \leq T_0^{(i)}, \end{aligned} \tag{6.6}$$

where $C_0^{(i)}$ and $T_0^{(i)}$ are the cost constraint and transmission delay constraint for the i-th frame, respectively. Similar to the minimum cost approach, $T_0^{(i)}$ can be recursively obtained from (6.5) to meet the deadline for each frame set by rate controller. Note that (6.6) is a dual problem of (6.4), thus can be solved in the same fashion.

6.4 Solution Algorithm

In this section, we present a solution approach for the minimum distortion problem in (6.4) based on Lagrangian relaxation and deterministic dynamic programming. The minimum cost problem can be solved in the same fashion.

6.4.1 Lagrangian Relaxation

First, by introducing Lagrange multipliers $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ for the cost and delay constraints, respectively, the constrained problem (6.4) can be converted into the unconstrained Lagrangian problem as,

$$\min_{\{\boldsymbol{\mu},\boldsymbol{\pi}\}} L(\boldsymbol{\mu},\boldsymbol{\pi},\lambda_1,\lambda_2) = \sum_{k=1}^{M} \{E[D_k] + \lambda_1 c_k(\pi_k) B_k(\mu_k) + \lambda_2 B_k(\mu_k)/R_k(\pi_k)\} \tag{6.7}$$

The solution of (6.4) can be obtained, within a convex hull approximation, by solving (6.7) with the appropriate choice of Lagrange multipliers, $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$. This can be accomplished by using a variety of methods such as cutting-plane methods, sub-gradient methods [142], or our proposed algorithm in Chapter 5. The solution usually has high computational complexity to tackle such an optimization problem with two Lagrange multipliers.

Alternatively, according to Theorem 1 (see Appendix A), by using only one Lagrange multiplier $\lambda \geq 0$ for the cost constraint, (6.4) can be converted to

$$\begin{aligned} &\min_{\{\boldsymbol{\mu},\boldsymbol{\pi}\}} \sum_{k=1}^{M} \{E[D_k(\boldsymbol{\mu},\boldsymbol{\pi})] + \lambda c_k(\pi_k) B_k(\mu_k)\} \\ &\text{s.t.} \quad \sum_{k=1}^{M} B_k(\mu_k)/R_k(\pi_k) \leq T_0^{(i)} \end{aligned} \tag{6.8}$$

With an appropriate $\lambda \geq 0$, (6.4) can be solved within a convex hull approximation by solving (6.8). Since the objective function in (6.8) is a typical Lagrangian with one Lagrange multiplier, the appropriate $\lambda \geq 0$ can be found by using the bi-section iterative search or other fast search algorithms, according to Theorem 2 and 3 (see Appendix A) [37, 136]. It is worth noting that although (6.8) involves only one Lagrange multiplier, it does not mean that (6.8) is always easier to solve than (6.7) [Actually, in most cases, (6.8) is harder to solve than (6.7).] This is because extra dependency may be introduced by the constraint in (6.8) compared with (6.7), which complicates the optimization. However, if the extra dependency due to the constraint can be reduced, the advantage in solving (6.8) instead of (6.7) may be significant.

For simplicity, let $D(k)$, $C(k)$, and $T(k)$ denote the expected distortion, cost and transmission delay, for packet k, respectively. The Lagrangian in (6.8) can be expressed as $L(\boldsymbol{\mu}, \boldsymbol{\pi}, \lambda) = \sum_{k=1}^{M} J(k)$, where $J(k) = D(k) + \lambda C(k)$.

Next, we consider evaluating $J(k) = D(k) + \lambda C(k)$. Note that this Lagrangian is not separable because the distortion, $D(k)$, depends on the encoding modes and QoS classes chosen for the previous packets. From (2.9), (2.7), and (6.8), the cost of each packet $J(k)$ is a function of π_k, μ_k, $\tau(k)$ and $E[D_{L,k}]$. As shown in (6.2), $\tau(k)$ is a function of $w(k)$. In addition, $w(k)$ is recursively calculated from (6.3). Therefore we have

$$J(k) = J(\mu_1, \pi_1, ..., \mu_{k-1}, \pi_{k-1}, \mu_k, \pi_k), \tag{6.9}$$

i.e., the cost of each packet depends not only on its own coding and priority decision but also on the decisions for all previous packets. Ignoring the delay constraint, optimizing $\sum_{k=1}^{M} J(k)$ can be done via dynamic programming as in the previous chapters.

However, because of the dependencies involved, this essentially results in an exhaustive search through all quantizers and priority choices. The time complexity of such a search is $O(|\Pi \times \mathcal{Q}|^M)$ [136]. Although tree pruning techniques may be used in some cases, e.g., if the unconstrained problem satisfies monotonicity, the complexity reduction is still very limited [37]. In addition, the dependency caused by the transmission delay constraint makes the DP problem more complicated, i.e., fewer branches can be pruned during the tree construction.

In the next section, we present a different DP algorithm, in which the packet waiting time $w(k)$ is used as part of the state description. This allows us to reduce the dependencies in both the objective function (the Lagrangian) and the constraint function in (6.8), making the proposed algorithm much more efficient.

6.4.2 DP Solution

As shown in (2.7), the error concealment scheme introduces dependencies between packets. Specifically, for the particular error concealment scheme discussed in Sect. 2.3.6, $E[D_{L,k}]$ depends on the prediction mode (INTRA/SKIP or INTER) and QoS class π_{k-1}, for the previous packet, through the calculation of ρ_{k-1} and $E[D_{C,k}]$, as shown in (2.15). With the introduction of the packet waiting time $w(k)$, the cost of the k-th packet can be described as

$$J(k) = J(\mu_{k-1}, \pi_{k-1}, \mu_k, \pi_k, w(k)). \tag{6.10}$$

If we evaluate the objective function with respect to $w(k)$, evaluating the Lagrangian $J(k)$ in (6.10) will be much more efficient than in (6.9), due to the reduced dependencies. Note that $E[D_{L,k}]$ does not depend on which quantization parameter is used for the previous packet. This characteristic can be employed in an efficient tree-pruning algorithm.

Next, we discuss the dependency introduced by the transmission delay constraint in (6.8). Recall that now we are evaluating the DP problem with respect to the system state, $w(k)$. Let $s(k)$ be the accumulated transmission delay

$$s(k) = \sum_{j=1}^{k} T(j), \quad \text{for} \quad k = 1, ..., M. \tag{6.11}$$

The constraint (6.8) can be rewritten in terms of $s(k)$, using the fact that $s(k) = s(k-1) + T(k)$. Thus, the accumulated transmission delay up to packet k depends on which path is selected from state 1 to state $k-1$ only through $s(k-1)$ and the transmission delay for packet k, $T(k)$. The constraint in (6.8) is then rewritten as

$$s(k) \leq T_0, \quad \text{for} \quad k = 1, ..., M, \tag{6.12}$$

where for simplicity, we have ignored the frame index, i.e., $T_0(i) = T_0$. Recall that in our solution, the system state includes the packet waiting time. Equation (6.3) can now be written as

$$\begin{aligned} w(k) &= w(k-1) + T(k-1) - T_p = w(1) + \sum_{j=1}^{k-1} T(j) - (k-1)T_p \\ &= w(1) + s(k-1) - (k-1)T_p, \quad \text{for} \quad k = 2, ..., M+1. \end{aligned} \tag{6.13}$$

Therefore, $w(k)$ and $s(k-1)$ have a one-to-one correspondence, and so the dependency in the constraint in (6.8) can be removed by using $w(k)$ as part of the system state. Based on (6.13), (6.12) can be expressed in terms of $w(k)$ as

$$w(k+1) - w(1) + kT_p \leq T_0, \quad \text{for} \quad k = 1, ..., M. \tag{6.14}$$

From the above observations, we propose a DP solution of (6.8) based on the use of $w(k)$ as part of the system state. Our approach is based on a similar technique used in [127].

Since $w(k) \in [0, T_{max}]$ is real-valued, the resulting state space is infinite. For computational complexity consideration, we quantize $w(k)$ into a set of N_W values, $\mathcal{W} = \{w_0, w_1, ..., w_{N_W-1}\}$, with $w_i = (iT_{max})/(N_W - 1)$. Finer quantization of $w(k)$ leads to a better approximation of the optimal solution, at the cost of more computations. The effect of this approximation is to restrict the set of feasible choices for each system state. Therefore, the resulting solution will be a conservative approximation to the optimal solution. The optimization problem can now be re-formulated as

$$\begin{aligned} &\min_{\{u(k)\in\mathcal{U}(w(k))\}} \sum_{k=1}^{M} J_k = \sum_{k=1}^{M} J(\mu_{k-1}, \pi_{k-1}, \mu_k, \pi_k, w(k)) \\ &\text{s.t.} \quad \mathcal{U}(w(k)) = \left\{ u(k) \in \Pi \times \mathcal{Q} : 0 \le \frac{B_k(\mu_k)}{R_k(\pi_k)} + w(k) - T_p \le T_{min} \right\}, \end{aligned} \tag{6.15}$$

where $T_{min} = \min(T_{max}, T_0 + w(1) - kT_p)$ and $\mathcal{U}(w(k))$ is the set of feasible choices, for the k-th packet at state $w(k)$.

Figure 6.3 depicts the directed acyclic graph (DAG) of the state diagram. In this diagram, three stages corresponding to packets k-1 to k+1 are shown. At each stage, the possible system states[2] are represented by an encoder buffer waiting time (N_W =4 in this example). Each branch in the graph corresponds to a choice from $\mathcal{U}(w(k))$ for the packet from which the branch starts. For each choice of $u(k) \in \mathcal{U}(w(k))$, the cost incurred for packet k is given by (6.15). Note that for a given branch, e.g., the branch starting from w_3 at packet k and ending at w_2 at packet $k+1$, the cost associated with this branch is unknown until the encoding mode μ_{k-1}

[2] To be precise, the system state at the k-th stage includes μ_{k-1} and π_{k-1}, but we suppress this to simplify our discussion.

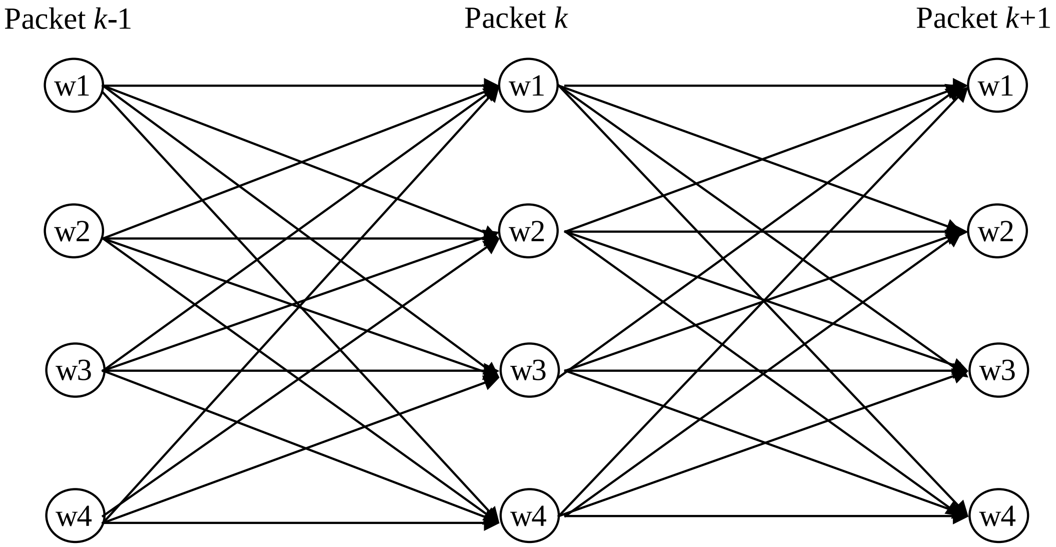


Figure 6.3: DAG of state diagram.

and QoS class π_{k-1} of packet $k-1$ is known. The optimization is achieved by choosing the path through the trellis with the minimum cost among all feasible paths.

Due to the special structure of the objective function of (6.15), the relaxed problem can be solved using techniques from forward DP (i.e., the Viterbi algorithm). From the DP recursion, the M-dimensional optimization problem, which has time complexity $O(|\Pi \times \mathcal{Q}|^M)$, is reduced into M 3-dimensional optimization problems, where M is the number of packets in a frame [136]. Notice that in (6.10), for each $w(k)$, there are at most $|\Pi \times \mathcal{Q}|$ possible choices for (μ_{k-1}, π_{k-1}) and also $|\Pi \times \mathcal{Q}|$ choices for (μ_k, π_k). Thus, the complexity in evaluating (6.15) is $O(M \cdot |\mathcal{W}| \cdot |\Pi \times \mathcal{Q}|^2)$. In addition, because the cost associated with a branch depends only on the prediction mode and QoS class of its previous packet, the solution can be further simplified by using the following tree pruning algorithm, as described next.

6.4.3 Proposed Tree Pruning Technique

First, we start from the first packet in a frame. Assume the initial system state, which is the waiting time for packet 1, is w_2. Then given all feasible choices $u(1) \in \mathcal{U}(w(1))$, costs of all possible branches can be decided. We next move to the second packet. For all feasible choices $u(2) \in \mathcal{U}(w(2))$ for the second packet, costs of all possible branches can also be decided, because the prediction mode of each state for packet 2 is already known. Then the costs for any feasible path starting from packet 1 ending at packet 3 can be calculated, which is the sum of the costs of the given path. Note that we cannot employ a regular tree pruning at this stage [by regular, we mean eliminating all choices of $w(3)$ but the one emanating from each state of $w(2)$], since the costs the next branch emanating from packet 3 depend on which path ends at $w(3)$. However, as discussed above, the future costs depend only on the prediction mode chosen at this state and not on the quantization parameter. Therefore, for all feasible branches $u(2) \in \mathcal{U}(w(2))$ ending at the same state, we can prune out all choices except the one with minimum cost associated with INTRA/SKIP mode and the one with minimum cost associated with INTER mode for each QoS class. For simplicity, only one branch with INTRA/SKIP and one with INTER mode are shown in Fig. 6.4. Therefore, after pruning at this stage, the diagram will look like what is shown in Fig. 6.4. The branches with the same line style belong to one path. For each state of packet 3, there are at most $1 + |\Pi|$ incoming paths, one associated with INTRA/SKIP mode for packet 2, and $|\Pi|$ with INTER mode for each QoS class for packet 2.

We then move forward to step 2, where we perform tree pruning between packet k and packet $k+1$. Assume pruning is already done between packet $k-1$ and packet

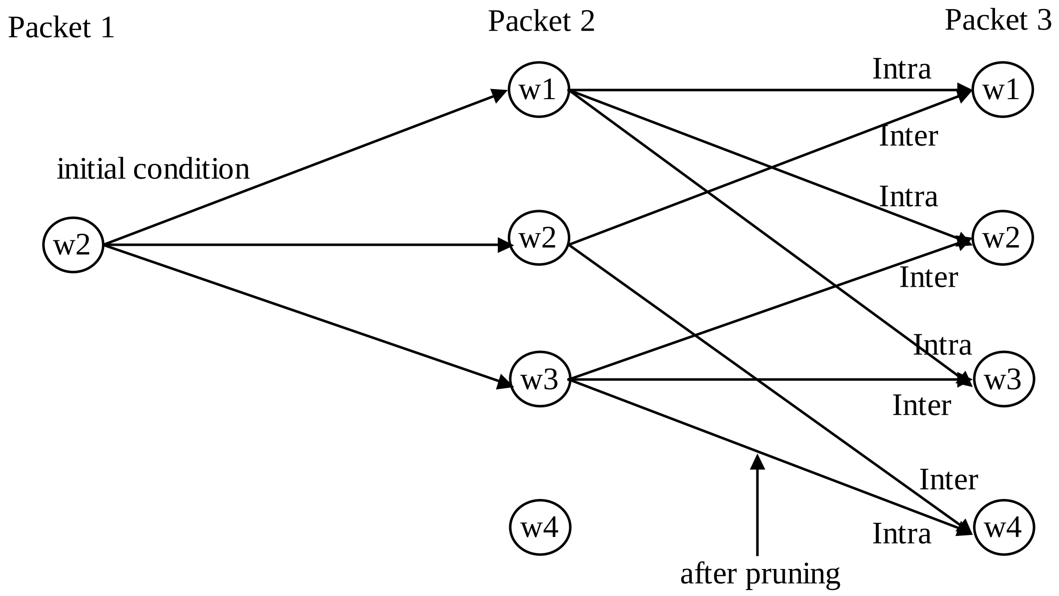


Figure 6.4: Tree pruning, step 1: initial state is given.

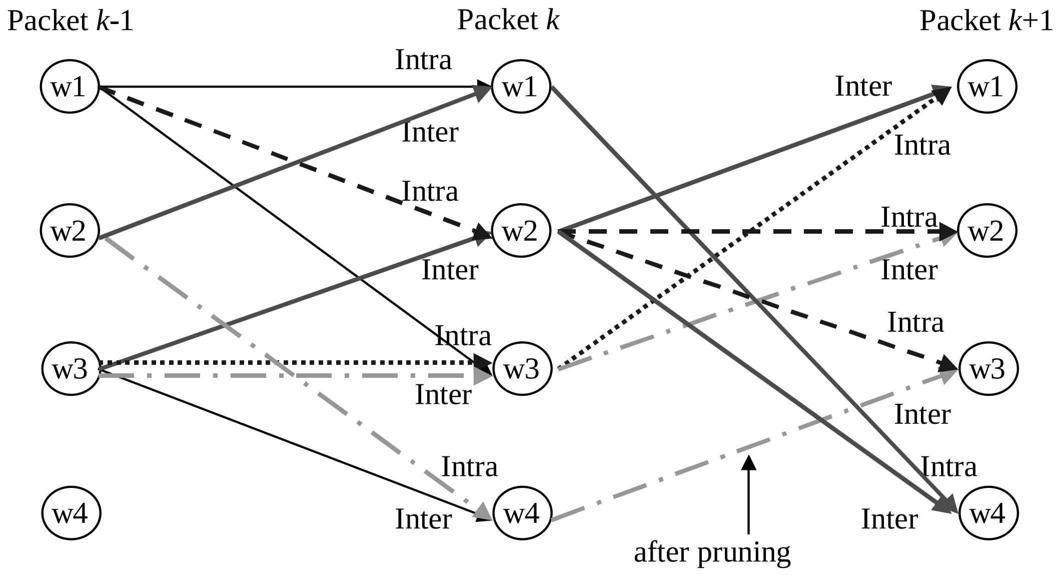


Figure 6.5: Tree pruning, step 2: move forward, prune branches between packet *k* and *k*+1.

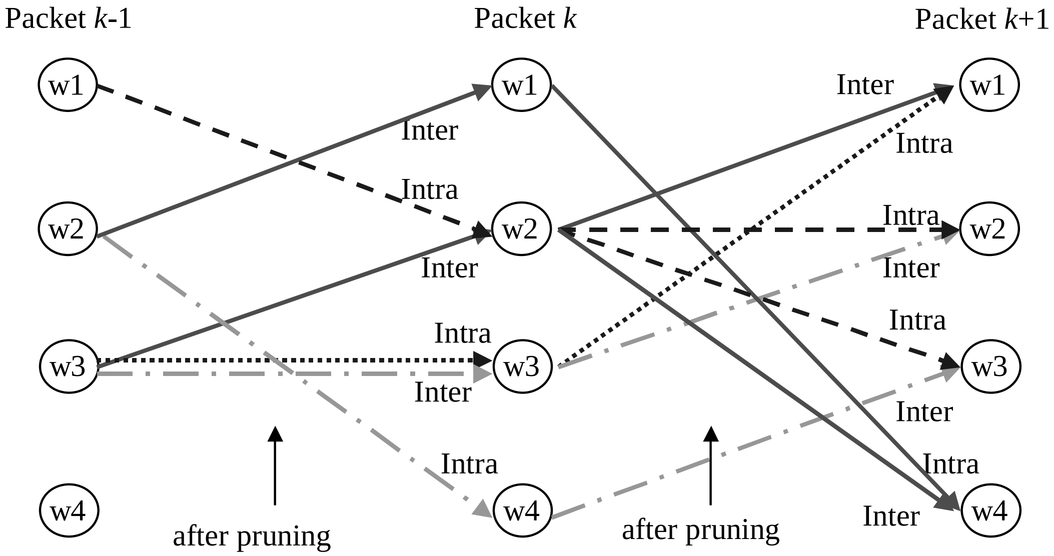


Figure 6.6: Tree pruning, step 3: move backward: prune branches between packet $k-1$ and k.

k, i.e., for each state of packet k, there are at most $1+|\Pi|$ incoming paths. Then for each feasible choice $u(k) \in \mathcal{U}(w(k))$, $|\mathcal{Q} \times \Pi|$ costs need to be calculated. Similarly, because the cost calculation of the future branch depends only on the prediction mode for packet k, we perform tree pruning at this stage. For each state for packet $k+1$, at most $1+|\Pi|$ paths for packet k are left after tree pruning, one associated with INTRA/SKIP mode for packet k, and the other $|\Pi|$ with INTER mode for each class for packet k.

In the last step, we go back to check all the branches between packet $k-1$ and packet k. Some branches left over from previous tree pruning may not be parts of the survived paths. Thus we prune out all these invalid branches between packet $k-1$ and packet k. Figure 6.6 shows the remaining paths after pruning has been completed for packet k and $k+1$.

In going through the above steps, starting from the first packet and moving toward the final packet in a frame, we always keep at most $(1 + |\Pi|) \cdot |\mathcal{W}|$ paths at each stage of the DP algorithm. The optimal solution is thus obtained by forward DP, where the time complexity[3] is further reduced from $O(M \cdot |\mathcal{W}| \cdot |\Pi \times \mathcal{Q}|^2)$ to $O(M \cdot (1 + |\Pi|) \cdot |\mathcal{W}| \cdot |\Pi \times \mathcal{Q}|)$.

6.5 Experimental Results

In this section, we report experimental results that demonstrate the performance of the proposed formulation. The packets can be transmitted at different priorities specified as $\Pi = \{1, 2, 3, 4\}$, whose parameters are defined in Table 6.1. Each class has a different transmission rate and loss probability. We choose these parameters using a model-based TCP-friendly congestion controller as in [8,62]. The network delay is modeled by the shifted Gamma distribution shown in (2.10). The costs for each class are set proportional to the average throughput of the class, which takes into account the transmission rate, probability of packet loss, and network delay distribution. As for the quantizer set $\mathcal{Q}$, we consider quantization steps $\{8, 12, 18, 24\}$ for INTRA mode, $\{4, 6, 8, 10\}$ for INTER mode, and SKIP mode. We set $T_{max} = 333$ milliseconds and $N_W = 300$.

6.5.1 Reference Systems

To illustrate the advantage of jointly selecting the source coding parameters and QoS class, we use a reference system where only one QoS class is available. In

[3] Note the pruning itself has a complexity of $O(M \cdot |\mathcal{W} \times \Pi \times \mathcal{Q}|)$.

class	1	2	3	4
probability of packet loss	0.2	0.1	0.05	0.001
transmission rate (kbps)	210	280	350	420
cost(microcents per kilobits)	25	50	75	100
γ (milliseconds)	40	30	20	10
n	2	2	2	2
α (1/milliseconds)	1/40	1/30	1/20	1/10
mean delay(milliseconds)	120	90	60	30

Table 6.1: Parameters of four service classes.

this reference system, source coding decisions are made to minimize the expected end-to-end distortion subject to the transmission delay constraint, as shown below.

$$\begin{aligned} &\min_{\{\boldsymbol{\mu}\in\mathcal{Q}\}} \sum_{k=1}^{M} E[D_k(\boldsymbol{\mu}, \pi_{ref})] \\ &\text{s.t.} \quad \sum_{k=1}^{M} B_k(\mu_k)/R_k(\pi_{ref}) \le T_0^{(i)}, \end{aligned} \tag{6.16}$$

where π_{ref} is the reference QoS class. The transmission delay constraint for the i-th frame, $T_0(i)$, is recursively obtained as in (6.5). Since our focus here is not on rate control, we set $T_{rc}(i) = 1/F$ seconds for all frames.

6.5.2 Experiments

In this subsection, we compare the proposed DiffServ approach, with multiple QoS classes (6.4), to the reference system (6.16), which uses only one QoS class. We consider four reference systems, each of which uses only one of the four service classes. Each reference system generates a different optimized distortion $D_0(i)$, as well as the corresponding cost $C_0(i)$ and delay $T_0(i)$. The results from reference systems are used as constraints [$C_0(i)$ and $T_0(i)$ for the minimum distortion approach; $D_0(i)$ and $T_0(i)$

for the minimum cost approach] for the corresponding DiffServ system. In other words, we are comparing the four reference systems with four different optimized DiffServ systems, with matching constraints in each case.

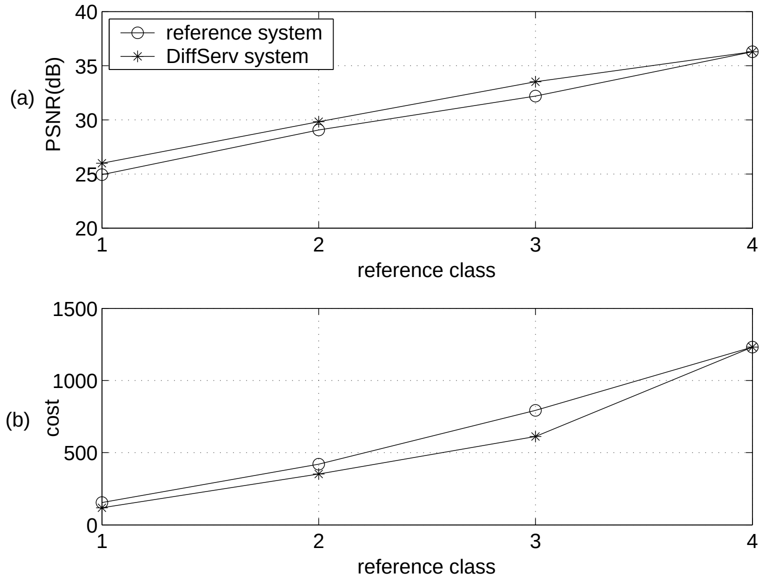


Figure 6.7: Comparison of DiffServ approach with reference system: (a) Minimum distortion approach (b) Minimum cost approach.

Next, we report our experimental results mainly based on the QCIF Foreman sequence with a frame rate 30 fps. The results in using other sequences are briefly reported at the end. First, we compare the proposed minimum distortion approach, with multiple service classes, to the reference systems. We illustrate the performance of the two systems in Fig. 6.7(a), which shows the average decoded PSNR for the Foreman sequence (300 frames) averaged over 50 random channel error realizations (We obtained almost identical results by considering the expected PSNR calculated at the encoder.) The DiffServ approach outperforms the corresponding reference systems by 1.06 dB, 0.76 dB, 1.31 dB, and 0 dB of average PSNR, respectively. We also compare the proposed minimum cost approach, with multiple service classes, to the reference systems. This is shown in Fig. 6.7(b), where the average cost per frame for the Foreman sequence is plotted. In this case, for the same distortion and delay, the minimum cost approach has a cost saving of 23%, 18%, 20%, and 0%, over the corresponding reference system, respectively. In our preliminary work [29], we have observed similar performance gains using different parameters. Of course, different choices of the parameter sets in Table 6.1 may result in different gains.

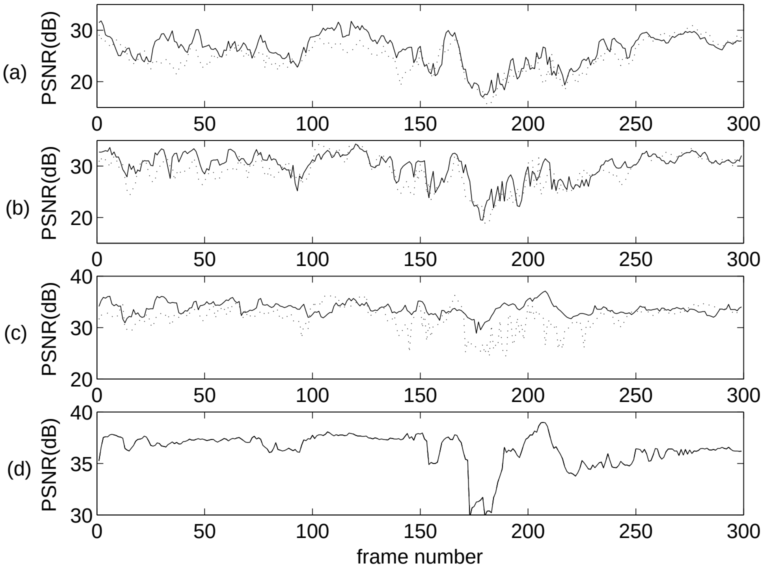


Figure 6.8: One channel realization of minimum distortion approach (solid lines) and reference system (dotted lines), with reference (a) class 1, (b) class 2, (c) class 3, (d) class 4.

Next, we show the temporal behavior of these approaches for one channel error realization. Figure 6.8 shows the PSNR per frame for one channel error realization of the minimum distortion approach and its corresponding reference systems. Figure 6.9 shows the cost per frame for the minimum cost approach and its corresponding reference systems. In Fig. 6.10, the number of packets that the DiffServ approach allocates

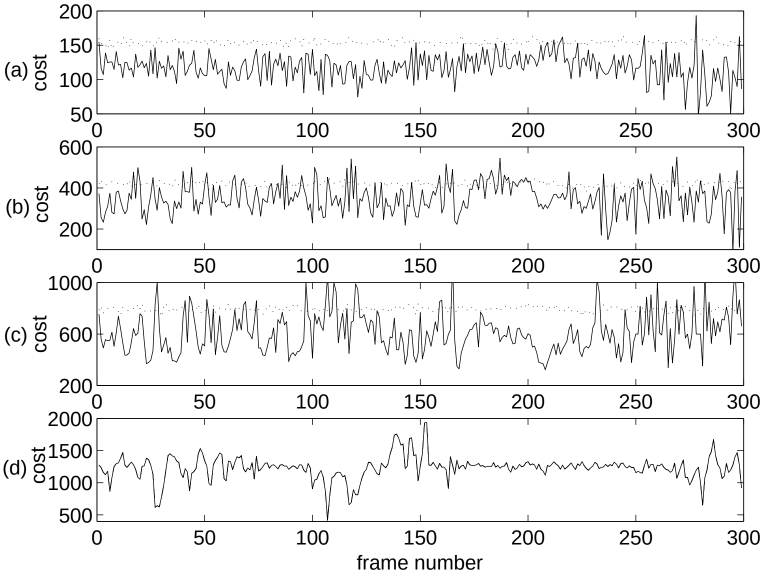


Figure 6.9: One channel realization of minimum cost approach (solid lines) and reference system (dotted lines), with reference (a) class 1, (b) class 2, (c) class 3, (d) class 4.

to each service class is shown for each reference system.

As shown in these figures, the proposed DiffServ system, which jointly adapts the source coding and packet classification, can significantly outperform an approach which uses a fixed service class, except when the fixed class is class 4. When the reference system is class 2 or 3, the performance gain is mainly due to the flexibility in choosing the service class per packet. As shown in Fig. 6.10, the DiffServ approach allocates a significant number of packets to each of the available service classes for systems 2 and 3. If the reference system is class 1, the DiffServ approach still provides significant gains. As shown in Fig. 6.10, nearly all packets are still assigned to class 1, although a few are assigned to the higher service classes. Even this slight flexibility in priority assignment enables the DiffServ system to improve performance. This is

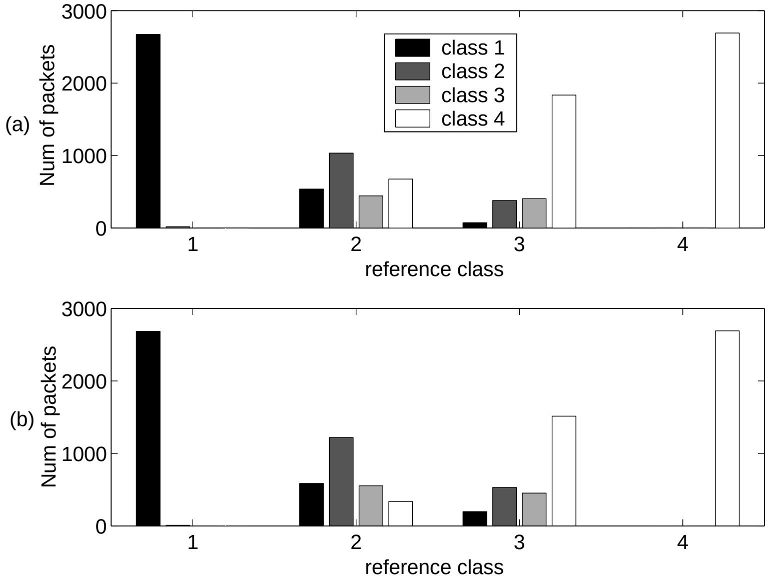


Figure 6.10: Distribution of packet classification in the DiffServ system: (a) Minimum distortion approach (b) Minimum cost approach.

because the packets that are assigned to higher QoS classes have smaller transmission delays. As a result, the encoder buffer occupancy is reduced, and the probability of packet loss due to excessive delay has less effect than if all packets are transmitted at the slowest rate (class 1). Under this situation, the transmission delay constraint is not always active in the optimization problem (6.4).

Unlike reference system 1, 2, and 3, the DiffServ approach does not provide any gain over reference system 4. In this case, the reference system transmits all packets at the highest QoS. Hence, the corresponding constraints are such that there is no advantage in sending a packet at a lower service class; this will only result in longer transmission delay and higher probability of loss. Therefore, as shown in Figs. 6.8-6.10, the DiffServ approach ends up with the same solution as reference system 4. Similar gains are obtained when the Akiyo, Container, Coastguard, and Mother and daughter sequences are tested, as shown in Table 6.2.

Sequence	PSNR gains (dB) minimum distortion approach				Cost savings minimum cost approach			
	class 1	class 2	class 3	class 4	class 1	class 2	class 3	class 4
Foreman	1.06	0.76	1.31	0	23%	18%	20%	0%
Akiyo	0.92	1.91	0.82	0	31%	14%	12%	0%
Container	0.76	0.76	1.22	0	36%	16%	10%	0%
Coastguard	0.62	0.75	0.80	0	21%	19%	24%	0%
Mother and daughter	0.68	0.63	0.75	0	32%	11%	10%	0%

Table 6.2: Average PSNR gains and cost savings of the proposed DiffServ systems compared with the corresponding reference systems (All sequences are in QCIF format, at 30 fps. Akiyo sequence has 100 frames, and each of the others has 300 frames.)

To conclude, the gains illustrated in the above experiments are mainly due to the use of multiple QoS channels in the proposed system. Specifically, for packets that are hard to conceal but easily encoded, the encoder tends to use coarser quantizers and higher QoS. These packets usually correspond to high motion areas that can be predicted from the previous frame. If such a packet is lost, it may be hard to conceal since the associated motion vector is lost. For packets that are hard to encode, the encoder may want to use a finer quantizer to reduce the quantization error and send them with higher QoS as well. The associated cost in this case is relatively high. These packets are usually hard to predict from the previous frame, and are therefore difficult to conceal as well. Packets that are easier to conceal can be sent using a lower QoS class. For these packets the cost is relatively low. In this way, the encoder can select different quantizers and QoS classes in order to optimally balance the received video quality and the overall cost.

6.6 Conclusions

In this chapter, we studied the problem of real-time video transmission over DiffServ networks, where the resource allocation involves the video encoder and network layer. Specifically, the optimal cross-layer resource allocation is achieved by jointly considering error resilient source coding and packet classification to maximize the network resource utilization. The optimization results in UEP for different packets, giving more protection to the most important parts of the bitstream, thus achieving the best video quality. Unlike the previous chapters, we consider the random network delay in this work, where the end-to-end packet delay is adjusted through provisioning the fullness of the encoder buffer to achieve optimal packet scheduling.

Chapter 7

Cross-Layer Resource Allocation for Scalable Video Transmission

In this chapter, we consider cross-layer resource allocation using scalable video. Specifically, in the Internet, we consider the problem of joint source-channel coding based on the H.263+ codec [21]. We also present formulations for scalable video transmission over DiffServ networks and show some preliminary results. In wireless networks, we study energy efficient wireless transmission based on the MPEG-4 FGS (Fine Granularity Scalability) codec [144].

7.1 SNR Scalable Coding

Scalable coding is a coding method that produces an encoded sequence capable of easily accommodating different bit rates. In order to generate a scalable bitstream, two bitstreams, commonly named Base Layer (BL) and Enhancement Layer (EL), are produced. The EL may be composed of multiple layers. Due to the layered representation, scalable coding is also called layered coding. The BL can be decoded independently from the EL, and produces a low quality reconstruction of the video sequence. Based on the BL, the more EL decoded, the higher quality reconstruction can be achieved. Generally speaking, scalability consists of temporal, spatial, SNR scalability and any their combinations. In this work, we focus on SNR scalability.

Scalable coding has inherent error resilience benefits, as different parts of a scalable encoded stream have unequal contributions to the overall quality. When this property is exploited in transmission, e.g., in a DiffServ network or when prioritized FEC is employed, scalable video can maximize the perceived quality. Specifically, the base layer bitstream can be transmitted with higher priority, guaranteeing a basic quality of service, and the enhancement layer bitstreams can be transmitted with lower priority, refining the quality of service. This approach is commonly referred to as layered coding with transport prioritization [89]. Such an approach is especially beneficial in the multicast and broadcast scenarios, where the constraints, such as bit rate, display resolution, network throughput, and decoder complexity, cannot be foreseen at the time of encoding. Rate scalable representations are therefore necessary to allow adaptation to varying network throughput and different requirements of receiver end users without requiring computation at the sender [1].

The wavelet representation provides a multi-resolution/multi-scale expression of a signal with localization in both time and frequency. One of the advantages of wavelet coder in both still image and video compression is that it is free of blocking artifacts. In addition, it offers continuous data rate scalability.

During the last decade, the discrete wavelet transform (DWT) and subband transform have gained increased popularity in image coding due to the breakthrough work of Shapiro [145], Said and Pearlman [146], JPEG2000 [147], and others. Recently, there has also been active research applying the DWT to video coding [148–155]. Among the above studies, 3D wavelet or subband video codecs have gained special attention due to their inherent feature of full scalability [148,149,154,155], whereas the coding efficiency is not satisfactory because of the inefficient temporal filtering. A recent breakthrough comes from the technique of combining lifting techniques with the 3D wavelet or subband coding [156, 157], which brings the standardization of wavelet-based video coding on the corner.

The coding efficiency of 3D wavelet-based video coders still cannot compete with the state-of-the-art DCT based coding. In addition, 3D codecs usually require much more memory and introduce longer delay than 2D technologies. For these reasons, in this work, we consider two popular DCT-based video coding standards that support scalability: the H.263+ standard and the MPEG-4 FGS standard.

7.1.1 H.263+ SNR Scalability

In the H.263+ standard, Annex *O* supports SNR scalability [21]. Basically, there are several types of frames associated with SNR scalability in H.263+, including EI (enhancement I) type and EP (enhancement P) type. If the enhancement layer is upwardly predicted from the lower layer, it is referred to as an EI picture. It is also possible to create a modified bi-directionally predicted picture using both the prior EI picture and the current lower layer picture. This type of picture is termed as an EP picture. For both EI and EP pictures, upward prediction from the reference layer

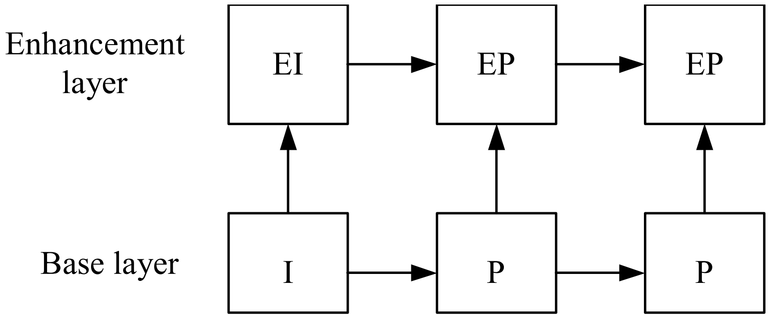


Figure 7.1: H.263+ SNR scalability

picture does not allow motion compensation. Only the forward prediction uses motion compensation. Therefore, to be reconstructed, EP pictures need motion vectors, which can be obtained from motion estimation based on its temporal reference pictures in the same layer (EI or EP). Note that the macroblock syntax of EI frames is the same as that of P frames, and the macroblock syntax of EP frames is the same as that of B frames. As for EI frames, an MB can be encoded by INTRA, UPWARD, or SKIP; while for EP frames, an MB has the encoding modes of INTRA, FORWARD, UPWARD, BIRECTIONAL, or SKIP. The dependency relationship of these types of pictures is shown in Fig. 7.1.

7.1.2 MPEG-4 FGS

Fine Granularity Scalability and Fine Granularity Scalability Temporal (FGST) Scalability have been proposed and adopted by MPEG-4 standard as a desired functionality, especially for streaming video applications [158]. The major difference between FGS and traditional layered technique is its capability to achieve a smooth transition between different bit rates. In MPEG-4 FGS, the BL behaves as a normal baseline MPEG-4 compressed bitstream. In the EL, the difference between the

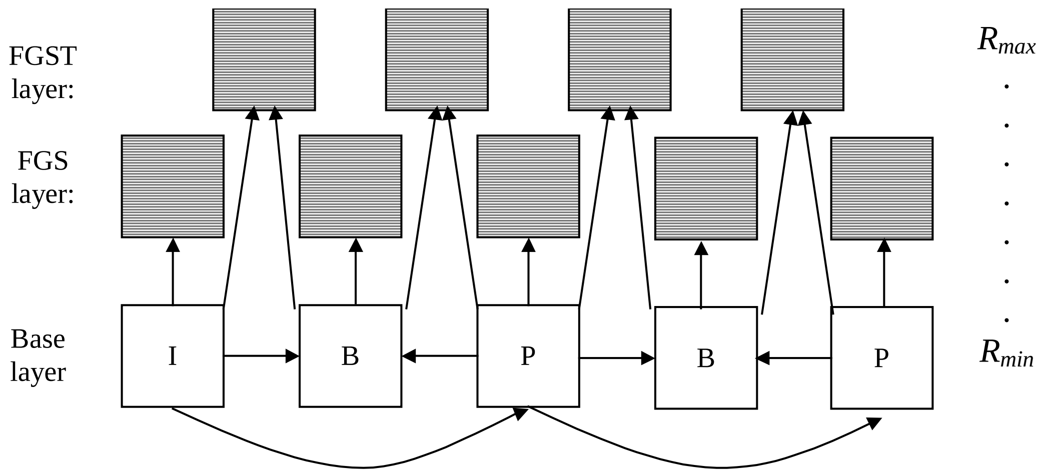


Figure 7.2: MPEG4 FGS and FGST

reconstructed BL and the original frame is first transformed using DCT. The DCT coefficients are bit-plane coded, where each quantized DCT coefficient is considered as a binary number of several bits, and is sent in an order starting from the most significant bit-plane (MSBP) to the least significant bit-plane (LSBP). Note that to reduce drift problem, the EL does not employ motion estimation/compensation. As shown in Fig. 7.2, if the channel capacity is greater than the bit rate of the BL, R_{min}, there is no drift.

Due to its structure, the EL can be truncated into any number of bits within each frame to provide partial enhancement proportional to the number of bits decoded for each frame. In addition, to further support a wide range of bit rate with a scalable bitstream, there is a need to combine FGS with temporal scalability so that not only quantization accuracy can be scalable, but also temporal resolution. This technique is called FGST, as shown in Fig. 7.2. Thus, FGS can provide flexible rate adaptation, complexity scalability, and easy resource adaptation, which make it suitable for real-time video streaming. An good overview of applications enabled by MPEG-4 FGS technology is given in [159].

7.2 Scalable Video Transmission over the Internet

In this section, we study the problem of joint source-channel coding for scalable multicast video transmission over the Internet based on the H.263+ codec. In using the resource-distortion optimization framework, we jointly consider error resilient source coding, prioritized FEC, and error concealment.

Before presenting the formulations, we need to define the distortion for scalable video. The overall distortion for a frame is defined as the sum of the distortions per layer, where the distortion of the first layer is defined in a normal way as in (3.3) and (2.7), and distortion at the higher layer is defined as the differential improvement due to the inclusion of the specific layer in the reconstruction. Therefore, in the absence of packet losses, only the distortion of the first layer is positive and the distortions of all the other layers would be negative, since inclusion of these layers reduces the overall distortion. Of course, it is possible that the distortion introduced by higher layers may become positive in the presence of packet losses due to badly reconstructed enhancement layers [74].

7.2.1 Problem Formulation

For simplicity, in this work, we consider transmitting a two-layer video to multiple users through the Internet. Within this scenario, two problem formulations are presented as follows.

Formulation 1: The first formulation is based on the assumption that the minimum bandwidth of users is known at the sender. The minimum bandwidth could arise from the bandwidth of the last hop of a subscribed user, its low computation capability for decoding, or its low display precision. In this case, the streaming system first needs to adapt the BL to the minimum bandwidth, ensuring that all the subscribed users can obtain a usable quality video. The rest of the bandwidth will then go to the EL. Let subscript "b" and "e" denote BL and EL respectively. We use $\boldsymbol{\mu}^{(b)} = \{\mu_1^{(b)}, \mu_2^{(b)}, ..., \mu_M^{(b)}\}(\boldsymbol{\mu}^{(e)} = \{\mu_1^{(e)}, \mu_2^{(e)}, ..., \mu_M^{(e)}\}$) and $\boldsymbol{\nu}^{(b)} = \{\nu_1^{(b)}, \nu_2^{(b)}, ..., \nu_M^{(b)}\}(\boldsymbol{\nu}^{(e)} = \{\nu_1^{(e)}, \nu_2^{(e)}, ..., \nu_M^{(e)}\}$) to denote the source coding and channel coding parameters for the M source packets in the BL (EL) for a frame, respectively. With transmission delay constraints $T_0^{(b)}$ and T_0 for the BL and whole frame, respectively, the transmission delay constraints can be written as

$$T^{(b)} = \sum_{k=1}^{M} B_{s,k}^{(b)}(\mu_k^{(b)})/R_T + \sum_{k=1}^{N(\boldsymbol{\nu^{(b)}})-M} B_{c,k}^{(b)}(\nu_k^{(b)})/R_T \leq T_0^{(b)}$$

and

$$T^{(e)} = \sum_{k=1}^{M} B_{s,k}^{(e)}(\mu_k^{(e)})/R_T + \sum_{k=1}^{N(\boldsymbol{\nu^{(e)}})-M} B_{c,k}^{(e)}(\nu_k^{(e)})/R_T \leq T_0 - T^{(b)}$$

where $B_{s,k}^{(b)}$ and $B_{s,k}^{(e)}$ are the source bits for the k-th packet in the BL and EL, respectively; $B_{c,k}^{(b)}$ and $B_{c,k}^{(e)}$ are the parity bits for the k-th packet in the BL and EL, respectively; $N(\boldsymbol{\nu^{(b)}})$ and $N(\boldsymbol{\nu^{(e)}})$ are the total number of source and parity packets in the BL and EL, respectively; and R_T is the transmission rate.

In order to clearly represent the formulation, we merge the additional transmission delay of the parity packets into their corresponding source packets, as shown below,

$$T^{(b)} = \sum_{k=1}^{M} B_k^{(b)}(\mu_k^{(b)}, \nu_k^{(b)})/R_T \leq T_0^{(b)}$$

and

$$T^{(e)} = \sum_{k=1}^{M} B_k^{(e)}(\mu_k^{(e)}, \nu_k^{(e)})/R_T \leq T_0 - T^{(b)}$$

where $B_k^{(b)}$ and $B_k^{(e)}$ are the total bits for the k-th packet that include the parity bits associated with this source packet, in the BL and EL, respectively. Thus, we have

$$B_k^{(b)}(\mu_k^{(b)}, \nu_k^{(b)}) = B_{s,k}^{(b)}(\mu_k^{(b)})/R_{c,k}^{(b)}(\nu_k^{(b)})$$

and

$$B_k^{(e)}(\mu_k^{(e)}, \nu_k^{(e)}) = B_{s,k}^{(e)}(\mu_k^{(e)})/R_{c,k}^{(e)}(\nu_k^{(e)})$$

where $R_{c,k}^{(b)}$ and $R_{c,k}^{(e)}$ are the channel coding rate[1] associated with the k-th source packet in the BL and EL, respectively. The problem is then formally formulated as,

$$\begin{aligned} \min_{\{\boldsymbol{\mu}^{(b)}, \boldsymbol{\mu}^{(e)}, \boldsymbol{\nu}^{(b)}, \boldsymbol{\nu}^{(e)}\}} & \sum_{k=1}^{M} E[D_k^{(b)}(\boldsymbol{\mu}^{(b)}, \boldsymbol{\nu}^{(b)})] + \sum_{k=1}^{M} E[D_k^{(e)}(\boldsymbol{\mu}^{(b)}, \boldsymbol{\mu}^{(e)}, \boldsymbol{\nu}^{(b)}, \boldsymbol{\nu}^{(e)})] \\ \text{s.t. } T^{(b)} = & \sum_{k=1}^{M} B_k^{(b)}(\mu_k^{(b)}, \nu_k^{(b)})/R_T \leq T_0^{(b)} \\ T^{(e)} = & \sum_{k=1}^{M} B_k^{(e)}(\mu_k^{(e)}, \nu_k^{(e)})/R_T \leq T_0 - T^{(b)}. \end{aligned} \tag{7.1}$$

By Assuming that the minimum available bandwidth among users is R_{min}, the transmission delay constraint for the BL can be calculated by $T_0^{(b)} = R_{min}T_0/R_T$.

Formulation 2: In the second formulation, we consider the case where the subscribed users have a minimum requirement for their received video quality. Instead of using all the estimated bandwidth for a single layer video, a safer way is to satisfy

[1] The channel coding rate is defined as the number of information bits per channel bit.

the minimum quality requirement by the BL, leaving the rest bandwidth for the EL to improve the perceived video quality as much as possible. This is because a single layer video is usually much more sensitive to the varying channels, where packet loss may cause dramatically decreased quality. However, when layered video is transmitted with prioritized FEC protecting the BL better than the EL, it is very possible that the BL can reach the end user with little packet loss, so that the minimum quality requirement can be satisfied. Formally, the problem is formulated as,

$$\begin{aligned} &\min_{\{\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(b)},\boldsymbol{\nu}^{(e)}\}} \sum_{k=1}^{M} E[D_k^{(b)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\nu}^{(b)})] + \sum_{k=1}^{M} E[D_k^{(e)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(b)},\boldsymbol{\nu}^{(e)})] \\ &\text{s.t.} \quad E[D^{(b)}] = \sum_{k=1}^{M} E[D_k^{(b)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\nu}^{(b)})] \leq D_0^{(b)} \qquad (7.2) \\ &\quad T^{(b)} + T^{(e)} = \sum_{k=1}^{M} B_k^{(b)}(\mu_k^{(b)},\nu_k^{(b)})/R_T + \sum_{k=1}^{M} B_k^{(e)}(\mu_k^{(e)},\nu_k^{(e)})/R_T \leq T_0 \end{aligned}$$

where $D_0^{(b)}$ is the distortion constraint for the BL, which corresponds to the minimum quality requirement imposed by the application.

7.2.2 Implementation Issues

In order to clearly illustrate the concept, a different packetization scheme is used in this work, where each packet has only one MB and every MB is therefore independently decodable. Systematic RS codes are used to perform FEC. Packetization scheme 1 described in Sect. 4.2.1 is employed in simulations, thus the channel coding rate is fixed for all the packets in a layer of a frame (although it can still vary from frame to frame).

For error concealment in the BL, the lost packet is concealed using the motion vector of its preceding packet if available. Otherwise, the zero motion vector is used to perform the concealment. When a packet in the EL is lost, the decoder always uses the "upward" concealment to replace the lost packet by the corresponding MB in the BL of the same frame.

The expected distortion for the BL is calculated the same way as in (2.7). Due to the different coding structures and error concealment strategies used for the EL, we derive the detailed expressions for the expected distortion at the EL based on the ROPE algorithm [84] given by Appendix B.

7.2.3 Sub-Optimal Solution

In this section, we present a solution approach for (7.1) and (7.2) based on Lagrangian relaxation and deterministic DP.

With the use of Lagrange multiplier $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$, (7.1) can be converted into an unconstrained problem shown as below,

$$\begin{aligned}\min_{\{\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(b)},\boldsymbol{\nu}^{(e)}\}} & \sum_{k=1}^{M} E[D_k^{(b)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\nu}^{(b)})] + \sum_{k=1}^{M} E[D_k^{(e)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(b)},\boldsymbol{\nu}^{(e)})] \\ & +\lambda_1 \sum_{k=1}^{M} B_k^{(b)}(\mu_k^{(b)},\nu_k^{(b)})/R_T + \lambda_2 \sum_{k=1}^{M} B_k^{(e)}(\mu_k^{(e)},\nu_k^{(e)})/R_T,\end{aligned} \tag{7.3}$$

which is

$$\begin{aligned}\min_{\{\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(b)},\boldsymbol{\nu}^{(e)}\}} & \sum_{k=1}^{M} \Big\{ E[D_k^{(b)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\nu}^{(b)})] + \lambda_1 B_k^{(b)}(\mu_k^{(b)},\nu_k^{(b)})/R_T \Big\} \\ & + \sum_{k=1}^{M} \Big\{ E[D_k^{(e)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(b)},\boldsymbol{\nu}^{(e)})] + \lambda_2 B_k^{(e)}(\mu_k^{(e)},\nu_k^{(e)})/R_T \Big\}.\end{aligned} \tag{7.4}$$

The same as above, formulation 2 in (7.2) can be converted into an unconstrained problem as shown below:

$$\begin{aligned}\min_{\{\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(b)},\boldsymbol{\nu}^{(e)}\}} &\sum_{k=1}^{M} E[D_k^{(b)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\nu}^{(b)})] + \sum_{k=1}^{M} E[D_k^{(e)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(b)},\boldsymbol{\nu}^{(e)})] \\ &+\lambda_1 \sum_{k=1}^{M} E[D_k^{(b)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\nu}^{(b)})]+ \\ &\lambda_2\{\sum_{k=1}^{M} B_k^{(b)}(\mu_k^{(b)},\nu_k^{(b)})/R_T + \sum_{k=1}^{M} B_k^{(e)}(\mu_k^{(e)},\nu_k^{(e)})/R_T\},\end{aligned} \tag{7.5}$$

which is

$$\begin{aligned}\min_{\{\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(b)},\boldsymbol{\nu}^{(e)}\}} &\sum_{k=1}^{M} \Big\{(1+\lambda_1)E[D_k^{(b)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\nu}^{(b)})] + \lambda_2 B_k^{(b)}(\mu_k^{(b)},\nu_k^{(b)})/R_T\Big\} \\ &+\sum_{k=1}^{M} \Big\{E[D_k^{(e)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(b)},\boldsymbol{\nu}^{(e)})] + \lambda_2 B_k^{(e)}(\mu_k^{(e)},\nu_k^{(e)})/R_T\Big\}.\end{aligned} \tag{7.6}$$

Note that the Lagrangians above are not separable, due to the dependency within each layer and between the layers. There are two explanations for such dependency: 1) in the BL, the distortion for the k-th packet, $E[D_k^{(b)}]$, depends on the encoding modes and FEC parameter chosen for the previous packets; 2) in the EL, the distortion $E[D_k^{(e)}]$ depends on the encoding parameter for the corresponding packet in the BL. In addition, there are two Lagrange multipliers involved, which makes the problem even more complicated. Due to those reasons, we propose a sub-optimal solution to tackle the problem of (7.1). Instead of directly solving (7.4), we break up the dependency and sequentially minimize the two terms of summation, as shown below,

$$\begin{aligned}\text{BL:}\quad &\min_{\{\boldsymbol{\mu}^{(b)},\boldsymbol{\nu}^{(b)}\}} \sum_{k=1}^{M} \Big\{E[D_k^{(b)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\nu}^{(b)})] + \lambda_1 B_k^{(b)}(\mu_k^{(b)},\nu_k^{(b)})/R_T\Big\} \\ \text{EL:}\quad &\min_{\{\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(e)}\}} \sum_{k=1}^{M} \Big\{E[D_k^{(e)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(b)},\boldsymbol{\nu}^{(e)})] + \lambda_2 B_k^{(e)}(\mu_k^{(e)},\nu_k^{(e)})/R_T)\Big\}.\end{aligned} \tag{7.7}$$

The good news is that in each of the optimization problems above, there is only one Lagrange multiplier involved. In addition, because the encoding of BL itself does not depend on that of EL, and a better BL usually leads to a better quality of the overall frame, (besides, a better BL leads to a better reconstruction of the next frame, which is a property not fully captured by this formulation), it is reasonable to believe that the sub-optimal solution would be very close to the optimal one, but with greatly reduced computational complexity.

Next we discuss the solution of (7.7). As the expressions for the BL and EL are in the same form, they are solved in the same way. Thus we only discuss one, and the subscript "b" and "e" are ignored. In addition, for simplicity, the channel codes for one layer are fixed for all the packets in this layer. Then the problem we need to solve is shown as below,

$$\min_{\{\boldsymbol{\mu},\boldsymbol{\nu}\}} \sum_{k=1}^{M} J_k(\boldsymbol{\mu},\boldsymbol{\nu}) = \min_{\{\boldsymbol{\mu},\boldsymbol{\nu}\}} \sum_{k=1}^{M} \{E[D_k(\boldsymbol{\mu},\boldsymbol{\nu})] + \lambda B_k(\mu_k,\nu_k)/R_T\} \tag{7.8}$$

The above unconstrained minimization problem is equivalent to

$$\min_{\{\boldsymbol{\nu}\}} \sum_{k=1}^{M} J_k(\boldsymbol{\mu}^*|\boldsymbol{\nu}) = \min_{\{\boldsymbol{\nu}\}} \left\{ \min_{\{\boldsymbol{\mu}\}} \sum_{k=1}^{M} J_k(\boldsymbol{\mu},\boldsymbol{\nu}) \right\}, \tag{7.9}$$

which can be solved in two steps: optimal mode selection given the source bit rate constraint, and optimal bit allocation between the source coding and channel coding given the total transmission bit rate constraint.

After an error concealment strategy is chosen, the mode selection for each MB only depends on the encoding of its previous MB. Therefore, dependency is constrained within one row. With K_r and N denoting the number of MBs in a row and the number of rows in a frame, respectively, (7.9) can be decoupled into

$$\min_{\{\boldsymbol{\nu}\}}\left\{\min_{\{\boldsymbol{\mu}\}}\sum_{k=1}^{M}J_k(\boldsymbol{\mu},\boldsymbol{\nu})\right\}=\min_{\{\boldsymbol{\nu}\}}\left\{\sum_{r=1}^{N}\min_{\{\boldsymbol{\mu}\}}\sum_{k=1}^{K_r}J_k(\boldsymbol{\mu},\boldsymbol{\nu})\right\}, \tag{7.10}$$

where minimization is independently performed within each row. This relaxed problem can be efficiently solved using DP. The same as formulation 1, a sub-optimal solution of formulation 2 is presented below:

$$\begin{aligned}&\text{BL:}\quad \min_{\{\boldsymbol{\mu}^{(b)},\boldsymbol{\nu}^{(b)}\}}\sum_{k=1}^{M}\left\{(1+\lambda_1)E[D_k^{(b)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\nu}^{(b)})]+\lambda_2B_k^{(b)}(\mu_k^{(b)},\nu_k^{(b)})/R_T\right\}\\&\text{EL:}\quad \min_{\{\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(e)}\}}\sum_{k=1}^{M}\left\{E[D_k^{(e)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\nu}^{(b)},\boldsymbol{\nu}^{(e)})]+\lambda_2B_k^{(e)}(\mu_k^{(e)},\nu_k^{(e)})/R_T\right\}.\end{aligned} \tag{7.11}$$

which can be solved in the same way presented for the formulation 1, since it has the same form as (7.7).

7.2.4 Experimental Results

The simulation is based on the H.263+ SNR scalable codec [21]. The test sequence is QCIF Foreman with frame rate 30 fps. The channel transmission rate is 360 kbps (not to be confused with the channel capacity – the theoretical maximum transmission rate at which information passes error free over the channel; channel transmission rate is obtained from the estimated channel capacity), with 180 kbps for the BL and 180 kbps for the EL.

This experiment is to calculate the R-D bound of the proposed scheme, which is obtained based on the assumption that the encoder has accurate estimation of channel capacity. The BL and EL bitstreams are transmitted to the same network, thus they present the same probability of packet loss before error recovery. To illustrate the

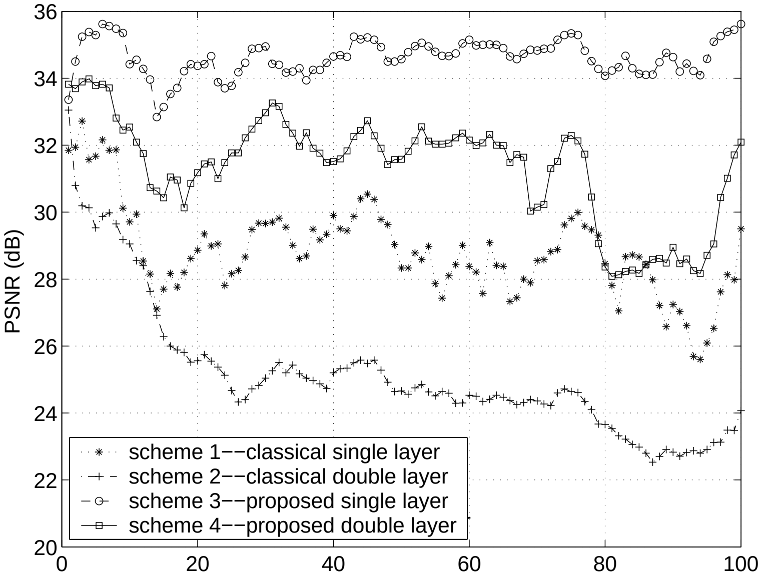


Figure 7.3: One realization of the four schemes (transmission rate: 360 kbps; channel capacity: 306kbps).

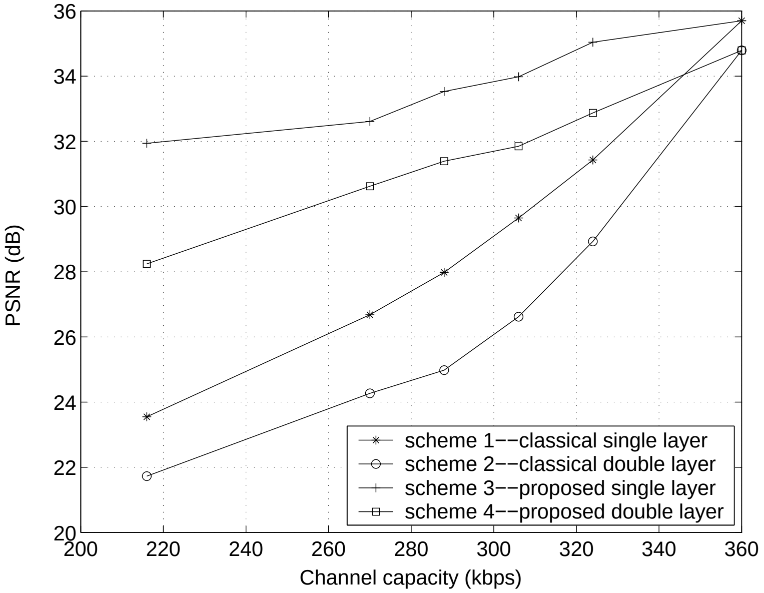


Figure 7.4: R-D bounds of the four schemes (transmission rate is 360 kbps, with 180 kbps for the BL and the EL respectively for a double layer video).

Channel capacity (kbps)	216	270	288	306	324	360
Protection ratio at BL	0.50	0.33	0.28	0.21	0.15	0
Protection ratio at EL	0.04	0.07	0.05	0.04	0.02	0

Table 7.1: Protection ratio for scheme 4 (transmission rate: 360 kbps, with 180 kbps for the BL and the EL respectively).

effectiveness of the proposed scheme, we consider the classical approaches as the reference systems, where source coding is performed without taking into account channel errors and thus channel coding is used. Four schemes are compared: i) classical optimized non-scalable scheme (without taking into account the channel error and without using channel coding); ii) classical optimized double-layer scheme; iii) proposed optimized scheme applied on single-layer video; and iv) proposed optimized scheme applied on double-layer video.

Figure 7.3 shows a realization of the above four schemes in terms of PSNR versus frames, where the transmission rate is 360 kbps, and the channel capacity 306 kbps. Figure 7.4 depicts the R-D bounds of the four schemes. It can be seen from Fig. 7.4 that scheme 3 and 4 outperform schemes 1 and 2 by 0-8.5 dB and 0-6.5 dB, respectively. Scheme 3 has higher PSNR than that of scheme 4. This makes sense because when the encoder can be tailored accurately to the channel, non-scalable methods can achieve better performance than scalable ones due to the redundancy of layered approaches at the source coding and the overhead of packet headers. However, this does not mean that non-scalable methods are superior to scalable ones, because when the encoder cannot adapt to the channel accurately, the scalable method is more robust to a wide range of channel variations, which we will show later in this chapter. Another observation from this experiment is that, although we did not explicitly use

UEP for the BL and the EL, the optimization automatically results in UEP. As shown in Table 7.1, the protection ratio for the BL is always higher than that for the EL. It is achieved by using error concealment at the decoder, which renders the EL more robust to the packet loss than the BL. This is because if a packet in the EL is lost, it can be concealed from the BL of the same frame, while if a packet in the BL is lost, it can only get concealment from the previous frame. In addition, as the channel gets worse, the encoder will allocate more resources to protect the bitstream.

7.3 Scalable Video Transmission over DiffServ Networks

As discussed before, scalable video has inherent error resilience benefits when the layered source bitstream can be applied with prioritized transmission. FEC is one method supporting prioritized transmission at the application layer. DiffServ is another method implemented in the network layer to support prioritized transmission from the transport network. FEC depends on the accurate estimation of the channel state, because inaccurate estimation will result in inappropriate protection levels, which usually leads to poor performance, e.g., over-protection or under-protection. However, in DiffServ, the SLA assures the user of the channel information, such as bandwidth, probability of packet loss, delay, etc. Therefore, in this case, the benefits of layered coding can be realized through marking packets from different layers with different QoS level.

Parallel to formulation 1 and 2 above, we have the formulations for scalable video transmission over a DiffServ network as below:

$$\begin{aligned}
\min_{\{\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\pi}^{(b)},\boldsymbol{\pi}^{(e)}\}} & \sum_{k=1}^{M} E[D_k^{(b)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\pi}^{(b)})] + \sum_{k=1}^{M} E[D_k^{(e)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\mu}^{(e)},\boldsymbol{\pi}^{(b)},\boldsymbol{\pi}^{(e)})] \\
\text{s.t.} \quad & \sum_{k=1}^{M} B_k^{(b)}(\mu_k^{(b)})/R_T(\pi_k^{(b)})) \leq T_0^{(b)} \quad \text{or} \quad \sum_{k=1}^{M} E[D_k^{(b)}(\boldsymbol{\mu}^{(b)},\boldsymbol{\pi}^{(b)})] \leq D_0^{(b)} \\
C = & \sum_{k=1}^{M} B_k^{(b)}(\mu_k^{(b)})c(\pi_k^{(b)}) + \sum_{k=1}^{M} B_k^{(e)}(\mu_k^{(e)})c(\pi_k^{(e)}) \leq C_0 \\
T = & \sum_{k=1}^{M} B_k^{(b)}(\mu_k^{(b)})/R_T(\pi_k^{b)}) + \sum_{k=1}^{M} B_k^{(e)}(\mu_k^{(e)})/R_T(\pi_k^{(e)}) \leq T_0,
\end{aligned} \tag{7.12}$$

where $\boldsymbol{\pi}^{(b)} = \{\pi_1^{(b)}, \pi_2^{(b)}, ..., \pi_M^{(b)}\}$ and $\boldsymbol{\pi}^{(e)} = \{\pi_1^{(e)}, \pi_2^{(e)}, ..., \pi_M^{(e)}\}$ are the vectors of service class parameters for the M packets in a frame for the BL and EL respectively, C_0 is the cost constraint, and T_0 is the transmission delay constraint.

For simplicity, in this study, we assume that only two service classes are available: a premium one and a regular one. We next present the simulation results based on formulation 1.

7.3.1 Experimental Results

This experiment is to compare the performance of the double-layer video deliver to the single-layer video delivery in a channel with wide range channel capacity variations, based on the assumption that the estimated channel capacity may be different from the true capacity. This usually happens due to the delay in the feedback for estimation and time-varying channel states. For simplicity, we assume that an ideal DiffServ network is employed to perform UEP for BL and EL. Under this assumption, assuming the channel capacity is C, the packet loss probabilities can be

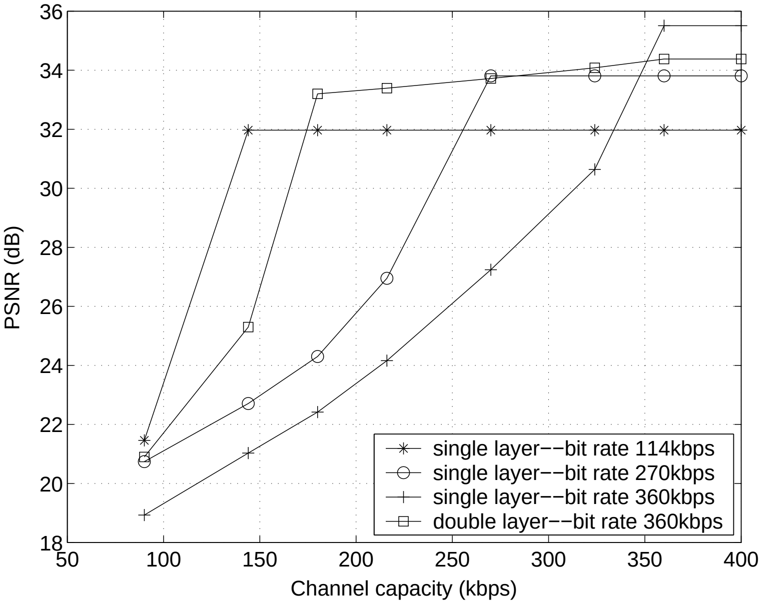


Figure 7.5: Scalable vs. non-scalable video (double layer video is tuned to the estimated rate of 270 kbps)

calculated as $\rho^{(b)} = \max\{0, 1 - C/R^{(b)}\}$, $\rho^{(e)} = \max\{0, \min\{1, 1 - (C - R^{(b)})/R^{(e)}\}\}$. That is, if C is greater than $R^{(b)}$, then the loss rate of the BL, $\rho^{(b)}$, is zero, and the packet loss only occurs in the EL. No channel coding is used in this experiment.

In Fig. 7.5, we plot the simulation results in terms of average PSNR vs. channel capacity. For double layer video, we calculate the overall PSNR. The source coding is optimized to the estimated channel capacity, which is 270 kbps in the experiment, using the proposed framework (7.12). Single layer videos are encoded at different rates, as shown in the figure. Each curve corresponds to one realization of one encoded bitstream at different channel capacities. The sharp dropping appears when the channel capacity is lower than the generated source bit rate, which corresponds to over estimation. It is clear that double layer video usually degrades more gracefully than single layer video with a wide range of channel capacities, due to its flexibility in allowing bit rate allocation to BL and EL and in performing UEP.

7.4 Scalable Video Transmission over Wireless Networks

In this section, we study the cross-layer resource allocation problem for energy efficient wireless video communications based MPEG-4 FGS video. This work investigates the transmission of FGS video over wireless links, using power management to provide UEP for the video bitstream.

With regard to related work, UEP between the BL and the EL using different techniques, such as ARQ and FEC, has been studied in [160]. The application of UEP within the EL of FGS bitstream is first considered in [161], where the frame-grained loss protection (FGLP) framework is introduced. Based on that framework, a "degressive" protection algorithm (DEP) is presented in [96] for the optimal protection of the EL using FEC. In [162], an R-D optimized UEP problem is studied for Progressive FGS (PFGS) over wireless channels using prioritized FEC for the BL and the EL. A similar problem is studied in [46] to minimize the processing power for PFGS video with given bandwidth and distortion constraints.

In this work, we consider the transmission of an MPEG-4 FGS video sequence over a single wireless channel for a single user. For simplicity, we assume that the BL is already encoded and can be transmitted without errors. By optimally allocating transmission power to the different packets in the EL, our objective is to minimize the end-to-end distortion with given transmission power and bit rate constraints.

7.4.1 Problem Formulation

Assume that the BL is always correctly received and its resultant distortion is $D^{(b)}$. In this case, the total distortion is $D^{(b)} - E[\Delta^{(e)}]$, where $E[\Delta^{(e)}]$ is the expected distortion improvement introduced by jointly decoding the BL and all the correctly received EL packets. If we assume that each packet can be successfully decoded only if this packet and all the previous EL packets are correctly received, we can write $E[\Delta^{(e)}]$ as:

$$E[\Delta^{(e)}] = \sum_{l=1}^{L} \prod_{i=1}^{l} (1 - \rho_i^{(e)}) \Delta_l, \tag{7.13}$$

where L is the number of packets in the EL, $\rho_i^{(e)}$ is the loss probability for the i-th packet, and Δ_l is the distortion improvement introduced by successfully decoding packet l and all its previous EL packets.

With the given maximum amount of energy E_0 for the frame, our goal is to determine how to allocate the available power in a way to minimize the overall distortion, shown as follows,

$$\begin{aligned} &\min_{\{P_l^{(e)}\}} \left\{ D^{(b)} - E[\Delta^{(e)}] \right\} \\ &\text{s.t.} \quad E_{tot}^{(e)} = E_0, \end{aligned} \tag{7.14}$$

where E_{tot} is the total energy used for transmitting the EL. It can be written as

$$E_{tot}^{(e)} = \sum_{l=1}^{L} \frac{B_l^{(e)} P_l^{(e)}}{R_T} \tag{7.15}$$

where $B_l^{(e)}$ is the number of bits in the l-th packet, $P_l^{(e)}$ is the power used for transmitting it, and R_T is the channel transmission rate.

7.4.2 Solution Algorithm

By introducing a Lagrange multiplier $\lambda \geq 0$, the solution of the constrained minimization problem can be found by solving the following unconstrained minimization problem:

$$\min_{\{P_l^{(e)}\}} J = \min_{\{P_l^{(e)}\}} \left\{ D^{(b)} - \sum_{l=1}^{L} \prod_{i=1}^{l} (1-\rho_i^{(e)})\Delta_l + \lambda \sum_{l=1}^{L} \frac{B_l^{(e)} P_l^{(e)}}{R_T} \right\}. \tag{7.16}$$

Note that to reach the optimal solution, the first derivative of the cost function J with respect to $P_l^{(e)}$ must equal zero. After some simple manipulations, for $j < L$, the following relationship must hold:

$$\left(\frac{\partial \rho_j^{(e)}}{\partial P_j^{(e)}}\right)^{-1} (1-\rho_j^{(e)}) = \left(\frac{\partial \rho_L^{(e)}}{\partial P_L^{(e)}}\right)^{-1} (1-\rho_L^{(e)}) \frac{B_L^{(e)}}{B_j^{(e)}} \left[1 + \sum_{l=j+1}^{L} \prod_{h=l}^{L} (1-\rho_h^{(e)})^{-1} \frac{\Delta_{l-1}}{\Delta_L}\right]$$
$$\text{for} \quad j = 1, ..., L-1. \tag{7.17}$$

The left side in the above expression represents the information related to the j-th packet, which depends only on the power of the following packets, $(j+1)$ to L. Readers are referred to [163] for the details of the derivations.

The average transmission power used by a modulation scheme directly affects the probability of packet loss. We assume that the relationship between the probability of loss for the l-th packet $\rho_l^{(e)}$ and the transmission power $P_l^{(e)}$ is known at the transmitter. This relationship can be defined using an analytical model of the wireless channel or can be determined from empirical measurements. Based on the channel model described in [73]:

$$\rho_j^{(e)} = 1 - e^{-k/P_j^{(e)}}, \tag{7.18}$$

we can derive the following expression from (7.17):

$$\left(P_j^{(e)}\right)^2 = \left(P_L^{(e)}\right)^2 \frac{B_L^{(e)}}{B_j^{(e)}} \left[1 + \sum_{l=j+1}^{L} \prod_{i=l}^{L} e^{k/P_i^{(e)}} \frac{\Delta_{l-1}}{\Delta_L}\right]. \tag{7.19}$$

Thus $P_j^{(e)}$ $(j = 1, ..., L-1)$ can be recursively calculated once $P_L^{(e)}$ is known, as shown above. In other words, for a given value of $P_L^{(e)}$, we can recursively calculate $P_j^{(e)}$ in terms of $P_{j+1}^{(e)}, ..., P_L^{(e)}$, and therefore obtain the resultant total energy based on (7.15). The minimization problem (7.14) can then be solved by finding the value of $P_L^{(e)}$ that satisfies the energy constraint. A closed form solution is difficult to compute analytically. Alternatively, a numerical method such as the bisection method can be used.

Extensive experimental results using different types of packetization at different channel transmission rates were presented in [163]. Simulations using the proposed optimal power allocation algorithm outperformed the schemes based on equal energy scheme and various empirical models by 0.24 dB to 1.48 dB based on the QCIF Foreman test sequence.

7.5 Conclusions

In this chapter, we extended our work in the previous chapters to scalable video, i.e., we studied cross-layer resource allocation for scalable video transmission in various network infrastructures. Specifically we studied i) JSCC for Internet video transmission based on the H.263+ scalable codec; ii) JSCC for DiffServ video transmission based on the H.263+ scalable codec; and iii) optimal power allocation for energy efficient wireless video transmission based on the MPEG-4 FGS codec. Simulation results showed that our schemes are able to adaptively allocate the available network resources based on the transmission parameters and the layered source characteristics, and consequently achieve better video quality.

Chapter 8

Conclusions

This dissertation is concerned with error control for real-time video transmission using the approach of cross-layer resource allocation. Specifically, we have focused on the end system design. By assuming that the encoder can access the network resource allocation parameters in the underling layers, our proposed resource-distortion framework can be used to optimally allocate resources across layers in order to achieve the best video quality.

In Chapter 3, we proposed a general resource-distortion optimization framework to study the problem of joint source and network coding. This framework jointly considers the available error control components in different network infrastructures such as error resilient source coding, channel coding, power adaptation, packet classification, and error concealment to achieve the best video quality. The framework focuses on the end system design for video transmission systems, and performs cross-layer resource allocation to those layers that can be controlled and specified by the end system. Based on this framework, network resources can be optimally and dynamically allocated in a way so that the end system is adaptive to the changing channel conditions. Thus this framework is dependent on accurate CSI estimation and requires increased protocol layer interaction. For actual applications of the framework, three special cases have been extensively studied.

In Chapter 4, we studied cross-layer resource allocation for real-time video transmission over the Internet. In this scenario, the available error control components are error resilient source coding, channel coding and error concealment. Among those error control components, we focused on application layer channel coding. We first studied application layer packetizations in providing FEC. Then we introduced retransmission into the framework, and studied the performance of different channel coding scenarios such as pure FEC, pure retransmission, and hybrid FEC/retransmission. We have shown that FEC and retransmission each can be optimal depending on different network round-trip-time, channel loss rates, and channel transmission rates. The application of hybrid FEC/retransmission produced better results than the applications of only one of the two error control methods.

In Chapter 5, we studied cross-layer resource allocation for real-time video transmission over energy efficient wireless networks. In this case, based on the assumption that the physical layer is accessible and the transmitter power levels can be specified by the encoder, the available error control components are error resilient source coding, transport-layer FEC, link-layer FEC, power adaptation, and error concealment. The study was carried out in a joint source-channel and power adaptation (JSCCPA) framework, which is a special case of the resource-distortion optimization framework. Our focus was on the channel coding and power adaptation. We first showed the superb performance of the proposed product FEC which can provide optimal UEP for video streams. Next, through simulations on a hybrid wireless network, we showed that transport-layer FEC is not necessary if the wired link has no error, based on our simulation setups. In addition, we showed the advantage of jointly adapting the link-layer FEC and transmission power to the varying wireless channel conditions. The study showed that although both channel coding and power adaptation can be used to achieve prioritized protection, each has its effective working region. This observation can help assess the effectiveness of different adaptation components in practical system design. Furthermore, the proposed algorithm, which is based on Lagrangian relaxation, can be used to tackle other discrete optimization problems with two constraints.

In Chapter 6, we studied joint source coding and packet classification for real-time video transmission over DiffServ networks, which support QoS and thus provide transport prioritization. The network delay is modeled by a random process and the end-to-end packet delay is managed through provisioning based on the fullness of the encoder buffer. By jointly adapting the source coding and packet classification, the optimization results in UEP for different packets, giving more protection to the most important parts of the bitstream, thus achieving the maximum video quality at the receiver end. In addition, the solution algorithm presented in this chapter, which is based on Lagrangian relaxation and system state discretion, can also be applied to tackle a discrete optimization problem with two constraints, as an alternative to the algorithm presented in Chapter 5.

In Chapter 7, we extended the formulations presented in Chapter 3 to scalable video. We first studied the problem of joint source-channel coding for transmitting scalable video over the Internet based on an H.263+ SNR scalable codec. By jointly optimizing the source coding parameters and the FEC, the optimization results in UEP for the BL and the EL, achieving prioritized FEC for layered coding. In addition, we studied the problem of optimal power allocation to different video packets based on MPEG-4 FGS codec for energy efficient wireless video transmission.

A number of future directions naturally grow out of the work reported in this dissertation. Specifically, we are interested in the following studies.

First, a series of other important considerations should be included into the general framework. For example, rate control, an important part of multimedia communication system design, has not yet been incorporated into our framework. Although rate control is usually designed separately from source coding and error control, it may improve the overall performance if it is designed jointly with the other parts of the system. In addition, to the best of our knowledge, rate control in the context of DiffServ networks is still an open problem.

Second, under the current standard practice, video packets corrupted by bit error are rejected at the application end in the framework during wireless transmission. Recent studies, however, have shown that passing corrupted video packets to the application end may achieve significant gain [45]. Thus, further research should be conducted in this direction.

Third, the work on hybrid FEC/retransmission showed that delay-constrained retransmission can greatly outperform FEC under certain network conditions. This suggests that link-layer ARQ mechanism may need to be adapted based on each application's latency and reliability requirements, as well as the traffic load, indicating that link-layer ARQ should also be brought into the optimization framework [111].

Fourth, the channel models used in the simulations assumed uncorrelated channel errors. For example, packet loss in the Internet is modeled as a Bernoulli random process. Besides, we assumed i.i.d. fading for the wireless simulations. Those models do not reflect the bursty nature of errors caused by network congestion and correlated fading. Two-state or higher order Markov chains, however, are more realistic alternatives. Such a direction is currently under consideration in our recent work [164], where the effect of correlated channel fading on the optimization framework is studied.

Fifth, the work in this dissertation primarily focuses on optimal resource allocation for a single user in a unicast scenario. However, the framework can be extended to multiuser and multicast video transmission systems. In that scenario, studies of scalable video transmission become more important. Our work in Chapter 7 highlighted the major considerations of optimal cross-layer resource allocation problems for scalable video transmission over the Internet, wireless networks, and DiffServ networks. But the detailed studies remain to be done. In addition, issues such as fairness as well as distributed versus centralized resource allocation must be addressed in the multicast setting.

All the above-mentioned directions fall into the area of cross-layer design. As a long-term research direction, we plan to expand the proposed framework from end-system design to include network design. We also plan to apply the framework to wireless ad hoc networks, where, due to the mobility of wireless nodes and limited energy of each node, cross-layer design will be a more critical issue.

References

[1] B. Girod, J. Chakareski, M. Kalman, Y. J. Liang, E. Setton, and R. Zhang, "Advances in network-adaptive video streaming," in *Proc. Tyrrhenian International Workshop on Digital Communications*, Capri, Italy, Sept. 2002, pp. 1–8.

[2] P. Chaudhury, W. Mohr, and S. Onoe, "The 3GPP proposal for IMT-2000," *IEEE Commun. Mag.*, pp. 72–81, Dec. 1999.

[3] 3GPP2 specification, *Requirement for a 3G network based on Internet protocol (All-IP) with support for TIA/EIA-41 interoperability*, Oct. 2000.

[4] *Part 11: Wireless LAN Medium Accesscontrol (MAC) and Physical Layer (PHY) Specifications*, Aug. 1999, IEEE 802.11-1999 Standard.

[5] *Part 11: Wireless LAN Medium Accesscontrol (MAC) and Physical Layer (PHY) Specifications, High-Speed Physical Layer in the 5-GHz Band*, Sept. 1999, IEEE 802.11a, Supplement to IEEE 802.11 Standard.

[6] B. Girod and N. Färber, *Wireless video, in compressed video over network, edited by A. Reibman and M. T. Sun*, Marcel Dekker, 2000.

[7] D. Wu, Y. Hou, and Y.-Q. Zhang, "Scalable video coding and transport over broad-band wireless networks," *Proc. IEEE*, vol. 89, pp. 6–20, Jan. 2001.

[8] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Resource allocation for multimedia streaming over the Internet," *IEEE Trans on Multimedia*, vol. 3, no. 3, pp. 339–355, Sept. 2001.

[9] A. Sehgal and P. A. Chou, "Cost-distortion optimized streaming media over Diffserv networks," in *Proc. of IEEE Int. Conf. Multimedia and Expo*, Lausanne, Switzerland, Aug. 2002.

[10] H. Zheng, "Optimizing wireless multimedia transmissions through cross layer design," in *Proc. IEEE*, Baltimore, MD, July 2003, vol. 1, pp. 185–188.

[11] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, pp. 23–50, Nov. 1998.

[12] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J. 27*, pp. 379–423 and 623–656, July 1948.

[13] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Transporting real-time video over the Internet: Challenges and approaches," *Proc. IEEE*, vol. 88, pp. 1855–1877, Dec. 2000.

[14] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers," RFC 2474, IETF, Dec. 1998, http://www.rfc-editor.org/rfc/rfc2474.txt.

[15] B. E. Carpenter and K. Nichols, "Differentiated Services in the Internet," *Proc. IEEE*, vol. 90, no. 9, pp. 1479–1494, Sept. 2002.

[16] Q. Chen and K. P. Subbalakshmi, "Joint source-channel decoding for MPEG-4 video transmission over wireless channels," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 1780–1789, Dec. 2003.

[17] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra, "Error control for receiver-driven layered multicast of audio and video," *IEEE Trans. Multimedia*, pp. 108–122, March 2001.

[18] P. A. Chou and A. Sehgal, "Rate-distortion optimized receiver-driven streaming over best-effort networks," in *Proc. IEEE International PacketVideo Workshop*, Pittsburgh, PA, April 2002.

[19] ITU-T, *Video codec for audiovisual services at $p \times 64$ kbits*, ITU-T Recommendation H.261, Mar. 1993, Version 2.

[20] ITU-T, *Video coding for low bitrate communication*, ITU-T Recommendation H.263, Nov. 1995, Version 1.

[21] ITU-T, *Video coding for low bitrate communication*, ITU-T Recommendation H.263, Jan. 1998, Version 2.

[22] ITU, *H.26L Test Model Term Number 8*, ITU-T Video coding Expert Group, July 2001, online, ftp://standard.pictel.com/video-site/h26L/.

[23] ISO/IEC, *Generic coding of moving pictures and associated audio*, ISO/IEC, 1994, Draft.

[24] ISO/IEC, *Generic coding of moving pictures and associated audio*, ISO/IEC JTC1/SC29/WG11, MPEG93/457, 1995, Draft.

[25] ISO/IEC, *Generic coding of audio-visual objects: Part 2-visual*, ISO/IEC JTC1/SC29/WG11, N2502, FDIS of ISO/IEC 14496-2, Nov. 1998.

[26] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. on Circ. and Syst. for Video Techn.*, vol. 13, pp. 560–576, July 2003, Special issue on the H.264/AVC video coding standard.

[27] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. on Circ. and Syst. for Video Techn.*, vol. 13, pp. 688–703, July 2003, Special issue on the H.264/AVC video coding standard.

[28] C. Y. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video transmission over burst-error wireless channels," *IEEE J. Select. Areas Commun.*, vol. 17, no. 5, pp. 756–773, May 1999.

[29] C. E. Luna, Y. Eisenberg, R. Berry, T. N. Pappas, and A. K. Katsaggelos, "Joint source coding and packet marking for video transmission over Diffserv networks," in *Proceedings of Tyrrhenian International Workshop on Digital Communications*, Capri, Italy, Sept. 2002.

[30] ISO/IEC, *MPEG Video Test Model 5*, ISO/IEC JTC1/SC29/WG11, MPEG93/457, Apr. 1993, Draft.

[31] J.Choi and D.Park, "A stable feedback control of the buffer state using the controlled Lagrange multiplier method," *IEEE Trans. Circuit Syst. Video Technol*, vol. 3, pp. 546–558, Sept. 1994.

[32] A. R. Reibman and B. G. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Trans. Circuit Syst. Video Technol*, vol. 2, pp. 361–372, Dec. 1992.

[33] B. G. Haskell and A. R. Reibman, "Multiplexing of variable rate encoded streams," *IEEE Trans. Circ. and Syst. for Video Techn.*, vol. 4, pp. 417–424, Aug. 1994.

[34] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Trans. Circuit Syst. Video Technol*, vol. 7, pp. 246–250, Feb. 1997.

[35] ISO/IEC, *Coding of Moving Pictures and Audio*, MPEG-4 Video Verification Model V8.0, ISO/IEC JTC1/SC29/WG11 N3093, Dec. 1999.

[36] B. Tao, B. W. Dickinson, and H. A. Peterson, "Adaptive model-driven bit allocation for MPEG video coding," *IEEE Trans. Circuit Syst. Video Technol*, vol. 10, pp. 147–157, Feb. 2000.

[37] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Trans. Image Processing*, vol. 3, pp. 533–545, Sept. 1994.

[38] T. Wiegand, M. Lightstone, D. Mukherjee, T. Campbell, and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, pp. 182–190, April 1996.

[39] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations," *IEEE Trans. Image Processing*, vol. 3, pp. 26–40, Jan. 1994.

[40] T. N. Pappas and R. J. Safranek, "Perceptual criteria for image quality evaluation," in *Handbook of Image and Video Processing*, A. C. Bovik, Ed. Academic Press, 2000.

[41] J. Chen and T. N. Pappas, "Perceptual coders and perceptual metrics," in *Human Vision and Electronic Imaging VI*, B. E. Rogowitz and T. N. Pappas, Eds., San Jose, CA, Jan. 2001, vol. 4299 of *Proc. SPIE*, pp. 150–162.

[42] Y. Eisenberg, F. Zhai, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Quality metrics for measuring end-to-end distortion in packet-switched video communication systems," in *Proc. SPIE*, San Jose, CA, Jan. 2004, vol. 5292.

[43] J. Postel, "RFC791: Internet protocol," RFC 791, Internet Engineering Task Force, Sept. 1981, http://www.rfc-editor.org/rfc/rfc791.txt.

[44] Douglas E. Comer, *Internetworking with TCP/IP*, vol. 1, Prentice-Hall, Upper Saddle River, NJ, 1995.

[45] L. Larzon, M. Degermark, and S. Pink, "UDP Lite for real time multimedia applications," in *Proc. of the QoS miniconference of IEEE Int. Conf. Communications, (ICC'99)*, Vancouver, Canada, June 1999.

[46] Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Network-adaptive scalable video streaming over 3G wireless network," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Thessaloniki, Greece, Oct. 2001.

[47] H. Schulszrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," RFC 3550, Internet Engineering Task Force, July 2003, http://www.networksorcery.com/enp/rfc/rfc3550.txt.

[48] H. Schulzrinne, A. Rao, and R. Lanphier, "RFC 2326: Real time streaming protocol RTSP," RFC 2326, IETF, April 1998, http://www.rfc-editor.org/rfc/rfc2326.txt.

[49] D. D. Clark and D. L. Tennenhouse, "Architecture considerations for a new generation of protocols," *Computer Communications Review*, vol. 20, no. 4, pp. 200–208, Sept. 1990.

[50] M. Handley and C. Perkins, "Guidelines for writers of RTP payload format specifications," RFC 2736, IETF, Dec. 1999.

[51] S. Wenger, "H.264/AVC over IP," *IEEE Trans. on Circ. and Syst. for Video Techn.*, vol. 13, pp. 645–656, July 2003, Special issue on the H.264/AVC video coding standard.

[52] D. Hoffman, G. Fernando, and V. Goyal, "RTP payload format for MPEG1/MPEG2 video," RFC 2038, Internet Engineering Task Force, Oct. 1996, http://www.faqs.org/rfcs/rfc2038.html.

[53] Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, and H. Kimata, "RTP payload format for MPEG-4 audio/visual streams," RFC 3016, Internet Engineering Task Force, Nov. 2000.

[54] T. Turletti and C. Huitema, "RTP payload format for H.261 video streams," RFC 2032, Internet Engineering Task Force, Oct. 1996, http://www.faqs.org/rfcs/rfc2032.html.

[55] M. Gallant and F. Kossentini, "Rate-distortion optimized layered coding with unequal error protection for robust Internet video," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 357–372, March 2001.

[56] Y. Wang, G. Wen, S. Wenger, and A. K. Katsaggelos, "Review of error resilience techniques for video communications," *IEEE Signal Processing Magazine*, vol. 17, pp. 61–82, July 2000.

[57] D. Wu and R. Negi, "Effective capacity: A wireless link model for support of qualify of service," *IEEE Trans. Wireless Communications*, vol. 2, pp. 630–643, July 2003.

[58] D. Wu, Y. T. Hou, W. Zhu, H.-J. Lee, T. Chiang, Y.-Q. Zhang, and H. J. Chao, "On end-to-end architecture for transporting MPEG-4 video over the Internet," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 923–941, Sept. 2000.

[59] S. McCanne, V. Jacobson, and M. Vitterli, "Receiver-driven layered multicast," in *Proc. ACM SIGCOMM'96*, Aug. 1996, pp. 117–130.

[60] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer," in *Proc. IEEE INFOCOM'98*, Mar. 1998, vol. 3, pp. 996–1003.

[61] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicast applications,” Tech report, International Computer Science Institutes, Berkley, CA, March 2000.

[62] P. A. Chou and Z. Miao, “Rate-distortion optimized streaming of packetized media,” *IEEE Trans. on Multimedia*, 2001, Submitted.

[63] R. Puri, K.-W. Lee, K Ramchandran, and V. Bharghavan, “An integrated source transcoding and congestion control paradigm for video streaming in the Internet,” *IEEE Trans. Multimedia*, vol. 3, pp. 18–32, March 2001.

[64] Y.-G. Kim, J.-W. Kim, and C.-C. Jay Kuo, “Network aware error control using smooth and fast rate adaptation mechanism for TCP-friendly Internet video transmission,” *IEEE CAS-VT Special Issue on Streaming Video*, 2000, Submitted.

[65] K. N. Ngan, C. W. Yap, and K. T. Tan, *Video coding for wireless communication systems*, Marcel Dekker, Inc., 2001.

[66] H. Lin and S. K. Das, “Performance study of TCP/RLP/MAC in next generation CDMA systems,” in *14th IEEE Int. Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Beijing, China, Sept. 2003.

[67] D. A. Eckhardt, *An Internet-style approach to managing wireless link errors*, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, May 2002.

[68] A. Majumdar, D. G. Sachs, I. V. Kozintsev, K Ramchandran, and M. M. Yeung, “Multicast and unicast real-time video streaming over wireless LANs,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, pp. 524–534, June 2002.

[69] M. Yajnik, S. Moon, J. Jurose, and et al., “Measurement and modeling of the temporal dependence in packet loss,” Tech. Rep. 98-78, UMASS CMPSCI, 1998.

[70] V. Paxson and S. Floyd, “Wide area traffic: the failure of Poisson modeling,” *IEEE Trans. Networking*, vol. 3, pp. 226–244, June 1995.

[71] G. Hooghiemstra and P. Van Mieghem, "Delay distributions on fixed Internet paths," Tech. Rep. report 20011020, Delft University of Technology, 2001.

[72] Y. Eisenberg, C. E. Luna, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Joint source coding and transmission power management for energy efficient wireless video communications," *IEEE Trans. on Circuits System Video Technology*, vol. 12, no. 6, pp. 411–424, June 2002.

[73] L. Ozarow, S. Shamai, and A. D. Wyner, "Information theoretic considerations for cellular mobile radio," *IEEE Trans. Vehicular Technology*, pp. 359–378, May 1994.

[74] L. P. Kondi, F. Ishtiaq, and A. K. Katsaggelos, "Joint source-channel coding for motion-compensated dct-based snr scalable video," *IEEE Trans on Image Processing*, vol. 11, pp. 1043–1052, Sept. 2002.

[75] F. Zhai, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Joint source-channel coding and power allocation for energy efficient wireless video communications," in *Proc. 41st Allerton Conf. Communications, Control, and Computing*, Oct. 2003.

[76] T. S. Rappaport, *Wireless communications principle and practice*, Prentice Hall, 1998.

[77] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, pp. 389–400, Apr. 1988.

[78] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, Aug. 2000.

[79] G. Cheung, W.-T. Tan, and T. Yoshimura, "Rate-distortion optimized application-level retransmission using streaming agent for video streaming over 3G wireless network," in *Proc. IEEE Inf. Conf. Image Processing*, Rochester, New York, Sept. 2002.

[80] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communications: a review," *Proc. IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

[81] G. Côté, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 952–965, June 2000.

[82] B. Girod and N. Färber, "Feedback-based error control for mobile video transmission," *Proceedings of the IEEE*, vol. 87, pp. 1707–1723, Oct. 1999.

[83] R. O. Hinds, T. N. Pappas, and J. S. Lim, "Joint block-based video source-channel coding for packet-switched networks," *Proc. SPIE*, vol. 3309, pp. 124–133, Jan. 1998.

[84] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 966–976, June 2000.

[85] D. Wu, Y. T. Hou, B. Li, W. Zhu, Y.-Q. Zhang, and H. J. Chao, "An end-to-end approach for optimal mode selection in Internet video communication: theory and application," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 977–995, June 2000.

[86] F. Zhai, R. Berry, T. N. Pappas, and A. K. Katsaggelos, "Rate-distortion optimized error control scheme for scalable video streaming over the Internet," in *Proc. IEEE Int. Conf. on Multimedia and Expro*, Bartimore, MD, July 2003.

[87] Y. Wang, M. T. Orchard, V. Vaishampayan, and A. R. Reibman, "Multiple description coding using pairwise correlating transforms," *IEEE Trans. Image Processing*, vol. 10, pp. 351–366, Mar. 2001.

[88] D. G. Sachs, R. Anand, and K. Ramchandran, "Wireless image transmission using multiple-description based concatenated codes," in *Proc. SPIE Image and Video Communications and Processing*, San Jose, CA, Jan. 2000, vol. 3974, pp. 300–311.

[89] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1737–1744, Nov. 1996.

[90] R. O. Hinds, *Robust model selection for block-motion compensated video encoding*, Ph.D. Thesis, MIT, Cambridge, MA, June 1999.

[91] F. Hartanto and H. R. Sirisena, "Hybrid error control mechanism for video transmission in the wireless IP networks," in *Proc. of IEEE Tenth Workshop on Local and Metropolitan Area Networks (LANMAN'99)*, Sydney, Australia, Nov. 1999, pp. 126–132.

[92] J. Rosenberg and H. Schulzrinne, "An RTP payload format for generic forward error correction," RFC 2733, IETF, Request for Comments (Proposed Standard), Dec. 1999.

[93] N. Celandroni and F. Pototì, "Maximizing single connection TCP goodput by trading bandwidth for BER," *Int. J. of Commun. Syst.*, vol. 16, pp. 63–79, Feb. 2003.

[94] T. Stockhammer and C. Buchner, "Progressive texture video streaming for lossy packet networks," in *Proc. International Packet Video Workshop*, Kyongju, Korea, April 2001.

[95] R. Zhang, S. L. Regunathan, and K. Rose, "End-to-end distortion estimation for RD-based robust delivery of pre-compressed video," in *Proc. 35th Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Oct. 2001.

[96] X. Yang, C. Zhu, Z. Li, G. Feng, S. Wu, and N. Ling, "Unequal error protection for motion compensated video streaming over the Internet," in *Proc. IEEE Int. Conf. Image Processing*, Rochester, New York, Sept. 2002.

[97] *Multiplexing protocol for low bitrate multimedia communication over highly error-prone channels*, ITU-T Draft of H.223 Annex C, Sept. 1997.

[98] B. J. Dempsey, J. Liebeherr, and A. C. Weaver, "On retransmission-based error control for continuous media traffic in packet-switched networks," *Computer Networks and ISDN Syst.*, vol. 28, pp. 719–736, Mar. 1996.

[99] G. J. Wang, Q. Zhang, W. W. Zhu, and Y.-Q. Zhang, "Channel-adaptive error control for scalable video over wireless channel," in *IEEE MoMuc 2000*, Oct. 2000.

[100] C. E. Luna, *Video Quality and Network Efficiency Trade-Offs in Video Streaming Applications*, Ph.D. Thesis, Northwestern University, Evanston, IL, June 2002.

[101] C.E. Luna, Y. Eisenberg, T. Pappas, R. Berry, and A. K. Katsaggelos, "Joint source coding and data rate adaptation for energy efficient wireless video streaming," in *Proc. IEEE International PacketVideo Workshop*, Pittsburgh, PA, April 2002.

[102] B. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture," RFC 1633, Internet Engineering Task Force, June 1994, http://www.ifla.org/documents/rfcs/rfc1633.txt.

[103] L. Zhang, S. E. Deering, S. Shenker, and D. Zappala, "RSVP: A new resource ReSerVation protocol," *IEEE Network*, vol. 7, pp. 8–18, 1993.

[104] S. Blake and et al., "An architecture for differentiated services," RFC 2475, IETF, Dec. 1998, http://www.rfc-editor.org/rfc/rfc2475.txt.

[105] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On multiple description streaming with content delivery networks," in *Proc. IEEE INFOCOM*, June 2002.

[106] A. Katsaggelos, F. Ishtiaq, L. P. Kondi M.-C. Hong, M. Banham, and J. Brailean, "Error resilience and concealment in video coding," in *Proc. Eur. Signal Processing Conf.(EUSIPCO)*, Rhodes, Greece, Sept. 1998, pp. 221–228.

[107] H. R. Rabiee, H. Radha, and R. L. Kashyap, "Error concealment of still image and video stream with multi-directional recursive nonlinear filters," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Atlanta, GA, May 1996, pp. 37–40.

[108] ITU-T, *Video codec test model near-term*, H.263 Test-Model Ad Hoc Group, Oct. 1999, Version 11 (TMN11), Release 2.

[109] F. Zhai, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Rate-distortion optimized hybrid error control for real-time packetized video transmission," in *Proc. IEEE Int. Conf. on Communications (ICC'04)*, Paris, French, June 2004.

[110] L. Hanzo, "Bandwidth-efficient wireless multimedia communications," *Proc. IEEE*, vol. 86, no. 7, pp. 1342–1998, July 1998.

[111] M. var der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 1752–1763, Dec. 2003.

[112] N. Farvardin and V. Vaishampayan, "Optimal quantizer design for noisy channels: an approach to combined source-channel coding," *IEEE Trans. Inform. Theory*, vol. IT-38, pp. 827–838, 1987.

[113] G. Davis and J. Danskin, "Joint source and channel coding for Internet image transmission," in *Proc. SPIE Conf. Wavelet Applications of Digital Image Processing XIX*, Denver, CO, Aug. 1996.

[114] B. Hong and A. Nosratinia, "Rate-constrained scalable video transmission over the internet," in *Proc. IEEE International PacketVideo Workshop*, Pittsburgh, PA, April 2002.

[115] Z. He, J. Cai, and C. W. Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," *IEEE Trans. Circ. and Syst. for Video Techn.*, vol. 12, pp. 511–523, June 2002.

[116] J. Kim, R. M. Mersereau, and Y. Altunbasak, "Error-resilient image and video transmission over the Internet using unequal error protection," *IEEE Trans. Image Processing*, vol. 12, pp. 121–131, Feb. 2003.

[117] S. Appadwedula, D. L. Jones, K. Ramchandran, and L. Qian, "Joint source channel matching for wireless image transmission," in *IEEE Int. Conf. Image Processing*, Chicago, IL, Oct. 1998.

[118] F. Zhai, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Packetization schemes for forward error correction in Internet video streaming," in *Proc. 41st Allerton Conf. on Communication, Control and Computing*, Otc. 2003.

[119] T. Wiegand, N. Färber, and B. Girod, "Error-resilient video transmission using long-term memory motion-compensated prediction," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 1050–1062, June 2000.

[120] S. Wenger, G. D. Knorr, J. Ott, and F. Kossentini, "Error resilience support in H.263+," *IEEE Trans. on Circ. and Syst. for Video Techn.*, vol. 8, no. 7, pp. 867–877, Nov. 1998.

[121] K. H. Yang, D. W. Kang, and A. F. Faryar, "Efficient intra refreshment and synchronization algorithms for robust transmission of video over wireless networks," in *Proc. IEEE Int. Conf. Image Processing*, Thesaloniki, Greece, 2001, pp. 938–941.

[122] H. Yang, R. Zhang, and K. Rose, "Drift management and adaptive bit rate allocation in scalable video coding," in *Proc. IEEE Int. Conf. Image Processing*, Rochester, New York, Sept. 2002.

[123] G. Cheung and A. Zakhor, "Bit allocation for joint source/channel coding of scalable video," *IEEE Trans. Image Processing*, vol. 9, pp. 340–356, Mar. 2000.

[124] F. Zhai, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Rate-distortion optimized hybrid error control for real-time packetized video transmission," *IEEE Trans. Image Processing*, Jan. 2004, Submitted.

[125] A. El Gamal, C. Nair, B. Prabhakar, E. Uysal-Biyikoglu, and S. Zahedi, "Energy-efficient scheduling of packet transmissions over wireless networks," in *Proc. of IEEE INFOCOM'02*, 2002.

[126] B. E. Collins and R. L. Cruz, "Transmission policies for time varying channels with average delay constraints," in *Proceedings of Allerton Conf. on Comm. Control and Computing*, Monticello, IL, Sept. 1999.

[127] C. E. Luna, Y. Eisenberg, R. Berry, T. N. Pappas, and A. K. Katsaggelos, "Joint source coding and data rate adaption for energy efficient wireless video streaming," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 1710–1720, Dec. 2003.

[128] J. Shin, J. Kim, and C.-C. Kuo, "Quality-of-service mapping mechanism for packet video in differentiated services network," *IEEE Trans. on Multimedia*, vol. 3, no. 2, June 2001.

[129] D. Quaglia and J. C. De Martin, "Delivery of MPEG video streams with constant perceptual quality of service," in *Proc. IEEE Int. Conf. on Multimedia and Expro*, Lausanne, Switzerland, Aug. 2002, vol. 2, pp. 85–88.

[130] D. Quaglia and J. C. De Martin, "Adaptive packet classification for constant perceptual quality of service delivery of video streams over time-varying networks," in *Proc. IEEE Int. Conf. on Multimedia and Expro*, Baltimore, MD, July 2003, vol. 3, pp. 369–372.

[131] F. Zhai, C. E. Luna, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Joint source coding and packet classification for real-time video transmission over differentiated services networks," *IEEE Trans. Multimedia*, 2004, To appear.

[132] Q. Zhang, W. Zhu, Z. Ji, and Y.-Q. Zhang, "A power-optimized joint source channel coding for scalable video streaming over wireless channel," in *Proc. IEEE ISCAS*, Aug. 2001, pp. 137–140.

[133] Y. Pei ad J. W. Modestino, "Multi-layered video transmission over wireless channels using an adaptive modulation and coding scheme," in *Proc. IEEE Int. Conf. Image Processing*, Thesaloniki, Greece, Oct. 2001.

[134] Y. S. Chan and J. W. Modestino, "A joint source coding-power control approach for video transmission over CDMA networks," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 1516–1525, Dec. 2003.

[135] S. Zhao, Z. Xiong, and X. Wang, "Joint error control and power allocation for video transmission over CDMA networks with multiuser detection," *IEEE Trans. Circ. and Syst. for Video Techn.*, vol. 12, pp. 425–437, June 2002.

[136] G. M. Schuster and A. K. Katsaggelos, *Rate-Distortion Based Video Compression: Optimal Video Frame Compression and Object Boundary Encoding*, Kluwer Academic Publishers, 1997.

[137] S. Falahati, A. Svensson, N. C. Ericsson, and A. Ahlén, "Hybrid type-II ARQ/AMS and scheduling using channel prediction for downlink packet transmission on fading channels," in *Nordic Radio Symposium*, 2001.

[138] P. G. Sherwood and K. Zeger, "Error protection for progressive image transmission over memoryless and fading channels," *IEEE Trans. Comm.*, vol. 46, pp. 1555–1559, Dec. 1998.

[139] V. Stanković, R. Hamzaoui, and Z. Xiong, "Product code error protection of packetized multimedia bitstreams," in *Proc. IEEE Int. Conf. Image Processing*, Barcelona, Spain, Sept. 2003.

[140] N. Bambos, "Toward power-sensitive network architectures in wireless communications: Concepts, issues, and design aspects," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 966–976, June 2000.

[141] R. Fletcher, *Practical methods of optimization*, New York: Wiley, 2nd edition, 1987.

[142] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[143] F. Zhai, C. E. Luna, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "A novel cost-distortion optimization framework for video streaming over differentiated services networks," in *Proc. IEEE Int. Conf. Image Processing*, Barcelona, Spain, Sept. 2003.

[144] *Coding of audio-visual objects, Part 2-visual: Amendament 4: streaming video profile*, ISO/IEC 14496-2/FPDAM4, July 2000.

[145] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3463, Dec. 1993.

[146] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circ. and Syst. for Video Techn.*, vol. 6, pp. 243–250, June 1996.

[147] *JPEG-2000 VM3.1 A Software*, ISO/IECJTC1/SC29/WG1 N1142, Jan. 1999.

[148] B.-J. Kim and W. A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees," in *Proc. of Data Compression Conference*, 1997, pp. 251–260.

[149] J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang, "Three-dimensional embedded subband coding with optimized truncation (3D ESCOT)," *Applied and Computational Harmonic Analysis 10*, pp. 290–315, 2001.

[150] K. Shen and E. J. Delp, “Wavelet based rate scalable video compression,” *IEEE Trans. Circ. and Syst. for Video Techn.*, vol. 9, pp. 109–122, Feb. 1999.

[151] Y. Yang and S. S. Hemami, “Generalized rate-distortion optimization for motion-compensated video coders,” *IEEE Trans. Circ. and Syst. for Video Techn.*, vol. 10, pp. 942–955, Sept. 2000.

[152] X. Yang and K. Ramchandran, “Scalable wavelet video coding using aliasing-reduced hierarchical motion compensation,” *IEEE Trans. Image Processing*, vol. 9, pp. 778–791, May 2000.

[153] Y.-Q. Zhang and S. Zafar, “Motion-compensated wavelet transform coding for color video compression,” *IEEE Trans. on Circ. and Syst. for Video Techn.*, vol. 2, pp. 285–296, Sept. 1992.

[154] J. R. Ohm, “Three-dimensional subband coding with motion compensation,” *IEEE Trans. Image Proc.*, vol. 3, pp. 559–571, Sept. 1994.

[155] S. Choi and J. W. Woods, “Motion-compensated 3-D subband coding of video,” *IEEE Trans. Image Proc.*, vol. 8, pp. 155–167, Feb. 1999.

[156] A. Secker and D. Taubman, “Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting,” in *Proc. IEEE International Conference on Image Processing (ICIP)*, Thessaloniki, Greece, Oct. 2001.

[157] J.-R. Ohm, “Motion-compensated wavelet lifting filters with flexible adaptation,” in *Proceedings of Tyrrhenian International Workshop on Digital Communications*, Capri, Italy, Sept. 2002.

[158] W. Li, “Overview of fine granularity scalability in MPEG-4 video standard,” *IEEE Trans on Circuits and Systems for Video Technology*, vol. 11, pp. 301–307, March 2001.

[159] M. va der Schaar, L. G. Boland, and Q. Li, “Novel applications of fine-granular-scalability:Internet and wireless video, scalable storage, personalized TV, universal media coding,” in *Proc. of SCI2001/ISAS2001*, Orlando, FL, 2001.

[160] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, "Robust Internet video transmission based on scalable coding and unequal error protection," *Image Communication*, vol. 15, pp. 77–94, Sept. 1999.

[161] M. van der Schaar and H. Radha, "Unequal packet loss resilience for Fine-Granular-Scalability video," *IEEE Trans. Multimedia*, vol. 3, pp. 381–394, Dec. 2001.

[162] G. Wang, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "Channel-adaptive unequal error protection for scalable video transmission over wireless channel," in *Proc. of SPIE, Visual Communications and Image Processing*, San Jose, CA, Jan. 2001, pp. 648–655.

[163] C. E. Costa, Y. Eisenberg, F. Zhai, and A. K. Katsaggelos, "Energy efficient wireless transmission of MPEG-4 Fine Granular Scalable video," in *Proc. IEEE Int. Conf. Communications (ICC'04)*, Paris, French, June 2004.

[164] E. Soyak, Y. Eisenberg, F. Zhai, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Channel modeling and its effect on the end-to-end distortion in wireless video communications," in *Proc. IEEE Int. Conference on Image Processing*, 2004, submitted.

[165] H. Wang, F. Zhai, Y. Eisenberg, and A. K. Katsaggelos, "Cost-distortion optimized unequal error protection for object-based video communications over lossy networks," *IEEE Trans. Circ. and Syst. for Video Techn.*, Jan. 2004, submitted.

Appendix A

Lagrangian Relaxation Method

Enclosed in this appendix are some theorems applied by the Lagrangian relaxation method. Those theorems form the basis of the proposed algorithm in Chapter 5.

Theorem 1. *Let S_H be a finite set and $H \in S_H$ be a member of that set. Then, for any $\lambda > 0$, the optimal solution $H^*(\lambda)$ to the following problem*

$$\begin{aligned} &\min_{\{H \in S_H\}} D(H) + \lambda C(H) \\ &\textit{subject to:} \quad T(H) \leq T_0 \end{aligned} \tag{A.1}$$

is also an optimal solution to the problem below

$$\begin{aligned} &\min_{\{H \in S_H\}} D(H) \\ \textit{subject to:} \quad &C(H) \leq C(H^*(\lambda)) \\ &T(H) \leq T_0 \end{aligned} \tag{A.2}$$

Proof. (By contradiction) Assume that the above theorem is false. This implies that there exists an $H \in S_H$ such that $D(H) < D(H^*(\lambda))$, $C(H) < C(H^*(\lambda))$, and

$T(H) < T_0$. Hence $D(H) + \lambda C(H) < D(H^*(\lambda)) + \lambda C(H^*(\lambda))$, subject to $T(H) < T_0$. This is a contradiction since $H^*(\lambda)$ is the optimal solution to (A.1).

□

Theorem 2. *(λ Theorem): In (A.1), if $C(H^*(\lambda_1)) > C(H^*(\lambda_2))$, then*

$$\lambda_2 \geq -\frac{D(H^*(\lambda_1)) - D(H^*(\lambda_2))}{C(H^*(\lambda_1)) - C(H^*(\lambda_2))} \geq \lambda_1 \tag{A.3}$$

Proof. By the optimality of $H^*(\lambda_1)$ and $H^*(\lambda_2)$, respectively, we have

$$D(H^*(\lambda_1)) + \lambda_1 C(H^*(\lambda_1)) \leq D(H^*(\lambda_2)) + \lambda_1 C(H^*(\lambda_2)) \tag{A.4}$$

$$D(H^*(\lambda_2)) + \lambda_2 C(H^*(\lambda_2)) \leq D(H^*(\lambda_1)) + \lambda_2 C(H^*(\lambda_1)) \tag{A.5}$$

From (A.4), we have

$$-\frac{D(H^*(\lambda_1)) - D(H^*(\lambda_2))}{C(H^*(\lambda_1)) - C(H^*(\lambda_2))} \geq \lambda_1 \tag{A.6}$$

From (A.5), we have

$$\lambda_2 \geq -\frac{D(H^*(\lambda_1)) - D(H^*(\lambda_2))}{C(H^*(\lambda_1)) - C(H^*(\lambda_2))} \tag{A.7}$$

□

Theorem 3. *$C(H^*(\lambda))$ of (A.1) is a non-increasing function of the Lagrange multiplier λ.*

Proof. (By contradiction using Theorem 2) or,

For any $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$, by the optimality, we have (A.4) and (A.5). Adding them results in

$$(\lambda_1 - \lambda_2)(C(H^*(\lambda_1) - C(H^*(\lambda_2)) \leq 0 \tag{A.8}$$

□

Appendix B

Distortion Calculation for H.263+ Scalable Video

This appendix gives the detailed algorithm for calculating the expected distortion when using H.263+ scalable codec. Since the base layer is encoded/decoded independently of the enhancement layer, the distortion calculation for the base layer is the same as described in Sect. 3.4.1, based on the ROPE algorithm [84]. Next we derive the expressions for calculating the expected distortion of the enhancement layer only.

The parameters used here are defined in Table 3.1, with added "(b)" and "(e)" representing BL and EL, respectively. The overall expected distortion levels of pixel i at the EL of frame n are given by

$$\begin{aligned} E[d_i^{(n)}(e)] &= E[(f_i^{(n)} - \tilde{f}_i^{(n)}(e))^2 - (f_i^{(n)} - \tilde{f}_i^{(n)}(b))^2] \\ &= E[\tilde{f}_i^{(n)}(e)^2] - E[\tilde{f}_i^{(n)}(b)^2] + 2f_i^{(n)}E[\tilde{f}_i^{(n)}(b)] - 2f_i^{(n)}E[\tilde{f}_i^{(n)}(e)] \end{aligned} \tag{B.1}$$

Note that in order to calculate the expected distortion, the first and second order expected values of each pixel, $E[\tilde{f}_i^{(n)}(b)]$, $E[\tilde{f}_i^{(n)}(e)]$, $E[\tilde{f}_i^{(n)}(b)^2]$, and $E[\tilde{f}_i^{(n)}(e)^2]$, are required. These values have different expressions with different modes. The expressions of $E[\tilde{f}_i^{(n)}(b)]$ and $E[\tilde{f}_i^{(n)}(b)^2]$ with different modes have been defined in (3.4), (3.5), and (3.6), respectively. The expressions of $E[\tilde{f}_i^{(n)}(e)]$ with INTRA, SKIP, FORWARD, UPWARD, and BIDIRECTIONAL modes are, respectively, shown below. The second order expected pixel value, $E[\tilde{f}_i^{(n)}(e)^2]$, can be calculated in the same fashion.

$$\text{INTRA:}\quad E[\tilde{f}_i^{(n)}(e)] = (1-\rho_k(e))E[\hat{f}_i^{(n)}(e)] + \rho_k(e)E[\tilde{f}_i^{(n)}(b)] \tag{B.2}$$

$$\text{SKIP:}\quad E[\tilde{f}_i^{(n)}(e)] = (1-\rho_k(e))(\tilde{f}_i^{(n-1)}(e)) + \rho_k(e)E[\tilde{f}_i^{(n)}(b)] \tag{B.3}$$

$$\text{FORWARD:}\quad E[\tilde{f}_i^{(n)}(e)] = (1-\rho_k(e))(\hat{e}_i^{(n)}(e) + E[\tilde{f}_l^{(n-1)}(e)]) + \rho_k(e)E[\tilde{f}_i^{(n)}(b)] \tag{B.4}$$

$$\text{UPWARD:}\quad E[\tilde{f}_i^{(n)}(e)] = (1-\rho_k(e))(\hat{e}_i^{(n)}(e) + E[\tilde{f}_i^{(n)}(b)]) + \rho_k(e)E[\tilde{f}_i^{(n)}(b)] \tag{B.5}$$

$$\text{BIDIRECT:}\quad E[\tilde{f}_i^{(n)}(e)] = (1-\rho_k(e))(\hat{e}_i^{(n)}(e) + E[\bar{f}_i^{(n)}(e)]) + \rho_k(e)E[\tilde{f}_i^{(n)}(b)] \tag{B.6}$$

where $\bar{f}_i^{(n)}(e)$ is the average of two predictions using forward and backward prediction. Accurate calculation of this term requires computing and storing all inter-pixel cross-correlation values for the two reference frames in the video sequence. This amount of computation and storage is usually infeasible; thus model-based cross-correlation approximation methods are usually preferred [165].