ADAPTIVE NETWORK ABSTRACTION LAYER PACKETIZATION FOR LOW BIT RATE H.264/AVC VIDEO TRANSMISSION OVER WIRELESS MOBILE NETWORKS UNDER CROSS LAYER OPTIMIZATION

ZHAO MING

(B.Eng. (1st class Hons.), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING
(ACCELERATED MASTER'S PROGRAMME)

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2005

Acknowledgements

I would like to take this opportunity to express my great attitude towards my supervisor, Dr. Le Minh Thinh. His trust, vision and the guidance in my research work are not the only things that I have been grateful for.

I want to thank my graduate student fellows Boon Leng, Yiqun and Xiaohua. They have offered me a lot of help in video coding algorithms and programming in C and C++. I also want to thank my lab mates Xu Ce and Yu Changbin (Brad), both of them share the ideas with me and teach me how to open my mind to the outside exciting world. Our lifelong friendship will never fade with time, no matter where we are. For Lee Wei Ling, Michelle and Goh Ying Tzu, Leslie, both of them are Final Year Project students working with me, I thank them for supporting me and sharing knowledge with me.

To my friends, Yang Wei, Ren Yu, Wu Tian, Jia Hui, xiaoxin, Haibo, Liu Xin, Naixi, Li Ming, Shijie, Lu Jia, Rong Chang, Liu Ming, and to all my friends although I may not list their names here, I sincerely thank them for giving me another way of support and entertainment. Without them, life would not be so interesting for a young man who does not know how to find ways for entertainment.

The rest of my thanksgivings are for my family. They are always wishing the best for me. Their love and never-ending support are things that I treasure the most.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
ARQ	Automatic Repeat reQuest
BER	Bit Error Rate
BLER	BLock Error Rate
BSC	Binary Symmetric Channel
CABAC	Context-based Adaptive Binary Arithmetic Coding
CAVLC	Context-based Adaptive Variable Length Coding
CCIR	Consultative Committee for International Radio
CDMA	Code Division Multiple Access
CIF	Common Intermediate Format
CODECs	COder / DECoder
CRC	Cyclic Redundancy Check
DCT	Discrete Cosine Transform
DMC	Discrete Memoryless Channel
DP	Dynamic Programming
DPCM	Differential Pulse Code Modulation
EREC	Error-Resilient Entropy Coding
FEC	Forward Error Correction
FGS	Fine Granularity Scalability
FQI	Frame Quality Indicator
GOP	Group of Pictures
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications

HVS	Human Visual System
IDR	Instantaneous Decoding Refresh
IEC	International Electrotechnical Commission
IP	Internet Protocol
ISO	International Organization for Standardization
ITU	International Telecommunications Union
ITU-T	Telecommunication Standardization Sector of the ITU
JVT	Joint Video Team
LTU	Logical Transmission Unit
MAD	Mean Absolute Difference
MB	Macro Block
MDC	Multiple Description Coding
MDS	Maximal Distance Separable codes
MMS	Multimedia Messaging Services
MPEG	Moving Picture Expert Group
MSE	Mean Square Error
MTU	Maximum Transfer Unit
NAL	Network Abstraction Layer
NALU	Network Abstraction Layer Unit
NMSE	Normalized Mean Square Error
OSI	Open System Interconnection
PCS	Packet-switched Conversational Services
PDCP	Packet Data Convergence Protocol
PDU	Protocol Data Unit
PER	Packet Error Rate

PPP	Point-to-Point Protocol
PSNR	Peak-Signal-to-Noise Ratio
PSS	Packet-switched Streaming Services
QCIF	Quarter Common Intermediate Format
QoS	Quality of Service
QP	Quantization Parameter
RLC	Radio Link Control
RLP	Radio Link Protocol
RoHC	Robust Header Compression
RS	Reed-Solomon
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
RTT	Round Trip Time
RVLC	Reversible Variable Length Coding
SAD	Sum of Absolute Difference
SDP	Session Description Protocol
SDU	Service Data Unit
SEI	Supplementary Enhancement Information
SIP	Session Initiation Protocol
SNR	Signal-to-Noise Ratio
TCP	Transport Control Protocol
UDP	User Datagram Protocol
UEP	Unequal Error Protection
UMTS	Universal Mobile Telecommunication Systems

UVLC	Universal Variable Length Coding
VCEG	Video Coding Expert Group
VCL	Video Coding Layer
VLC	Variable Length Coding
VOD	Video on Demand
W-CDMA	Wideband Code division Multiple Access
WLAN	Wireless Local Area Network

Abstract

The key problem of video transmission over the existing wireless mobile networks is the incompatibility between the time-varying and error-prone network conditions and the QoS requirements of real-time video applications. As new directions in the design of wireless systems do not necessarily attempt to minimize the error rate but to maximize the throughput, this thesis first proposes a novel adaptive H.264/AVC Network Abstraction Layer (NAL) packetization scheme in terms of adaptive slice partition and "Simple Packetization" with 2 motivations: i) To take advantage of slice-coding in assisting error control techniques by localizing the burst errors occurred in wireless environment so that the end-user quality can be improved with the assistance of error concealment techniques; ii) To facilitate throughput adaptation in time-varying wireless environment so that the network or system efficiency can be improved in conjunction with lower layer error control mechanisms under cross layer optimization.

This thesis also proposes a channel adaptive H.264/AVC video transmission framework under cross layer optimization. The novel adaptive H.264/AVC NAL packetization scheme works as built-in block with other channel adaptive blocks in the proposed framework to facilitate system throughput adaptation in time-varying wireless environment. Simulation results show that compared to the system with fixed NAL packetization under fixed error control configuration, the proposed framework can adapt system throughput to the variations of channel capacity with acceptable end-user quality such that channel usage and system efficiency can be enhanced whenever the channel condition is improved. And the proposed framework also shows better end-user quality compared to the system with fixed NAL packetization under channel adaptive error control configuration.

Chapter 1

Introduction

Wireless video applications and services have undergone enormous development due to the continuous growth of wireless communications, especially after the great successful deployment of *second generation* (2G) and *2.5 generation* (2.5G) cellular mobile networks such as *Global System for Mobile Communications /General Packet Radio Service* (GSM/GPRS). There is a tremendous demand of delivery video contents over wireless mobile networks due to the dramatic development of wireless access technology when the *third generation* (3G) cellular mobile networks was introduced in the first time. The demands for fast and location-independent access to video services require most current and future wireless mobile networks to support a large variety of packet-oriented transmission modes such that the transports of *internet protocol* (IP)-based video data traffic among mobile terminals or between mobile terminals and multimedia servers are flexible enough with *Quality of Service* (QoS) guaranteed.

From end-user point of view, QoS means the video displayed quality, video playback flexibility, user initial waiting time, and delay jitter, etc. More precisely, once the play starts, it must be continuous, smooth with guaranteed image quality. On the other hand, from network point of view, QoS means those pertaining to bandwidth, end-to-end delay, and *packet error rate* (PER), *block error rate* (BLER), or *bit error rate* (BER), etc. In order to fulfill the bandwidth requirement for wireless transmission, the video data are usually compressed prior to transmission. This compressed video

data are error sensitive so that it poses many challenges for video transmission in the time-varying and highly error-prone wireless environment. Therefore, the future system design of video transmission over wireless mobile networks should provide guaranteed QoS with efficient resource allocation in wireless environment.

1.1 Video Applications in Wireless Environment

1.1.1 Wireless Video Applications

There are three mayor service categories identified by most recent video coding standardization process [1]. The first service category is circuit-switched [2] and *packet-switched conversational services* (PCS) [3] for video telephony and conferencing. Such applications are characterized by very strict delay constraints—significantly less than one second end-to-end latency, with less than 100 ms being the goal [4]. Therefore, in conversational services, the end-to-end delay has to be minimized and the synchronization between audio and video streams has to be maintained in order to avoid any perceptual disturbance. The encoding, transmission, decoding and playing are performed in real time, with full-duplex.

The second category is live or pre-recorded video *packet-switched streaming* (PSS) services [5]. In PSS application, the user typically requests pre-coded sequences stored in a server. Such services have relaxed delay constraints compared to conversational services [4]. It allows video playback before the whole video stream has been transmitted. In other words, the encoding and transmission are usually separated, decoding and display start during the transmission with a initial delay of a few seconds used for buffering, and in a near real time fashion.

The third category is video in *multimedia messaging services* (MMS) [6]. In MMS applications, the bit stream is transmitted as a whole using reliable transmission

protocols, such as ftp or http. It does not obey the delay constraints and it is not real-time processing [4,7]. The encoding, transmission, and decoding are completely separated. The recorded video signal is off-line encoded and locally stored. The transmission could start at any time upon user demands, while the decoding process at the receiver in general does not start until the completion of download.

The transmission requirements for the three identified applications can be distinguished with respected to requested data rate, the maximum allowed end-to-end delay and the maximum delay jitter. This results in different system architectures for each of these applications. Figure 1.1 shows a simplified illustration [7].

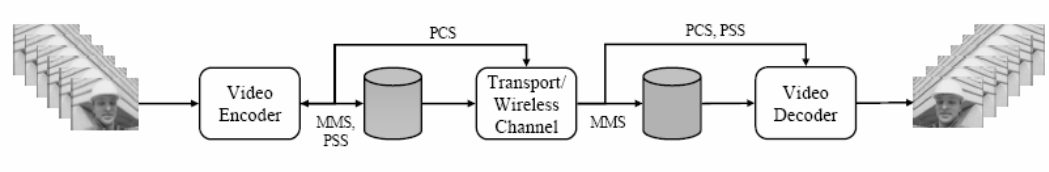


Figure 1.1: Wireless video application MMS, PSS and PCS differentiated by real-time or off-line processing for encoding, transmission and decoding

1.1.2 H.264/AVC Video Coding Standard

Digital video coding techniques, also known as video compression techniques, have played an important role in the world of telecommunication and multimedia systems where bandwidth is still a valuable commodity. Video compression techniques aim to reduce the amount of information needed for picture sequence without losing much of its quality. Currently, video coding technology is standardized by two separate standardization groups, namely the ITU-T *Video Coding Expert Group* (VCEG) and ISO/IEC *Moving Picture Expert Group* (MPEG). VCEG is older and more focusing on conventional video coding goals, such as low delay, good compression, and packet-loss/error resilience. MPEG is larger and taking on more ambitious goals, such as

"object-oriented video", "synthetic-natural hybrid coding", and digital cinema. The ITU-T video coding standards are called recommendations, and they are denoted with H.26x (e.g., H.261, H.262, H.263, H.26L and H.264). The ISO/IEC standards are denoted with MPEG-x (e.g., MPEG-1, MPEG-2, MPEG-4, MPEG-7, and MPEG-21).

In early 1998, ITU-T VCEG SG16 Q.6 issued a call for proposals on a project called H.26L [8], which targeted to double the coding efficiency compared to previous coding standards. In other words, H.26L could half the bit rate necessary for a given level of fidelity in comparison to any other existing video coding standards for a broad variety of applications. The first draft design for that new standard was adopted in October 1999. In December of 2001, VCEG and the MPEG ISO/IEC JTC 1/SC 29/WG 11 formed a *Joint Video Team* (JVT), with the mission to finalize the draft new video coding standard. The new video coding standard [9] is known as H.264/AVC, where AVC stands for MPEG-4 part 10 *Advanced Video Codec*. H.264/AVC represents a number of advances in standard video coding technology, in terms of both coding efficiency enhancement and flexibility for effective use over a broad variety of network types and application domains [10]. Its *Video Coding Layer* (VCL) provides slice-coded video streams with high compression efficiency, and its *Network Abstraction Layer* (NAL) provides network-friendly capability by packetizing the slice-coded video stream into independent and adaptive network packets, known as *NAL units* (NALUs). When use well together the new features, this latest video coding standard can provide approximately a 50% [11] bit rate saving for equivalent perceptual quality relative to the performances of previous standards.

1.1.3 H.264/AVC Video Transmission over Wireless Mobile Networks

Similar to data services, the transmission of multimedia contents such as image, audio, and video over wireless mobile networks relies on the current, recently proposed, and

emerging network protocols and architectures. Figure 1.2 shows H.264/AVC video transmission system with 7 major components: i) The source H.264/AVC encoder that compresses video into media streams in VCL, and sends the stream to NAL where NALUs are formed and ready to be delivered or uploaded to the media server for storage and later transmission on demand; ii) Application layer in charge of channel coding and packetization; iii) Transport layer 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; iv) Network layer realizes IP-based packet delivery; v) Data link layer provides radio resource allocation and media access control; vi) Physical layer where packets are delivered to the client through air interface; and vii) The receiver decompresses the video packets, and implements the interactive user controls based on the specific applications [12].

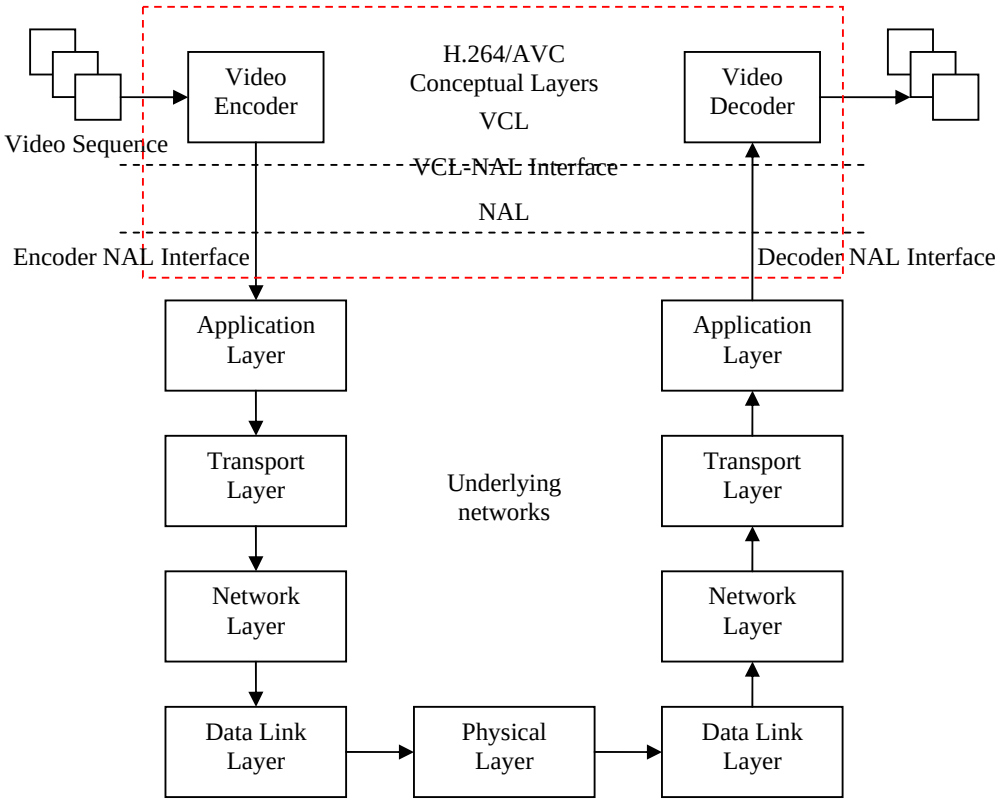


Figure 1.2: H.264/AVC video transmission system

Above system model is the simplified *Open System Interconnection* (OSI) 7-layer communication model, where application layer is the combination of traditional application layer, presentation layer and session layer. At the sender end, source bits are allocated to each video frame in VCL with bit rate constraint (e.g., available channel bandwidth) reported by lower layers. NALUs are generated in NAL by packetizing slice-coded video stream produced by VCL. After passing NALUs through the network protocol stack (e.g. RTP/UDP/IP), NALUs become transport packets, and enter a packet lossy network which can be a wired network, a wireless network, or a heterogeneous network. Some packets may be dropped in the network due to congestion, or at the receiver due to excessive delay or unrecoverable bit errors occurred in the network. To combat packet losses, packet-based *Forward Error Correction* (FEC) may be employed at application layer. In addition, lost packets may be retransmitted at transport layer or in terms of smaller blocks at data link layer if applicable. Packets that reach H.264/AVC video decoder on time are buffered in the decoder buffer. Then application layer is responsible for de-packetizing the received packets to NALUs from the decoder buffer, FEC decoding, and forwarding the intact and recovered NALUs to NAL. NAL de-packetizes NALUs to coded slices, and VCL decompresses the coded slices and displays the decoded video frames in real-time. The H.264/AVC video decoder may employ error concealment techniques to mitigate the end-user quality degradation due to loss of NALUs.

1.2 Challenge for Real-time Video Transmission

The main challenge to the real-time video communications over wireless mobile networks is how to reliably transmit video data over time-varying and highly error-prone wireless links, where fulfilling 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 wireless mobile networks is the incompatibility between the nature of wireless channel conditions and the QoS requirements (such as those pertaining to bandwidth, delay, and packet loss) of video applications. With a best-effort approach, the current IP core network was originally designed for data transmission, which has no guarantee of QoS for video applications. Similarly, the current wireless mobile networks were designed mainly for voice communication, which does not require as large bandwidth as video applications do. For the deployment of multimedia applications with video stream, which is more sensitive to delay and channel errors, the lack of QoS guarantees in today's wireless mobile networks introduces huge complications [4,7,11]. Several technological challenges need to be addressed in designing a high-quality and efficient video transmission system in wireless environment.

First of all, to achieve acceptable delivery quality, transmission of a real-time video stream typically has a minimum loss requirement. However, compared to wired links, wireless channel is much noisier due to path loss, multi-path fading, log-normal shadowing effects, and noise disturbance [13], which result in a much higher BER and consequently a lower system throughput.

Secondly, in wireless mobile networks, a packet with unrecoverable bit errors is usually discarded at data link layer according to the current standards [14]. This mechanism is not severe for traditional IP applications such as data transfer and email, where reliable transmission can always be achieved through retransmission at transport layer. However, for real-time video applications, retransmission-based techniques may not be always available due to the tight delay and bandwidth constraints.

Thirdly, since bandwidth is the scarce resource in wireless mobile communication, video data should be compressed prior to transmission. Most recent video coding

standards adopt predictive coding in the sense of motion compensation to remove spatial and temporal redundancies within frame itself or among consecutive frames, which are technically known as intra-frame coding and inter-frame coding. In addition, *variable length coding* (VLC) is adopted to compress residue video data even further. Predictive coding and VLC make the compressed video data sensitive to wireless channel errors. Even single bit error can cause the loss of synchronization between encoder and decoder due to VLC, and error propagation among frames due to predictive coding in motion compensation [15-18]. Both loss of synchronization and error propagation degrade end-user perceptive quality significantly although error concealment techniques [4,7,11,32,42] at decoder are implemented.

In the literature, above challenges could be addressed intuitively by enforcing error control, especially through *unequal error protection* (UEP) for video data that are usually of different importance. One of the main characteristics of video is that different portions of the bitstream have different importance in their contribution to the end-user quality of the reconstructed video. For example, intra-coded frames are more important than inter-coded frames. If the bitstream is partitioned into packets, Intra-coded packets are usually more important than Inter-coded packets [19]. If error concealment [32,38] 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 [20]. Error control techniques [20-21], in general, include error resilient video coding, FEC, retransmission/*Automatic Repeat reQuest* (ARQ), power control, and error concealment.

Besides error control techniques, above challenges can also be addressed by slice-based source coding. The concept of slice-coding is introduced to reduce error propagation by localizing channel errors to smaller region in the video frame. If the

slice is lost, error concealment techniques can conceal the loss within small areas. And the error propagation due to loss of slices can be minimized because slice is encoded and decoded independently. In H.264/AVC, each slice can be encapsulated into one network packet, and the smaller the network packet, the less probability it will be corrupted by channel burst errors [7]. Therefore, partitioning video frame into large number of slices is helpful to enhance error resilience of video data. However, large amount of slices per video frame will reduce the source coding efficiency and introduce additional overheads from network protocol headers. Hence, bandwidth requirement may not be fulfilled and system efficiency is reduced.

The above channel and source approaches could be jointly considered to design a high-quality and efficient video transmission system over wireless environment. Here, an efficient system is defined as system can transmit video data with acceptable end-user quality by using less source, channel, and network resources. Since new research directions in the design of wireless systems do not necessarily attempt to minimize the error rate but to maximize the throughput [7], an efficient system should be able to adapt its throughput to the variation of channel capacity so that the source, channel, and network resources are allocated subject to channel conditions.

Although current H.264/AVC wireless video transmission system [4,7,25-27] with fixed NAL packetization under fixed error control configuration has less computation and implementation complexities, in deed, it has low system throughput and end-user quality degradation due to most likely occurred over and under channel protections, which is less efficient because wireless channel is also time-varying and such system cannot response to channel variations. Meanwhile, the traditional layered protocol stack, where various protocol layers only communicate with each other in a restricted manner, has proved to be inefficient and inflexible in adapting to the constantly

changing network conditions [22]. Furthermore, conventional video communication systems have focused on video compression, namely, rate-distortion optimized source coding, without considering other layers [22-23]. While these algorithms can produce significant improvements in source-coding performance, they are inadequate for video communications in wireless environment. This is because Shannon's separation theorem [24], 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. Therefore, for the best end-to-end performance, multiple protocol layers should be jointly designed to react to the channel conditions in order to make the end-system network-adaptive. Recent research [22,59-60,70] has been focused on the investigation of joint design of end-system application layer source–channel coding with manipulations at other layers.

1.3 Contributions of the Thesis

As recent research on H.264/AVC [25-27] does not address the issues on NAL packetization to enhance error resilience and system efficiency, this thesis first proposes a novel adaptive H.264/AVC NAL packetization scheme with 2 motivations:

i) To take advantage of slice-coding in assisting error control techniques by localizing the burst errors occurred in wireless environment so that the end-user quality can be improved with the assistance of error concealment techniques;

ii) To facilitate throughput adaptation in time-varying wireless environment so that the network or system efficiency can be improved in conjunction with lower layer error control mechanisms under cross layer optimization.

With above motivations, this thesis further explores the possible solutions to the problem of coordinating slice-coding and error control mechanisms to design high

quality and efficient video transmission system over wireless environment. More precisely, this thesis also incorporates the novel adaptive H.264/AVC NAL packetization scheme to video transmission system over wireless mobile networks and proposes a channel adaptive H.264/AVC video transmission framework under cross layer optimization. Unlike the traditional approach that is trying to allocate source and channel resources by minimizing end-to-end video distortion, this framework aims to efficiently perform slice partition for NAL packetization and incorporate application layer FEC and data link layer selective ARQ to improve the system efficiency at multiple network layers.

The channel adaptive H.264/AVC video transmission framework focuses on the end-to-end system design. Such end-to-end system consists of five major channel adaptive components, namely adaptive H.264/AVC NAL packetization, end-to-end distortion estimation, channel quality measurement, bit rate estimation and error control adaptation. More specifically, the focus is on the interaction between the video codec and the underlying layers. At application layer, this framework selects slice partitions for NAL packetization and assigns FEC according to importance of the NALUs. The channel protected NALUs are attached with network protocol headers and processed by lower layers. At data link layer, selective ARQ is performed for each transport block. Here, the throughput is used as cost function in the optimization. In other words, with acceptable end-user quality, the slice partition for NAL packetization, level of FEC, and the number of allowed retransmission at data link layer within adaptation period are selected based on channel conditions such that the system throughput is adapted to variation of channel capacity with bandwidth as constraint. Furthermore, for completeness and flexibility, the proposed framework includes traditional approach of minimizing end-to-end video distortion as well.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 describes the overview of image processing and video compression techniques. Information such as color spaces, color conversions, spatial and temporal compression techniques, scalable video coding, error resilient coding, and error concealment techniques are introduced.

Chapter 3 introduces H.264/AVC video transmission in wireless environment. Concepts of NAL as well as the underlying network protocols are discussed. Mathematical models that describe the wireless environment are introduced followed by short discussion of error control techniques in terms of FEC and ARQ.

In Chapter 4, the novel adaptive H.264/AVC NAL packetization scheme is proposed. The motivation of proposing such scheme is discussed first followed by the detailed descriptions of the proposed scheme in terms of “simple packetization” and adaptive slice partition.

Chapter 5 proposes the channel adaptive H.264/AVC video transmission framework under cross layer optimization. The overall system is introduced follow by detailed discussion and analysis of each critical channel adaptive block.

Chapter 6 presents the performances of the proposed channel adaptive H.264/AVC video transmission framework in high-error and low-error channel conditions. The perfromances are also compared to video transmission system with fixed NAL packetization under fixed error control configuration and the system with fixed NAL packetization under channel adaptive error control configuration.

Chapter 7 draws to the closure of this thesis by giving the conclusion and the comments for the future work.

Chapter 2

Video Coding Techniques

Compressed video data have hierarchical organized bitstream which is different from conventional data service. In order to facilitate reliable and efficient video transmission in wireless environment, the characteristics of compressed video data should be explored. Although video coding terminology varies from standards to standards, the basic techniques are remained unchanged: i) as video is basically a sequence of images, each image should follow image processing principles; ii) for video compression techniques, spatial domain transformation and temporal domain predictive coding known as motion compensation are adopted; iii) for reliable transmission of video data, scalable video coding, error resilient video coding, and error concealment techniques are applied.

2.1. Image Processing Techniques

2.1.1. Color Spaces

Image data are represented by array of square pixels, in the form of row and columns, and the value of each pixel consists of three color components, which can be categorized into two color spaces. In computer graphic, these three color components are *Red*, *Green* and *Blue*, which are formally known as RGB values. In image and video processing, people are more familiar with *luminance* (Y for brightness) and *chrominance* (U and V or Cb and Cr for blue and red color components, respectively).

RGB uses additive color mixing and is the basic color model used in television or any other medium that projects color with light. Specifically, RGB can be thought of as three grayscale images (usually referred to as channels) representing the light values. The grayscale takes intensity value from 0 to 255, where 0 represents total black and 255 represents complete white, which can be represented by 8 bits in binary. A normal grayscale image has 8-bit color depth (256 grayscales), and a “true color” image has 24-bit color depth.

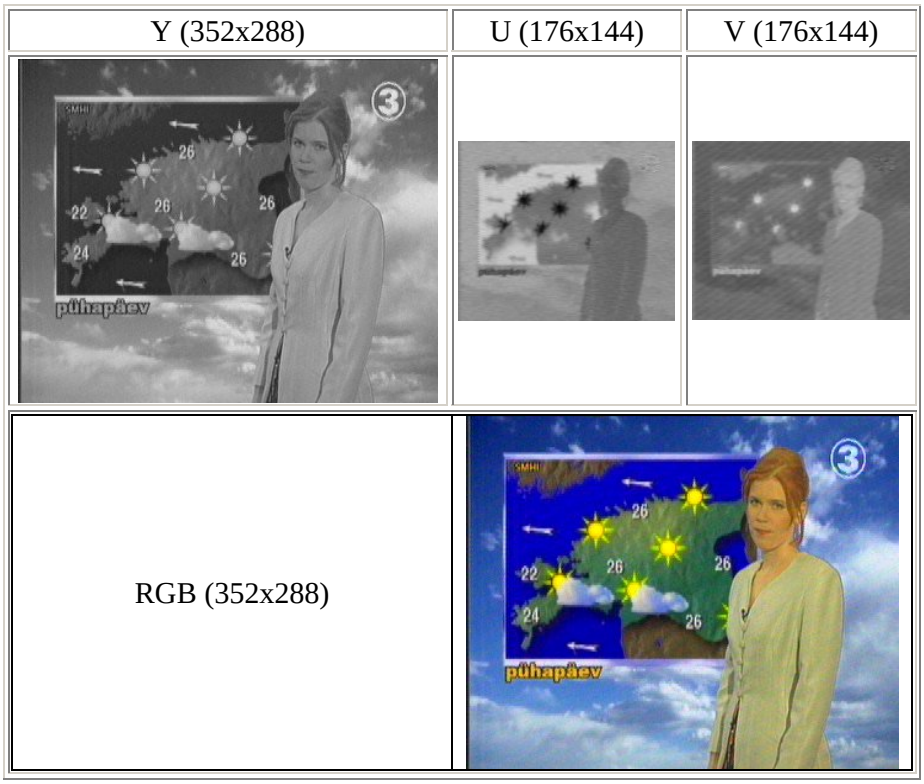


Figure 2.1: YUV image with separate components and RGB image

YUV is the color space used in the television broadcasting. The reasons to use the YUV decomposition are multiple. Firstly, a conversion of an RGB to YUV requires just a linear transform, which is really easy to do with analog circuitry and it is fast to compute numerically. Secondly, YUV allows separating the colour information from the luminance component. Since the human eye is much more responsive to luminance, chrominance components can be heavily compressed which are less likely to cause

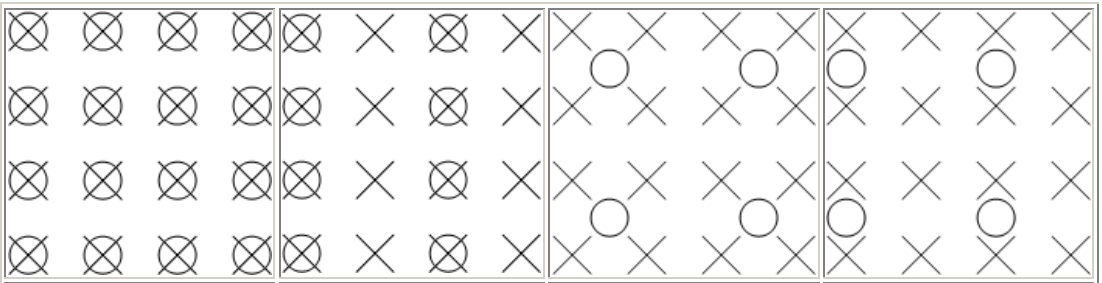
perceptible differences in the resulting image. For this reason YUV is used worldwide for television and motion picture encoding standards. Figure 2.1 shows YUV separate image components and its original RGB image. The image has dimensions of 352x288 pixels, and the YUV image is sampled at 4:2:0, which means that the U and V components are obtained by sampling half the dimentions of Y component in both horizontal and vertical directions.

2.1.2. YUV Sampling Techniques

Numerous YUV formats are defined throughout the video industry. The most common format is the 8-bit YUV formats that are recommended for video rendering in the Microsoft® Windows® operating system. One of the advantages of YUV is that the chrominance UV components can have a lower sampling rate than the luminance Y component without a dramatic degradation of the perceptual quality. A notation called the *A: B: C* notation is used to describe how often U and V are sampled relative to Y:

- 4:4:4 means no down-sampling of the chrominance components;
- 4:2:2 means 2:1 horizontal downsampling, with no vertical downsampling. Every scan line contains four Y samples for every two U or V samples;
- 4:2:0 means 2:1 horizontal downsampling, with 2:1 vertical downsampling;
- 4:1:1 means 4:1 horizontal downsampling, with no vertical downsampling. Every scan line contains four Y samples for every U or V sample. 4:1:1 sampling is less common than other formats.

Figure 2.2 shows the sampling grid used in above YUV sampling formats. Luminance samples are represented by a cross, and chrominance samples are represented by a circle. There are two common variants of 4:2:0 sampling. One of these is used in MPEG-2 video, and the other is used in MPEG-1 and in ITU-T H.26x.



4:4:4	4:2:2	4:2:0 (MPEG-1, H.26x)	4:2:0 (MPEG-2)

Figure 2.2: YUV sampling

2.1.3. Color Space Conversion

Consultative Committee for International Radio (CCIR) 601 [28] defines the relationship between YCrCb 4:4:4 and digital gamma-corrected 24-bit RGB values:

$$Y = 0.299R + 0.587G + 0.114B \tag{2.1}$$

$$C_r = 0.5R - 0.419G - 0.081B \tag{2.2}$$

$$C_b = -0.169R - 0.331G + 0.5B \tag{2.3}$$

$$R = Y + 1.402(C_r - 128) \tag{2.4}$$

$$G = Y - 0.34414(C_b - 128) - 0.71414(C_r - 128) \tag{2.5}$$

$$B = Y + 1.772(C_b - 128)\,. \tag{2.6}$$

2.1.4. Image Quality Evaluation Metric

Image reconstructed after lossy compression or transmission over error-prone environments usually has quality degradation compared to original images. Quality evaluation between the reconstructed and original images could be either subjective or objective. Subjective evaluation depends on *human visual system* (HVS) using quality rating scale, such as excellent, good, fair, poor, and bad, which varies person by person. Objective evaluation is widely used in image processing. Image distortion

measurement in the sense of Mean Square Error (MSE), Normalized MSE (NMSE), Mean Absolute Error (MAE), Sum of Absolute Difference (SAD), Signal-to-Noise Ratio (SNR) and Peak-Signal-to-Noise Ratio (PSNR) is the one of most common evaluation metrics. In [29], MSE is defined as:

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [x(i, j) - \hat{x}(i, j)]^2 \tag{2.7}$$

where M and N are the resolutions of the image, $x(i, j)$ and $\hat{x}(i, j)$ are the luminance or chrominance values at position (i, j) of original uncompressed image and reconstructed image, respectively. PSNR in dB is defined as:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} = 10 \log_{10} \frac{255^2}{\frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [x(i, j) - \hat{x}(i, j)]^2}. \tag{2.8}$$

By jointly considering luminance and chrominance components, the objective fidelity metric [30] is defined as:

$$MSE = w_Y \cdot MSE_Y + w_{C_b} \cdot MSE_{C_b} + w_{C_r} \cdot MSE_{C_r}. \tag{2.9}$$

Observe that the distortion is weighted by w_Y, w_{C_b} and w_{C_r} respectively. It is generally accepted that luminance components contributes more to the overall quality of the reconstructed frame than either of chrominance components do. In [31], $w_Y = 0.6$ and $w_{C_b} = w_{C_r} = 0.2$.

2.2. Video Compression Techniques

2.2.1. Principle behind Video Compression

A common characteristic of most images is that the neighboring pixels are most likely correlated and therefore contain redundant information. The foremost task then is to

find less correlated representation of the image. The most important components of compression are redundancy reduction, which aims at removing duplication from the signal source (image/video). In general, four types of redundancy can be identified:

- *Spectral Redundancy* between different color planes or spectral bands;
- *Spatial Redundancy* between neighboring pixel values;
- *Temporal Redundancy* between adjacent frames in a sequence of images;
- *Statistical Redundancy*, removed by *Universal Variable Length Coding* (UVLC), and *Context-based Adaptive Binary Arithmetic Coding* (CABAC).

Video compression techniques remove the redundant information in both spatial and temporal domains, and represent video sequences in minimum number of data while acceptable fidelity is maintained.

2.2.2. Spatial Domain Compression Techniques

Spatial domain compression techniques are referred as *intra-frame* (I-frame) coding. I-frame is coded independent without the knowledge from other frames. Predictive coding, scalar and vector quantization, transform coding, and entropy coding are common intra-frame coding techniques. Modern video coding standards employ most of them. Predictive coding and transform coding are highlighted as follows.

Predictive coding is originally widely used in voice communication systems known as *Differential Pulse Code Modulation* (DPCM). In intra-frame coding, it explores the mutual redundancy among neighboring pixels. Rather than encoding the pixel intensity directly, its value is first predicted from previous encoded pixels. Then the predicted pixel value is subtracted from the actual pixel value. In other words, only the prediction error is encoded instead of absolute value.

From frequency domain point of view, image data consist of low and high frequency coefficients. Human eyes are sensitive to low frequency coefficients while high frequency coefficients have less contribution to image quality. Hence, transform coding transforms image from spatial domain to frequency domain and exploits the fact that for typical images a large amount of signal energy is concentrated in a small number of coefficients at low frequencies. More precisely, image is partitioned into blocks, such as 4x4, 8x8, and 16x16, and transform coding is operated on the block basis. Transform matrix is basically a group of low pass, band pass and high pass filters which partition the power spectrum into distinct frequency band. Most image signal energy is within the low pass band and concentrates to the upper left corner of the image block as large coefficients. On the other hand, high frequency bands have less signal energy and their coefficients appearing as smaller numbers concentrate to lower right corner of the block. After quantization, the smaller high frequency band coefficients will become zero and are discarded in entropy coding. Many transform algorithms are proposed for transform coding, *Discrete Cosine Transform* (DCT) is typically used for signal has low pass characteristics, such as video and image.

2.2.3. Temporal Domain Compression Techniques

Temporal domain compression techniques are referred as *inter-frame* coding. Motion compensated predictive coding is the most important inter-frame coding technique. It consists of two core processes. The first core process is *motion estimation*, which attempts to find the most matched image regions between previous frame and current frame. After obtain the motion information, compression can be achieved by the second core process, the *compensated predictive coding*. Compensated predictive coding encodes the pixel block with motion vectors which respect to the best matched block in previous frame. There are 2 types of inter-frames, namely P-frames, which

stands for predicted frame, and B-frame, which stands for bi-directional predicted frame. P-frame is coded based on the previous coded frame, and B-frame is coded based on both previous and future coded frames. Figure 2.3 shows the I-frame, P-frame and B-frames. In video transmission, two frame orders are defined, namely the transmission order and display order. *Transmission order* is the order that encoder encodes the video sequence and decoder decodes the video data, *displays order* is the raw video sequence order that is arranged by time. Figure 2.4 shows the *group of pictures* (GOP), the arrows indicate the prediction dependencies between frames. The display order is {I0, B1, B2, P3, B4, B5, P6, B7, B8, I9}, and the transmission order is {I0, P3, B1, B2, P6, B4, B5, I9, B7, B8}.

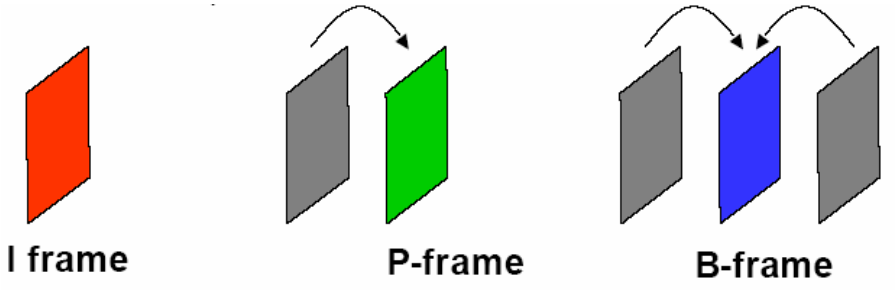


Figure 2.3: I-frame, P-frame and B-frame

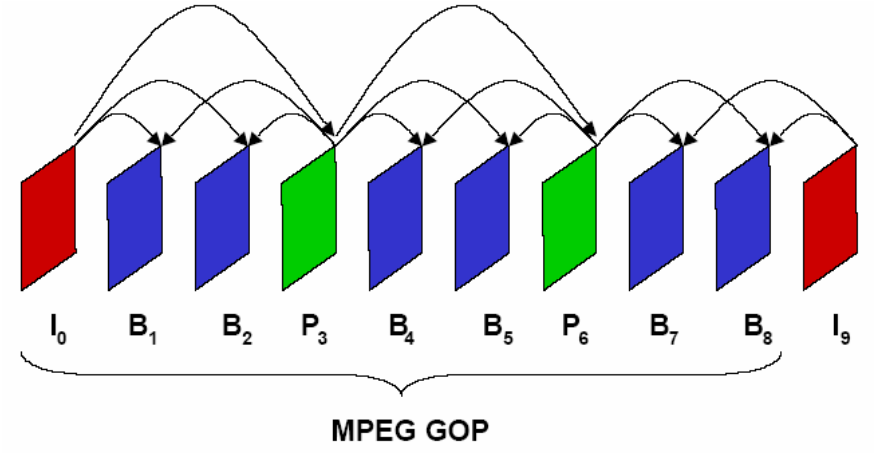


Figure 2.4: Prediction dependencies between frames

2.2.4. Scalable Video Coding Techniques

Scalable video coding techniques have many advantages over non-scalable video coding [20,31], such as compress efficiency, robustness with respect to packet loss due to channel errors or congestions, adaptability to different available bandwidths, and adaptability to memory and computational power for different mobile clients. Generally speaking, video data are coded into base layer and enhancement layers. *Base layer* carries video information with the minimum end-user quality requirement, and can be independently encoded, transmitted, and decoded to obtain basic video quality. *Enhancement layers* carry the additional video information such that the end-user quality could be improved based on that the base layer or the previous enhancement layers are received correctly.

Conceptually, scalable video coding can be classified into four categories, namely spatial scalability, temporal scalability, SNR scalability and hybrid scalability. In *spatial scalability*, the base layer is designed to generate bitstream of reduced-resolution pictures. When combined with the enhancement layer, pictures at the original resolution are produced. In *temporal scalability*, the input video is temporally demultiplexed into two pieces, with each layer carries one piece at different frame rate. In *SNR scalability*, at the base layer, a coarse quantization of the DCT coefficients is employed which results in fewer bits and a relatively low quality video. The coarsely quantized DCT coefficients are then inversely quantized (Q^{-1}) and fed to the enhancement layer to be compared with the original DCT coefficients. Their difference is finely quantized to generate a DCT coefficient refinement, which, after VLC, becomes the bitstream in enhancement layer. In *hybrid scalability*, any two of above scalable coding techniques could be combined, such as spatial-temporal scalability, SNR-spatial scalability, and SNR-temporal scalability.

2.2.5. Error-Resilient Video Coding Techniques

In wireless mobile networks, it is important to devise video encoding/decoding schemes that can make the compressed bitstream resilient to transmission errors. Many error-resilient video coding techniques [15] have been proposed in the literature. These techniques insert redundancy into the bitstream or reorder the symbols to increase the video quality in the error prone environment. *Data partition* [32] is proposed in both MPEG-4 and H.263++ onwards. Without data partition, all the syntax of the bitstream from picture level to block lever are placed nearby. Since the error can cause symbol to lose synchronization, any data after the first error are useless. In data partition mode, important data, such as headers, and motion vectors are placed at front side with stronger protection. Data partition usually works with ARQ and FEC since it only reorders the bitstream symbols. The *resynchronization marker* [33] is used to regain symbol synchronization between encoder and decoder when error occurs. It can be optimized to get a better video quality by using the rate-distortion synchronization marker insertion scheme. *Reversible variable-length codes* (RVLC) [34-35] are variable length codes that can be decoded from the opposite side. When error occurs in the middle of two resynchronization markers, the decoder can decode from forward and backward directions. RVLC results in longer codeword, which reduces the compression efficiency. *Error resilience entropy coding* (EREC) [36] is used to achieve symbol synchronization at the start of the fixed-length packet. Unlike resynchronization marker, EREC imposes little overhead which is efficient in wireless transmission where small packets are preferred. *Multiple-description coding* (MDC) [37] uses multiple video streams to represent a video sequence. It is usually combined with scalable video coding techniques. This technique results quite high overhead, and is not suitable for low bit rate applications.

2.2.6. Error-Concealment Techniques

Whenever the errors in the video bitstream cannot be corrected, error concealment techniques [32,38] can be applied. The simplest error concealment technique is to replace the current corrupt frame with previous decoded frame. Advanced error concealment techniques can be specified into spatial concealment and temporal concealment, such as spatial and temporal interpolation. Maximally smooth recovery and Projection onto convex sets [21] are proposed in literature recently. Most of error concealment techniques could be combined with error-resilient techniques, but the computation complexity would be of great concern on the portable devices.

2.3. Video Coding Hierarchy

Unlike conventional data service, video data is represented hierarchically. Input video picture is referred to as ***frame***. Frame is partitioned into ***block***, which usually has 4x4 pixels or 8x8 pixels. Several blocks form ***Macroblocks* (MBs)**, for example, one MB has 16x16 pixels, which is equivalent to four 4x4 blocks. MBs are the basic building elements where the spatial and temporal coding techniques are normally carried on, such as motion estimation. A sequence of MBs forms ***slice***, and a frame can be splitted into one or several slices. Within a frame, different slices can be grouped into ***slice groups***. The final video bitstream will have a layered structure as follows:

- Video Sequence Layer
- Group of Pictures (GOP) Layer
- Picture Layer
- Group of Block (GOB) / Slice layer
- Macroblock Layer

- Block Layer

With this hierarchy, input video frames are either intra-frame coded or inter-frame coded at encoder side. Intra-frame is intra predicted, DCT transformed, zig-zag scanned, quantized, and entropy coded. Inter-frame is motion compensated predicted, and subtracted with original input fame to produce residue coefficients, which are also DCT transformed, zig-zag scanned, quantized, and entropy coded with motion vectors. Finally, the header information for respective hierarchical layers, control information, motion vectors, and residue coefficients for both intra-frames and inter-frames form the video bitstream. At decoder, the header and control information are extracted for decoder configurations. Intra-frame residue coefficients are entropy decoded, inverse quantized, and inverse DCT transformed to reconstruct the original frame. The residue coefficients for inter-frame are entropy decoded, inverse quantized, inverse transformed, and motion compensated with entropy decoded motion vectors to reconstruct the original frame. One example of video codec is shown in Figure 2.5.

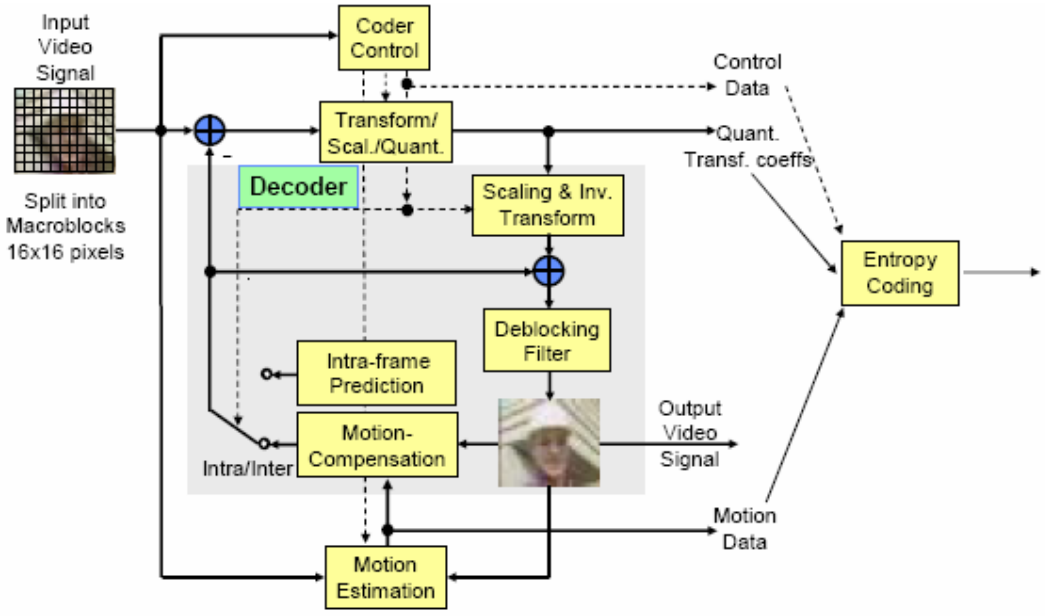


Figure 2.5: H.264/AVC video codec

In most video coding standards, the spatial and temporal domain compression techniques are employed in the implementation of encoder and decoder. Scalable coding, error-resilient coding, and error concealment coding serving as enhanced and robust coding tools, are implemented according to the requirements of applications. Encoder is much more complicated compared to decoder in the sense that the combination of various coding techniques and motion estimations consume significant computational power. In order to evaluate the features of different encoder and decoder configurations, ***profiles*** and ***levels*** are specified as conformance points. *Profile* defines a set of coding tools or algorithms that can be used in generating a conforming bitstream, whereas *level* places constraints on certain key parameters of bitstream. In H.264/AVC, three profiles [9] are defined, namely Baseline, Main, and Extended profile. All decoders conforming to a specific profile must support all features in that profile. Encoder is not required to make use of any particular set of features supported in a profile, but has to provide conforming bitstreams which can be decoded by conforming decoders.

2.4. Summary

This chapter reviews the image processing and video coding techniques. Image processing techniques include color spaces, sampling techniques, space conversion and quality evaluation metrics. Video coding techniques refer to spatial and temporal compression techniques, scalable video coding techniques, error-resilience coding, and error concealment video coding techniques. Finally, the video coding hierarchy employed in most video coding standards is presented with an example of H.264/AVC codec architecture.

Chapter 3

H.264/AVC Video Transmission in Wireless Environment

The transmission of video in wireless environments requires not only coding efficiency, but also the seamless and easy integration of the coded video into all current and possible future protocols, multiplexing and delivery architectures. H.264/AVC has been designed and proved to be attractive for wireless applications due to its low bit rate and network friendly adaptation. The low bit rate characteristic comes from the superior compression in *Video Coding Layer* (VCL) and the network friendly adaptation capability is due to the design of *Network Abstraction Layer* (NAL). NAL generates independent H.264/AVC slice-coded *NAL units* (NALUs) and deliver them as video packets to the underlying wireless networks. Various transport protocols dedicated to different network layers are designed to transmit video packets reliably and efficiently. Furthermore, mathematical models are employed to analyze the time-varying and highly error-prone characteristics of wireless channel to facilitate the performance analysis of error control techniques employed at different network layers.

3.1. H.264/AVC Network Abstraction Layer

3.1.1. Motivation of H.264/AVC NAL

The NAL is designed to provide “network friendliness”. The main motivation for introducing NAL, and its separation from VCL can be explained in twofold. First of all, the H.264/AVC recommendation [9] defines an interface between the signal

processing methodology of the VCL, and the transport oriented mechanisms of the NAL. This allows for a clean design of a VCL implementation, probably on a different processor platform than the NAL. Secondly, both VCL and NAL are designed in such way that in heterogeneous transport environments, no source-based transcoding is necessary. In other words, gateways never need to reconstruct and re-encode a VCL bit stream because of different networks environment. The NAL adapts the bitstream generated by VCL to various network and multiplex environments. It covers all syntactical levels above the slice level. In particular, it includes mechanisms [7] for:

- The representation of the data required to decode individual slices (Data that reside in picture and sequence headers in previous video coding standards);
- The start code emulation prevention;
- The support of *supplementary enhancement information* (SEI);
- The framing of the bitstream that represent coded slice for the use over bit-oriented networks.

Unfortunately, the full degree of customization of video contents to fit the needs of each particular application is outside the scope of H.264/AVC coding standard [9], but it does anticipate a variety of mappings in conceptual level. The key concepts behind NAL are NAL unit, parameter sets, access unit, and coded video sequence, which are described in following sections.

3.1.2. NAL Unit

NAL organizes the code video stream into NALUs, which contains syntax elements of a certain class. The first byte of each NALU is the header byte which indicates the data type of NALU. The 1-byte NALU header has three fixed-length bit fields in following formats:

- NALU type (T): 5-bit field indicating NALU as one of 32 different types;
- nal_reference_idc (R): 2-bit field employed to signal the importance of a NALU for the reconstruction process. A value of 0 indicates that the NALU is not used for prediction, and hence can be discarded by the decoder or by network elements without risking drifting effects. Values higher than 0 indicate that the NALU is required for a drift-free reconstruction, and the higher the value, the higher the impact of a loss of that NALU would be;
- forbidden_bit: 1-bit field specified to be zero in H.264/AVC encoding, which is reserved for error indication.

The remaining bytes are the payload data of the type indicated by header. NALU can be classified into VCL or non-VCL NALU. The VCL-NALU contains the data that represented by the values of samples in the video frames. For example, NALUs carry a coded slice, a type A, B, C data partition [39]. The non-VCL NALU contains any associated additional information such as parameter sets and SEI.

3.1.3. Parameter Sets

Parameter sets contains information that is expected to be rarely changed in decoding of a large number of VCL-NALU. This mechanism decouples the transmission of infrequently changing information from the transmission of frequently changed coded samples in the video pictures. There are two types of parameter sets: ***sequence parameter sets*** apply to a series of consecutive video pictures called coded video sequence whereas ***picture parameter sets*** apply to one or more individual pictures within a coded video sequence. Each VCL-NALU contains an identifier that refers to the relevant picture parameter set and each picture parameter set contains an identifier that refers to the content of the relevant sequence parameter set. In this manner, a small

amount of data (the identifier) can be used to refer to a larger amount of information (the parameter set) without repeating that information within each VCL-NALU.

Parameter sets can be sent either "in-band" or "out-of-band" well ahead of the VCL-NALU that they apply to, and can be repeated to provide robustness against data loss. In "in-band" applications, parameter sets may be sent within the channel that carries the VCL-NALUs. In "out-of-band" applications, it can be advantageous to convey the parameter sets using a more reliable transport mechanism than the video channel itself, for example, through *Real-time Transport Control Protocol* (RTCP) during video session initialization or feedback. Figure 3.1 shows the mechanism of the "out-of-band" transmission.

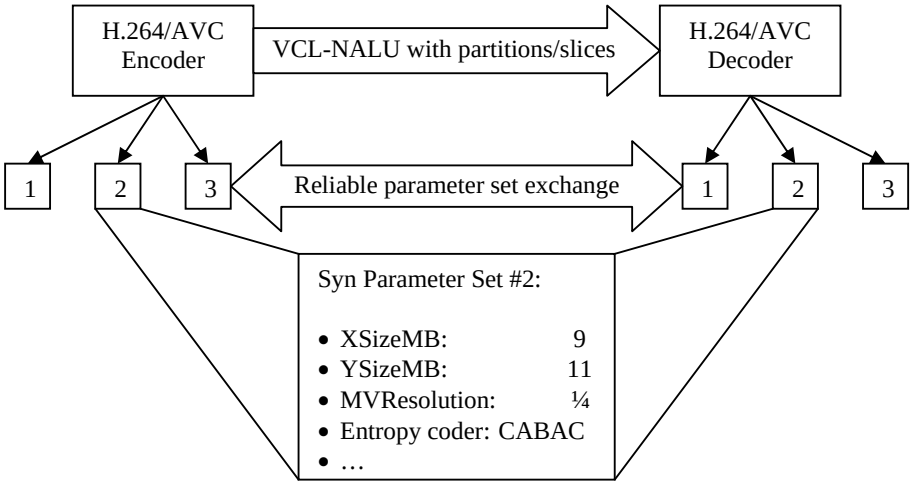


Figure 3.1: "out-of-band" transmission of parameter sets

3.1.4. Access Unit

A set of NALUs in a specified form is referred to as an access unit. The decoding of each access unit results in one decoded picture. Each access unit contains a set of VCL-NALUs that together make up a ***primary coded picture***. It may also be prefixed with an ***access unit delimiter*** to aid in locating the start of the access unit. Some SEI containing data such as picture timing information may also precede the primary coded

picture. Following the primary coded picture, there may be some additional VCL-NALUs that contain redundant representations of areas of the same video picture. They are referred to as ***redundant coded pictures***, and are available for use by a decoder in recovering from loss or corruption of the data in the primary coded pictures. Finally, if the coded picture is the last picture of a coded video sequence (a sequence of pictures that is independently decodable and uses only one sequence parameter set), an ***end of sequence*** NALU may be present to indicate the end of the sequence. And if the coded picture is the last coded picture in the entire NALU stream, an ***end of stream*** NALU may be present to indicate that the stream is ending.

3.1.5. Coded Video Sequence

A coded video sequence is similar to a GOP in previous video coding standards. It consists of a series of access units that are sequential in the NALU stream, and use only one sequence parameter set. Each coded video sequence can be decoded independently of any other coded video sequence, given the necessary parameter set information. At the beginning of a coded video sequence is an ***instantaneous decoding refresh*** (IDR) access unit. An IDR access unit contains an I-frame at the beginning, and the presence of an IDR access unit indicates that no subsequent frame in the stream will require reference to frames prior to the I-frame it contains in order to be decoded. A NALU stream may contain one or more coded video sequences.

3.2. Protocol Environment for Transport H.264/AVC Video

Protocols adopted in different network layers are usually dedicated to a particular service and referred as protocol stack. Table 3.1 shows the examples of protocol stacks for various services up to network layer. The protocols in data link layer and physical layer are network dependent.

Table 3.1: Protocol stacks for various services

<table>
<tr><td>Video
Audio
Speech</td><td colspan="2">Capacity Exchange
Scene Description
Presentation Description
Image, Graphics
Text</td><td colspan="2">Capacity Exchange
Presentation Description</td></tr>
<tr><td>Payload Formats
RTP</td><td colspan="2">HTTP</td><td colspan="2">RTSP</td></tr>
<tr><td>UDP</td><td colspan="2">TCP</td><td>TCP</td><td>UDP</td></tr>
<tr><td colspan="5">IP</td></tr>
</table>

3.2.1. Application Layer

Application layer protocols consist of data protocols and control protocols. *Real-time Transport Protocol* (RTP) [40] is the typical data protocol. It provides end-to-end network transport suitable for transmitting real-time data over multicast or unicast networks. It can be used for media-on-demand as well as interactive services. Each RTP packet consists of a 12-byte RTP header, optional payload header, and the payload itself. For H.264/AVC, NALU is mapped as RTP payload.

Control protocols are used to announce the availability of a media stream, to establish virtual or physical connections, to negotiate the capabilities between sender and receivers, and to control a running session. RTCP [41] is a control protocol which cooperates with RTP. RTCP provides support for data delivery in terms of time-stamping, sequence numbering, identification, as well as multicast-to-unicast translators. It offers QoS feedback from the receivers to the multicast group as well as support for the synchronization of different media streams. Another example of control protocol is *Real Time Streaming Protocol* (RTSP) [42], which acts as a network remote control for multimedia servers. It establishes and controls one or more time-synchronized continuous media delivery with real-time constraints. Such controls include absolute positioning within the media stream, such as 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.

3.2.2. Transport Layer

There are two protocols at transport layer, namely the *Transport Control Protocol* (TCP) and *User Datagram Protocol* (UDP) dedicated to transport layer. TCP offers a connection-oriented and guaranteed transport service, which is based on fully duplex error retransmission and timeout mechanisms for error control. Due to its unpredictable delay characteristics, it is not suitable for real-time video applications [7]. UDP offers a simple, but unreliable transport service. The 8-byte UDP header contains a checksum, which can be used to detect and remove packets corrupted by channel errors. UDP offers the same best effort service, where packets may be lost, duplicated, or reordered during transmission. In other words, packets are either perfectly received or completely lost. In wireless environment, packet loss rate are extremely high since UDP does not provide any error recovery, and traditional UDP is not efficient because it fails to incorporate the properties of the wireless channel, where channel errors only corrupt one part of the packet. UDP discards the whole packet which contains only small part of corrupt data, at such, it also throws out error-free data within the packet.

Indeed, the current and emerging multimedia coding technologies are focusing on providing error resilience so that the media decoder can tolerate a certain amount of channel errors. To support this feature, wireless systems revise the UDP protocol to reduce or avoid unnecessary packet discarding. *Reliable UDP* (RUDP) [43] was proposed to provide reliable in-order delivery up to a maximum number of retransmissions for virtual connections. RUDP can calculate the *Cyclic Redundancy Check* (CRC) based on packet header or header plus payload. *UDP Lite protocol* [44]

was proposed to prevent unnecessary packet loss at the receiver if channel errors are located only in the packet payload. The CRC is constructed based on packet header, so that only corrupt packet headers result in packet loss. UDP Lite protocol delivers packet payload, whether perfect or erroneous to the upper layers. Those erroneous packets will be corrected by application layer FEC. However, Zheng *et al.* [45] point out that existing UDP and UDP Lite protocols fails to incorporate all the channel information from physical layer so that FEC coding cannot be utilized to full effectiveness. In [45], a *complete UDP* (CUDP) protocol was proposed to capture channel information from the physical and data link layers for assistance in error recovery at the packet level by using *Maximal Distance Separable* (MDS) codes.

3.2.3. Network Layer

IP-based networks use *Internet Protocol*. It provides connectionless delivery services. Each packet is routed separately and independently regardless of its source or destination. IP provides best-effort and thus unreliable delivery services. Splitting and recombining of *service data units* (SDU) larger than *maximum transfer unit* (MTU) size is handled by IP. IP header is 20 bytes long in IPv4 and 40 bytes in IPv6.

3.2.4. Data Link Layer

Protocols in data link layer vary on different networks where circuit-switched or packet-switched transmission modes are adopted. For real-time video services over 3G mobile networks, two protocol stacks are of major interest. 3GPP has specified a multimedia telephony service for circuit-switched channels [2] based on ITU-T Recommendation H.324M [46]. For IP-based packet-switched communication, 3GPP has chosen to use *Session Initiation Protocol* (SIP) and *Session Description Protocol* (SDP) for call control [47] and RTP/UDP/IP for media transport. While the H.324 and

the RTP/UDP/IP stacks have different roots, the loss and delay effects on media data transmitting over wireless dedicated channels are very similar. Figure 3.2 shows a typical packetization of a NALU encapsulated in RTP/UDP/IP through 3GPP2 user plane protocol stack. After *Robust Header Compression* (RoHC) [48], this RTP/UDP/IP packet is encapsulated into a *Packet Data Convergence Protocol* / *Point-to-Point Protocol* (PDCP/PPP) packet that becomes a *Radio Link Control* (RLC) SDU. As video packets are of variable lengths by nature, the lengths of RLC-SDUs vary as well. If an RLC-SDU is larger than a RLC-*protocol data unit* (PDU), the RLC-SDU is segmented into several RLC-PDUs.

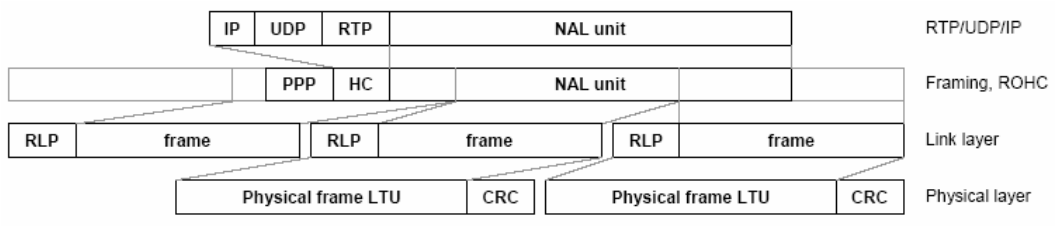


Figure 3.2: Packetization through the 3GPP2 user plane protocol stack (CDMA-2000)

The RLC protocol can operate in three modes, namely Transparent, Unacknowledged and Acknowledged mode [49]. The RLC protocol provides segmentation and retransmission services for both users and control data. The transparent and unacknowledged mode RLC entities are defined to be unidirectional and acknowledged mode entities are described as bi-directional. For all RLC modes, CRC error detection is performed on the physical layer and the result is delivered to the RLC together with the actual data. In the *transparent* mode, no protocol overhead is added to higher layer data. Erroneous PDUs can be discarded or marked as erroneous. In the *unacknowledged* mode, no retransmission protocol is in use and data delivery is not guaranteed. Received erroneous data are either marked or discarded

depending on the configuration. In the *acknowledged* mode, ARQ mechanism is used for error correction. Hence, in unacknowledged mode, if any of the RLC-PDUs containing data from a certain RLC-SDU has not been received correctly, the RLC-SDU is typically discarded. In acknowledged mode, the RLC/*Radio Link Protocol* (RLP) layer can perform retransmissions.

3.3. Mathematical Models for Wireless Channel

An analytical model of a wireless communication environment is a mathematical representation that describes the way signals are affected by interference, path loss, fading, and noise. The error performance of wireless channels is usually modeled by capturing the statistical nature of the interactions among reflected radio waves. Such statistical calculation for BER, which is generally used to characterize channel errors at the physical layer, is a well known practice in ***physical-layer oriented modeling*** [50].

From the perspective of higher layers, network protocol developers and algorithm designers are interested in block/packet errors, as most of the higher-layer applications (running on top of data link layers) exchange blocks of data between peers. For example, bit errors in a data link layer packet may result in the loss of the entire packet. Therefore, it is desirable to have accurate error models for wireless channels, which can be used by network protocol developers and network system engineers to simulate and analyze the end-to-end performance at the packet level. Such modeling is known as ***higher-layer oriented modeling*** [50]. Since this thesis focuses on the end-to-end video transmission through cross layer design, higher-layer oriented model is preferred.

Starting from the physical layer, there are many analytical channel models, such as *Binary Symmetric Channel* (BSC), *Gilbert-Elliot Channel* (two-state Markov model) [50-52], and *Rayleigh fading channel* [53]. The BSC is the simplest type of *Discrete*

Memoryless Channel (DMC) model, but if fails to explore the channel with memory. Rayleigh fading channel has been a good representation of physical layer characteristics, but it is mainly for physical layer design approaches. It has been observed empirically [54] that errors in wireless Rayleigh fading channel can be approximated by a first order two-state Markov process. Specifically, a well designed channel may enter a state where burst errors occur for a small time interval. Therefore, Gilbert-Elliot Channel model is applied in following analysis. In a simplified two-state Markov process, good state and bad state are defined as shown in Figure 3.3.

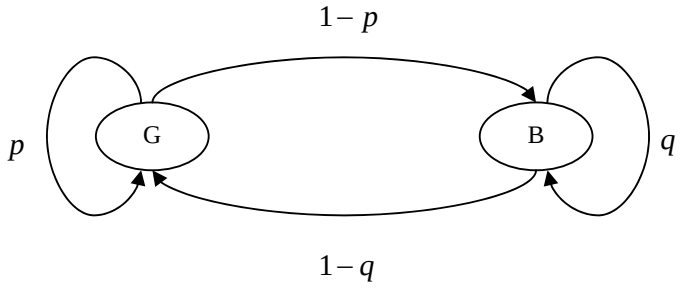


Figure 3.3: Two-state Markov model describing fading channel

The transition probability matrix is given by:

$$P_{transition}(n) = \begin{bmatrix} P_{GG} & P_{GB} \\ P_{BG} & P_{BB} \end{bmatrix}^n = \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix}^n . \tag{3.1}$$

At steady state, the state probability P_G and P_B for each state is given by:

$$\begin{bmatrix} P_G & P_B \end{bmatrix} = \begin{bmatrix} P_G & P_B \end{bmatrix} \cdot \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix}. \tag{3.2}$$

Solve equation (3.2) together with

$$P_G + P_B = 1. \tag{3.3}$$

Hence, at steady state,

$$\begin{bmatrix} P_G & P_B \end{bmatrix} = \begin{bmatrix} \frac{1-q}{2-p-q} & \frac{1-p}{2-p-q} \end{bmatrix}. \tag{3.4}$$

The expected length that the system stays in either good state or bad state is given by:

$$L_G = \frac{1}{1-p} \tag{3.5}$$

$$L_B = \frac{1}{1-q}. \tag{3.6}$$

The average BER is defined as the sum of all the products between the steady state probability and the probability of error for each state and is given by:

$$BER = P_G \cdot P_{GB} + P_B \cdot P_{BB} = \frac{(1-q)\cdot(1-p)}{2-p-q} + \frac{(1-p)\cdot q}{2-p-q} = \frac{1-p}{2-p-q}. \tag{3.7}$$

Furthermore, Markov model possesses a characteristic distribution of error-free intervals, or known as gaps. Let a correct bits gap of length i be the event that after an error of $(i-1)$ bits is received correctly, and then an error occurs again. The correct bits gap density function $p(i)$ gives the probability of a correct bits gap length i, i.e. $p(i) = \Pr[0^{i-1}1 \mid 1]$, where '1' denotes an error, and '0^{i-1}' denotes $(i-1)$ consecutive correctly received bits. The correct bits gap distribution function $P(i)$ gives the probability of a correct bits gap length greater than $(i-1)$, i.e. $P(i) = \Pr[0^{i-1} \mid 1]$. Hence, the following relations are hold:

$$p(i) = \begin{cases} q & i = 1 \\ (1-q)\cdot p^{i-2}\cdot(1-p) & otherwise \end{cases} \tag{3.8}$$

$$P(i) = \begin{cases} 1 & i = 1 \\ (1-q)\cdot p^{i-2} & otherwise \end{cases}. \tag{3.9}$$

Similarly, let an error gap of length i be the event that after a correct bit, $(i-1)$ errors occurs and then again a correct bits received. The error gap density function $q(i)$ gives the probability of a gap length i, i.e. $q(i) = \Pr[1^{i-1}0 \mid 0]$, where '0' denotes a correct bit, and '1^{i-1}' denotes $(i-1)$ consecutive errors. The error gap distribution function $Q(i)$ gives the probability of an error gap length greater than $(i-1)$, i.e. $Q(i) = \Pr[1^{i-1} \mid 0]$. Therefore, the following relations hold:

$$q(i) = \begin{cases} p & i = 1 \\ (1-p)\cdot q^{i-2}\cdot(1-q) & otherwise \end{cases} \tag{3.10}$$

$$Q(i) = \begin{cases} 1 & i = 1 \\ (1-p)\cdot q^{i-2} & otherwise \end{cases}. \tag{3.11}$$

Now let $R(n,k)$ be the probability of $(k-1)$ erroneous bits within the next $(n-1)$ bits following an error bit. It can be calculated using the recursion:

$$R(n,k) = \begin{cases} P(n) & k = 1 \\ \sum_{i=1}^{n-k+1} p(i)\cdot R(n-i,k-1) & 2 \le k \le n \end{cases}. \tag{3.12}$$

Therefore, the probability of k errors within a block of n bits is given by:

$$P_{error}(n,k) = \begin{cases} \sum_{i=1}^{n-k+1} P_B \cdot P(i)\cdot R(n-i+1,k) & 1 \le k \le n \\ 1-\sum_{k=1}^{n} P_{error}(n,k) & k = 0 \end{cases}. \tag{3.13}$$

where P_B is the steady state probability in bad state.

3.4. Error Control Techniques

Compressed video stream has strong inter-dependencies in both spatial and temporal domain. Motion compensated predictive coding and variable length coding make

compressed video stream sensitive to even single bit error. The error corrupt video packets can be discarded at the transport layer by UDP to avoid clash of the video decoder at application layer. In addition, error resilient source coding and error concealment are proposed in most video coding standards. However, they are still not sufficient to combat errors in wireless environment [17]. This is because 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. Therefore, error control techniques are employed at different layers in conjunction with UDP, error-resilient source coding, and error concealment techniques. Error control techniques generally refer to FEC and ARQ. The assignment of error control components in a video transmission system is illustrated in Figure 3.4.

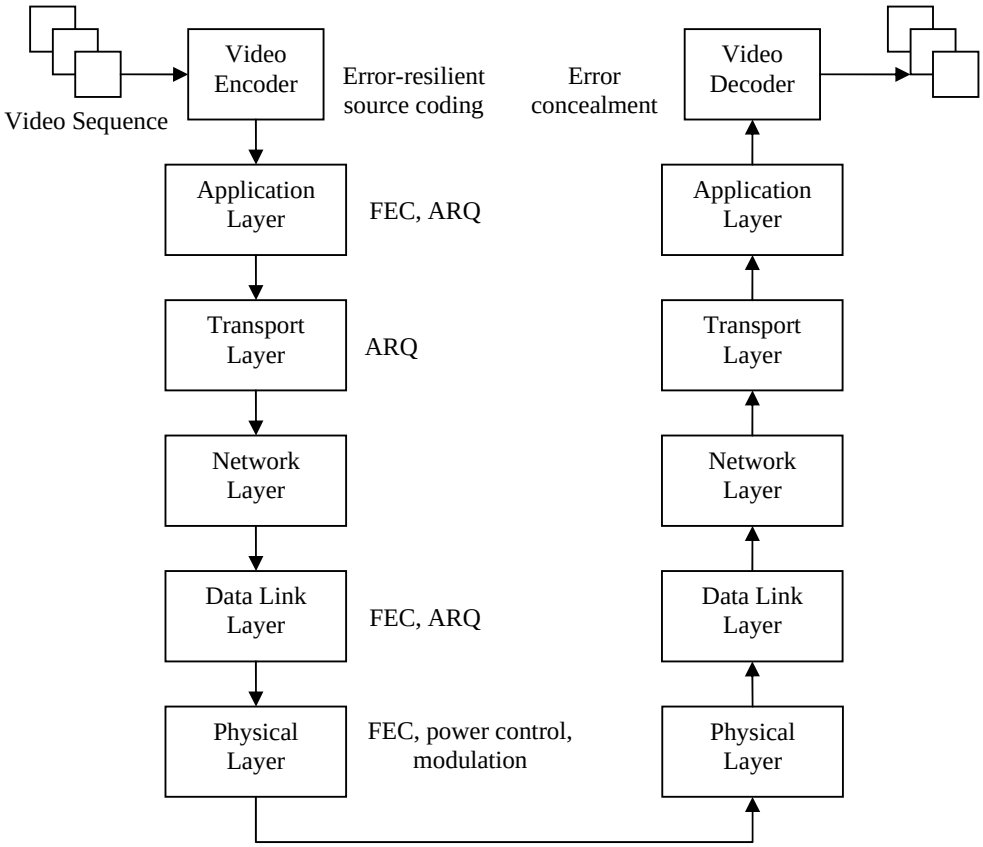


Figure 3.4: Error control techniques in video transmission system

3.4.1. Forward Error Correction

FEC is usually preferred for real-time video applications due to the strict delay requirements and its semi-reliable nature. FEC can usually be applied across packets (in the application or transport layer) [55-57] and within packets (in the data link layer) [58-60]. In applying packet-level FEC, parity packets are usually generated in addition to source packets to perform cross-packet protection. At data link layer, redundant bits are added within packet to perform byte-level FEC.

Packet-level FEC is popular in packet-switched networks [21]. A video stream is first partitioned into segments, and each segment is packetized into a group of m packets. A block code is then applied to the k packets to generate additional l redundant/parity packets resulting in a n packets block, where $n = k + l$. 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 [63-64], which have good erasure correcting properties, and have been widely used in practice. This thesis considers systematic RS codes, but the basic framework could be applied to other codes, such as BCH code and convolutional code.

A RS code is represented as $RS(n,k)$, where k is the number of source symbols and $(n-k)$ is the number of parity symbols. The code rate of a $RS(n,k)$ code is defined as k/n.The protection capability of an RS code depends on the block size and the code rate. These values are limited by the extra delay introduced by FEC. A RS code can be used to correct both errors and erasures, if an erasure occurs at where the position of an error symbol is known. A $RS(n,k)$ decoder can correct up to $(n-k)/2$ errors or up to $(n-k)$ erasures, regardless of which symbols are lost. For a packet-level RS code, channel errors are typically in the form of packet erasure, so a $RS(n,k)$

code applied across packets can recover up to $(n-k)$ lost packets. Thus, with packet losses modeled by a Bernoulli random process, the packet group failure probability (i.e., the probability that at least one of the original k packets is in error) is given by:

$$P_{Group} = 1 - \sum_{j=0}^{n-k} P(n, j) = 1 - \sum_{j=0}^{n-k} \binom{n}{j} \cdot \varepsilon_p{}^{j} (1-\varepsilon_p)^{n-j} \tag{3.14}$$

where ε_p is the individual packet loss probability, and $P(n, j)$ represents probability of j packet loss in transmission of n packets.

In a byte-level RS code, source bits in a packet are first partitioned into k symbols, and then $(n-k)$ parity symbols are generated and added to the source bits to form a packet. In this case, the noisy wireless channel causes symbol error within packets (but not erasure). As a result, the packet error probability for a $RS(n,k)$ code can be expressed as:

$$\varepsilon_p = 1 - \sum_{j=0}^{\lfloor (n-k)/2 \rfloor} P(n, j) = 1 - \sum_{j=0}^{\lfloor (n-k)/2 \rfloor} \binom{n}{j} \cdot \varepsilon_s{}^{j} \cdot (1-\varepsilon_s)^{n-j} \tag{3.15}$$

where ε_s is symbol error probability.

3.4.2. Retransmission

Although FEC is preferred for real-time video applications due to the strict delay requirements, it cannot completely avoid packet loss due to error correction capability. 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 [12,15,59-62] can automatically adapt to the channel loss characteristics by transmitting only as many redundant packets as they 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 where *Round Trip Time* (RTT) is relatively small.

There are three ARQ retransmission schemes, *Stop and Wait ARQ*, *Selective ARQ*, and *Go-Back-N ARQ*. This thesis assumes local retransmission is available at video proxy server located at the base station in wireless mobile networks, and Selective ARQ is adopted. Assume that at steady state, the packet loss probability is ε_p, let the maximum number of retransmissions for a packet be $N_{\max}$, the probability of successful transmission after j times retransmission is given by:

$$P_{succ}(j) = \varepsilon_P{}^{j}(1-\varepsilon_P) \qquad 0 \le j \le N_{\max}. \tag{3.16}$$

The probability of successful transmission within $N_{\max}$ retransmissions is given by:

$$\begin{aligned} P_{succ} &= \sum_{j=0}^{N_{\max}} P_{succ}(j) = \sum_{j=0}^{N_{\max}} \varepsilon_P^{j}(1-\varepsilon_P) \\ &= (1-\varepsilon_P) + \varepsilon_P(1-\varepsilon_P) + \varepsilon_P^2(1-\varepsilon_P) + ... + \varepsilon_P^{N_{\max}}(1-\varepsilon_P) \\ &= (1-\varepsilon_P)\cdot(1+\varepsilon_P+\varepsilon_P^2+...+\varepsilon_P^{N_{\max}}) \\ &= (1-\varepsilon_P)\frac{1-\varepsilon_P^{N_{\max}+1}}{1-\varepsilon_P} = 1-\varepsilon_P^{N_{\max}+1}. \end{aligned} \tag{3.17}$$

When $N_{\max} \to \infty$ such as in TCP, $P_{succ} \approx 1$, which means TCP ensures the reliable transmission. When $N_{\max} \ge 1$ the expected number of transmissions or the average/mean number of transmissions is given by:

$$E[i = j+1] = (1-\varepsilon_P)\cdot\sum_{j=0}^{N_{\max}}(j+1)\cdot\varepsilon_P^{j}$$

$$= (1-\varepsilon_P) + 2\cdot\varepsilon_P(1-\varepsilon_P) + 3\cdot\varepsilon_P^2\cdot(1-\varepsilon_P) + ... + (N_{\max}+1)\cdot\varepsilon_P^{N_{\max}}\cdot(1-\varepsilon_P)$$

$$= (1-\varepsilon_P)\cdot[1 + 2\cdot\varepsilon_P + 3\cdot\varepsilon_P^2 + ... + (N_{\max}+1)\cdot\varepsilon_P^{N_{\max}}]$$

$$= \frac{1-\varepsilon_P^{N_{\max}+1}}{1-\varepsilon_P} - (N_{\max}+1)\cdot\varepsilon_P^{N_{\max}+1}. \quad (3.18)$$

If $N_{\max} = 0$, then

$$E[i = j+1] = 1. \quad (3.19)$$

If $N_{\max} \to \infty$, then

$$E[i = j+1] = (1-\varepsilon_P)\cdot\sum_{j=0}^{\infty}(j+1)\cdot\varepsilon_P^j$$

$$= (1-\varepsilon_P) + 2\cdot\varepsilon_P(1-\varepsilon_P) + 3\cdot\varepsilon_P^2\cdot(1-\varepsilon_P) + 4\cdot\varepsilon_P^3\cdot(1-\varepsilon_P) + ...$$

$$= (1-\varepsilon_P)\cdot[1 + 2\cdot\varepsilon_P + 3\cdot\varepsilon_P^2 + 4\cdot\varepsilon_P^3 + ...]$$

$$= \frac{1}{1-\varepsilon_P}. \quad (3.20)$$

3.5. Summary

This chapter reviews the H.264/AVC NAL architecture, protocol environments, wireless channel models, and error control techniques that are necessary to design a reliable and efficient transport mechanism to deliver H.264/AVC video over wireless mobile networks. The transport mechanism should be dedicated to the hierarchical video stream characteristics, QoS requirements, network architecture, and wireless environments. Proper error control techniques should be chosen such that wireless channel errors can be effectively eliminated with the consideration of real-time delay and network resource constraints.

Chapter 4

Adaptive H.264/AVC Network Abstraction Layer Packetization

In H.264/AVC, slice-coded video stream is generated at VCL and packetized as independent NALUs at NAL such that NALUs can be transmitted over wireless mobile networks without any source-based transcoding at video proxy or gateways. It has been shown that partitioning video frame into many slices can improve end-user quality by localizing channel errors to smaller regions and using error concealment techniques to conceal the lost slices to minimize error propagation among video frames in hostile wireless environment. However, large number of slices per video frame will reduce VCL compression efficiency, and introduce additional overheads during NAL packetization. Hence, system efficiency is reduced even though the wireless channel condition may be improved. To deal with this dilemma, this chapter proposes a novel adaptive H.264/AVC NAL packetization scheme which partitions video frame into slices according to channel condition and encapsulates slices into NALUs through "Simple Packetization" with error indication.

4.1. The Pros and Cons of Slice-Coding in NAL Packetization

The introduction of slices to represent parts of video frame has two beneficial aspects when video data are transmitted in wireless environment. The first positive effect is to reduce the NALU error probability by using shorter packets from NAL packetization. There are two types of errors in error-prone environment, known as burst error and

random error. *Burst error* has the characteristics that the received data bits are in consecutive errors between two correct received bits. In other words, if one bit is in error, it is most likely that the next bit is in error too. Likewise, if one bit is correctly received, it is most likely that the next bit is correct. *Random error* is where the error occurs in random. Unlike burst error, each bit has an equal probability of getting in error, regardless of whether previous bit is in error or not. In wireless environment, errors usually occur in burst due to multipath fading [13]. Hence, wireless channel has been modeled as channel with memory in Chapter 3. Research has shown that the smaller the packet size, the less likely it will be hit by the burst errors [7].

The second positive effect is the resynchronization possibility within one video frame, which allows restarting the decoding process at each slice, and applying error concealment if the slice is lost. This is because each slice can be decoded independently without using the data from other slices. Hence, it can effectively minimize error propagation in such a way that burst error can be localized in a small region represented by error slices whereas other parts of video frame remain correct.

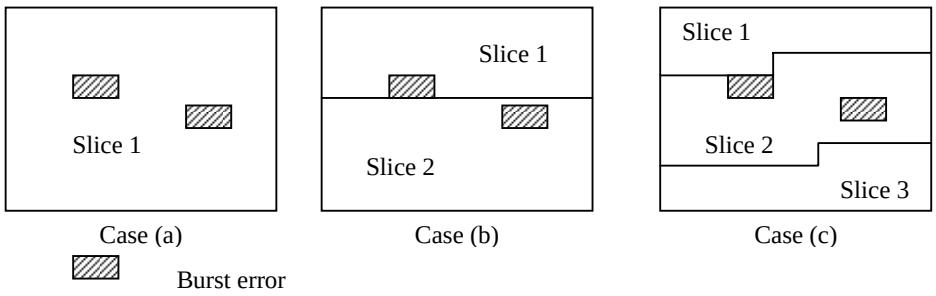
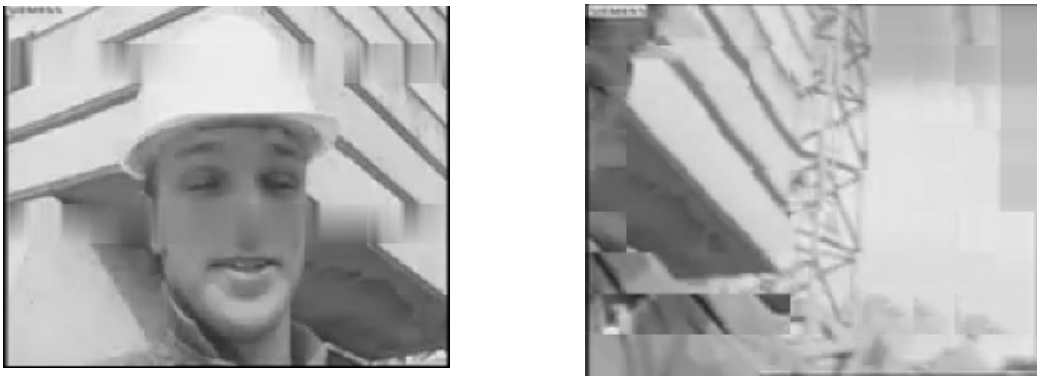


Figure 4.1: Slice partition to localize burst errors

Figure 4.1 shows the advantage of slice partition to localize the burst errors. There are two burst errors within one video frame. In both case (a) and case (b), defining 1 slice and 2 slices per video frame cannot localize the errors, and the whole video frame is corrupt. In case (c), by partitioning video frame into 3 slices, only Slice 2 is corrupt,

and both Slice 1 and Slice 3 are received correctly. The error video frame can be decoded with acceptable end-user quality by concealing Slice 2. Figure 4.2 shows the reconstructed error video frames by using intra- and inter- error concealments.



Intra error concealment　　Inter error concealment

Figure 4.2: Intra and inter error concealments with slice-coding

Figure 4.3 shows the PSNR performances for transmitting 400-frame "Foreman" sequence under JVT test conditions [65] error pattern 1. Error pattern 1 has BER 9.3×10^{-3}, which can be considered as a high-error channel. Figure 4.4 shows the PSNR performances under JVT test conditions error pattern 2. Error pattern 2 has BER 2.9×10^{-3}, which can be considered as low-error channel. In these two simulations, FEC is disabled and only maximum 3 times RLC/RLP retransmissions [60] are set. It can be seen that with the increased number of slices per video frame, the end-user quality is improved if loss of slices occurs in hostile wireless environment.

Therefore, the benefit of introducing slice-coding with multiple slices per video frame for NAL packetization to trade off constant channel protection such as FEC is obvious. The gains come from error concealment, and increase with increasing number of slices per video frame, because better concealment is possible due to increased number of correctly received neighboring MBs in case of losing a NALU with a single slice [7] encapsulated.

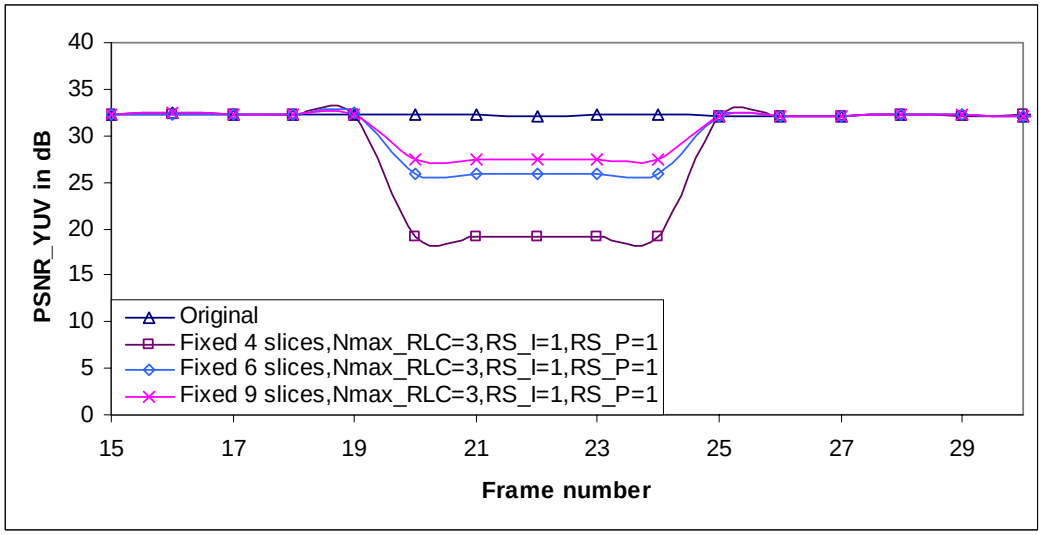


Figure 4.3: PSNR performances resulted from the transmission of "Foreman" sequence with different number of slices per video frame in high-error channel

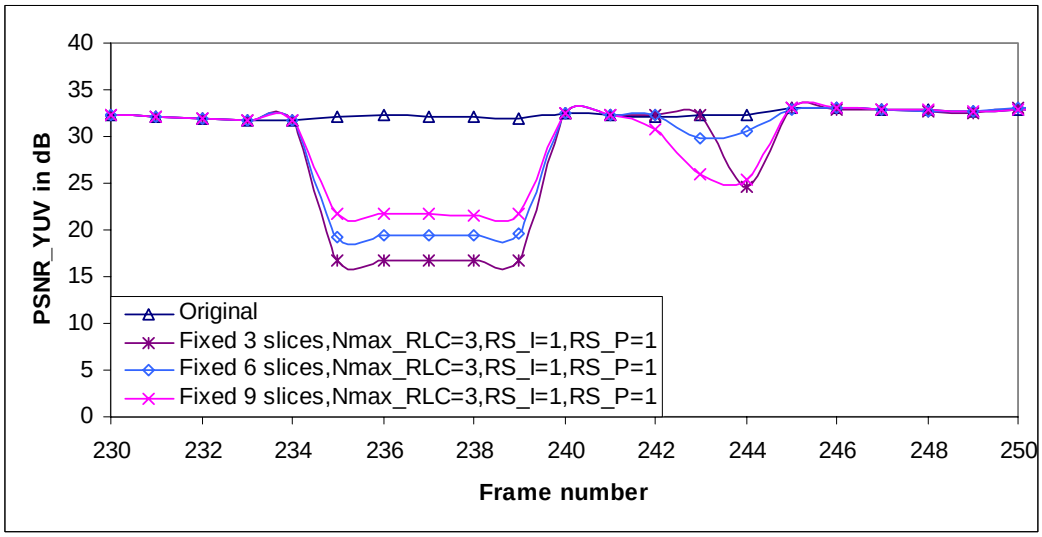


Figure 4.4: PSNR performances resulted from the transmission of "Foreman" sequence with different number of slices per video frame in low-error channel

On the other hand, although the slice-coding with multiple slices per video frame for NAL packetization has benefits in hostile wireless environment, it adversely affects source coding efficiency due to reduced prediction within the video frame, because motion vector prediction and spatial intra prediction are not allowed over slice

boundaries. The direct effect of this drawback is the sharply increase in the source bit rate. Figure 4.5 shows that the source bit rate increases as the number of slices per video frame increases in Foreman (400 frames, $QP = 36$, $\overline{PSNR\_Y} = 30.88dB$), Carphone (382 frames, $QP = 36$, $\overline{PSNR\_Y} = 31.88dB$), Suzie (150 frames, $QP = 38$, $\overline{PSNR\_Y} = 31.90dB$) and Claire (494 frames, $QP = 42$, $\overline{PSNR\_Y} = 30.83dB$) video sequences.

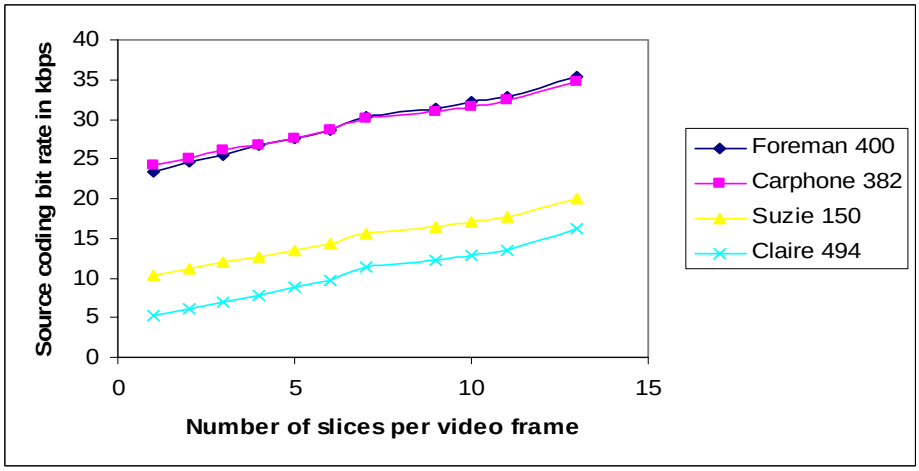


Figure 4.5: Source coding bit rate vs. Number of slices per video frame

All the video sequences are encoded at 10 frames per second (fps). Foreman and Carphone sequences represent scenes which have highest motion information. Suzie sequence represents scenes with moderate motion information, while Claire is a simple "head and shoulder" sequence with only lips and head movement. These four video sequences typically cover a wide range of scenes with different level of motion information. From Figure 4.5, it can be observed that for scenes with high motion, such as Foreman and Carphone, the source coding bit rate when there are 13 slices per video frame increases up to 51.5% over "one frame-one slice" case. And for moderate motion scene, such as Suzie, the source bit rate increases up to 94.4%. Finally, for

simple motion scene, such as Claire, the increase rate is as high as 311.2%. Subsequently, the channel bit rate increases as well.

Furthermore, as stated before, in lossy wireless environment, smaller packets are usually preferred because it is less likely to be corrupted by burst errors compared to larger packets. While there are no theoretical limitations [5] for the usage of small packet sizes, implementers must be aware of the implications of using too small RTP packets. The usage of such kind of packets would produce following drawbacks:

- $12+8+20=40$ bytes RTP/UDP/IP packet header overhead becomes too large compared to the media/source data;
- For a given media bit rate, bandwidth for the bearer allocation increases;
- The packet rate increases considerably, producing challenging situations for server, network and mobile client;
- Research in [7] shows that for 6 packets per video frame, the PSNR curve flattens out, and it decreases again for higher 12 packets per video frame due to increased packet overhead and the reduced source coding efficiency.

The packet header overhead and the payload (video data) efficiency are defined as

$$\rho_{overhead} = \frac{HeaderSize}{PacketSize} \times 100\% \tag{4.1}$$

$$\eta_{payload} = \frac{PayloadSize}{PacketSize} \times 100\% . \tag{4.2}$$

Figure 4.6 shows the bandwidth repartition among RTP payload and RTP/UDP/IP headers for different RTP payload sizes. The example assumes IPv4, which has 40-byte RTP/UDP/IP headers. The space occupied by RTP payload header is considered to be included in the RTP payload. As shown in Figure 4.6, too small packet sizes (less

than 100 bytes) give rise to RTP/UDP/IPv4 header overheads from 29% to 74%. When using large packets (greater than 750 bytes) the overhead ranges from 3% to 5%. The overheads in slice-coding for "Foreman", "Carphone", "Suzie", and "Claire" video sequences are shown in Appendix B.

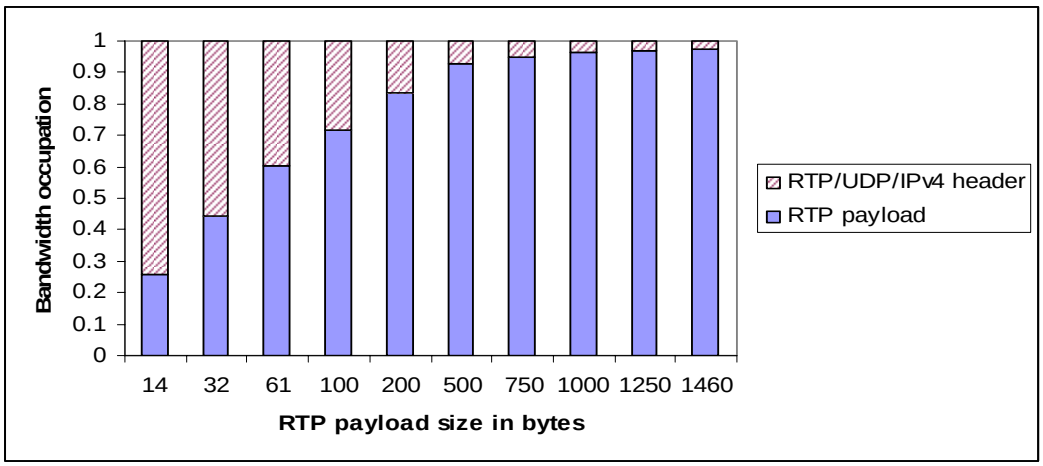


Figure 4.6: Bandwidth repartition between RTP/UDP/IPv4 header and RTP payload

From above discussions, it is obvious that slice-coding is just like a double-sided sword. It has the advantage of improving end-user quality by partitioning video frame into large number of slices for NAL packetization. However, in this case, it also reduces source coding efficiency and introduces unnecessary overheads from network protocol headers in the packetization process.

4.2. Motivation of Adaptive H.264/AVC NAL Packetization

As seen in previous discussion on the pros and cons of slice-coding, a clear optimal number of slices per video frame for NAL packetization cannot be determined. [4] and [7] claim that reasonable number of packets/slices per video frame is around 10, and the resulting packet size in this case is in the range of 100 bytes. In packet-lossy transmission over wired networks, it is all right to employ such fixed 10-slice NAL

packetization to have constant packet payload size because the wired channel is not time-varying and the error rate is sufficient low so that reasonable network throughput and system efficiency can be achieved. However, in wireless networks, as the channel changes with time, it may be hostile at the moment, but it may also be improved in the following period. Figure 4.7 shows the channel status in terms of *block error rate* (BLER) and BER for JVT test conditions error pattern 1, which characterized with mobility $3km/h$ and average BER 9.3×10^{-3}. The BLER is defined such that as long as there is at least one bit in the block is erroneous, the whole block is declared as corrupt.

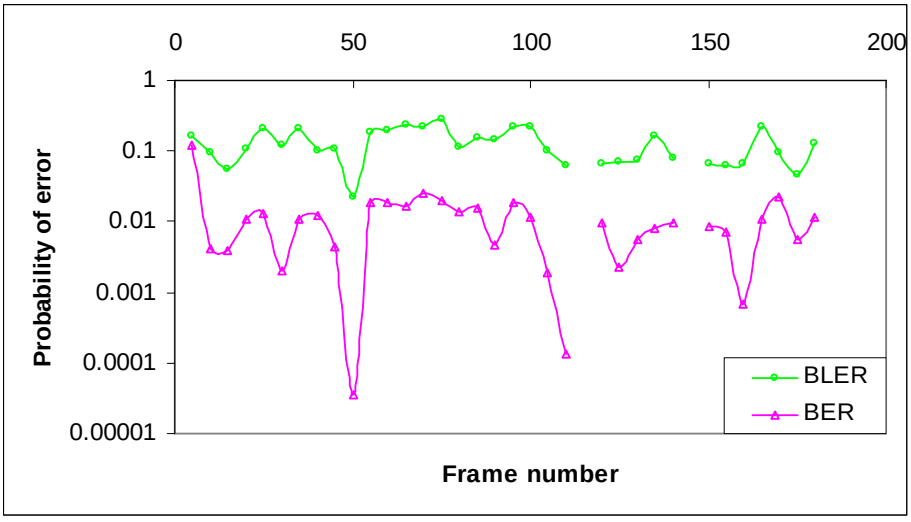


Figure 4.7: Time-varying channel status

It can be seen that the channel is continuously varying with time in the long run. In the short run, it varies slowly. And BLER generally follows the BER. The discontinuity in the figure shows that the wireless channel is error free at that particular moment, which cannot be plotted on logarithmic scale. Such time-varying nature of wireless channel means that during the error free period, more source data can be sent and therefore, scarce wireless bandwidth can be fully utilized. However, with such fixed payload size for video data regardless of channel condition, wireless channel cannot be fully utilized. In addition, this fixed NAL packetization approach without

considering channel conditions will also affect the redundancies added by lower layer FEC and ARQ if error control techniques are adopted. In other words, it is possible that the system runs into the situation that there is less channel protection for source video data when the channel is hostile, while the source video data is overly protected when channel is amiable. Such inflexibility will lead to low throughput, high transmitting power, and less efficiency of video transmission system.

Unfortunately, recent research on H.264/AVC [25-27] does not address the issues on the NAL packetization to enhance error resilience and system efficiency, fixed NAL packetizaiton scheme, more precisely, NAL packetization with fixed slice partition is adopted without considering wireless channel conditions [66]. Nevertheless, since it is possible to estimate the channel behavior for next short period based on current channel status, above observations motivate the possibility of H.264/AVC NAL packetization with slice partition adaptively to wireless channel conditions. In addition, Stockhammer *et al.* point out that it is worth noting that new directions in the design of wireless systems do not necessarily attempt to minimize the error rates in the system but to maximize the throughput [7]. This is especially appealing for services with relaxed delay constraints and certain error tolerance in the end-user quality. Therefore, the motivations of proposing a novel adaptive H.264/AVC NAL packetization scheme can be summarized as follows:

i) To take advantage of slice-coding in assisting error control techniques by localizing the burst errors occurred in wireless environment so that the end-user quality can be improved with the assistance of error-concealment techniques;

ii) To facilitate throughput adaptation in time-varying wireless environment so that the network or system efficiency can be improved in conjunction with lower layer error control mechanisms under cross layer optimization.

More specifically, the novel adaptive H.264/AVC NAL packetization scheme consists of adaptive slice partition and "simple packetization" for the partitioned slices. When wireless channel is hostile, large number of slices per video frame is preferred to assist error control techniques such that heavy channel protection may not be necessary. When the channel is amiable, smaller number of slices per video frame is preferred to avoid unnecessary overheads from network protocol headers. By doing NAL packetization in such adaptive way, system efficiency can be improved.

4.3. Adaptive H.264/AVC NAL Packetization Scheme

4.3.1. Design Constraints and Assumptions

[7] and [39] define the design constraints for H.264/AVC NALU as RTP payload, the design of adaptive H.264/AVC NAL Packetization scheme partially follows these design constraints as guideline, which has been restated as follows:

i) It should have low overhead, so that MTU sizes of 100 bytes (or less) to 64 kbytes are feasible. [4] claims that most research assumes MTU size of around 1500 bytes for wired network and MTU size of around 100 bytes for JVT's wireless common conditions;

ii) It should be easy to distinguish "important" from "less important" RTP packets, without decoding the bit stream carried in the packet;

iii) The payload specification should allow the detection of data that becomes un-decodable due to other losses, without the need to decode the bit stream. For instance, gateways should be able to detect the loss of a type A partition and if desired, react to this by not sending the type B and type C partitions;

iv) It can support NALU fragmentation into multiple RTP packets;

v) It can support NALU aggregation.

In previous research regarding video transmission [12,15-22,25-27,30,33,55-62], constant packet size is assumed. In most video coding standards, for consistent end-user quality, QPs for I-frame, P-frame, and B-frame are usually fixed with same value, which means that the source coding bit rate for P-frame is much less than I-frame, and the source coding bit rate for B-frame is significant less than both P-frame and I-frame due to spatial characteristics of scene and temporal predictive coding. In [29], the example of MPEG-2 video shows that the source coding bit rate for P-frame is 1/3 and for B-frame is 1/9 of I-frame. Hence, to achieve constant packet size in H.264/AVC, either slices from I-frame have to be fragmented or slices from P-frame and B-frame have to be aggregated in the NAL packetization. Such fragmentation and aggregation will break the independency among slices. In order to take the advantage of improving end-user quality from independent slices originated in slice-coding, maintaining the independency among slices in the packetization process is necessary. Therefore, adaptive H.264/AVC NAL packetization scheme consider I-frame, P-frame and B-frame separately with following assumptions:

i) Within I-frame, slices can be considered having roughly the constant size as long as each slice contains equal number of MBs. The average number of bits per slice is used in the analysis. Bit stuffing for NALUs is employed to ensure they roughly have equal length. This is also applied to P-slice and B-slice;

ii) For performance analysis and simulation, this thesis uses baseline profile in which only I-frame and P-frame are considered without loss of generality;

iii) Data partition is not supported although H.264/AVC supports data partition to improve end-user quality. The idea behind data partition is to realize UEP for different video data syntax elements. However, in current implementation

[9,66], I-slice in data partition mode is the same as in non data partition mode. Only P-slice and B-slice have been partitioned into different categories [11] and put into individual NALU. As P-slice and B-slice already have very small packet size, after data partition, the packet size for each partition will be even small. Such situation may bring problems of packets overheads, protection redundancy, codec complexity, etc. In deed, for low bit rate video transmission, the decoding gain from data partition is marginal [29]. And when error concealment is adopted, certain loss of video data can be tolerable and the end-user quality can be maintained at acceptable level;

iv) In order to maintain the characteristics of independency among slices and to make the NALUs pass across different network layers easier, NALU fragmentation and aggregation are beyond the scope of this thesis;

v) Since the size of sequence parameter set and picture parameter set are too small [66] (around 20 bytes) and the NALU aggregation is not considered, they are assumed to be transmitted “out-of-band”.

4.3.2. Simple Packetization

“Simple Packetization” [4] is adopted in adaptive H.264/AVC NAL packetization scheme to maintain the independency among slices. “Simple Packetization” means that one slice is packetized into one NALU, and one NALU is packetized into one RTP packet. The slice-coding and separately considering the I-frame and P-frame ensure that the video packets are less than the maximum 64kbytes MTU size defined by network layer. The NALU serves as the payload of RTP packet, but unlike in [4], in the proposed adaptive NAL packetization scheme, the 1-byte NALU header does not co-serve as RTP header, it remains as where it is. Figure 4.8 shows the “Simple

Packetization" format. The 12-byte RTP packet header is defined in [40]. In current H.264/AVC reference codec implementation [66], the F bit (forbidden bit) in NALU header is forced to zero, and the decoder assumes that the NALUs passed to the decoder are all error free. As this thesis also considers the video transmission framework with cross layer error protection and adopts UDP Lite protocol [44] in the framework, video packets corrupted by wireless channel errors may be corrected by FEC. Therefore, no matter whether NALU is corrupt or not, it will be passed to decoder. The decoder will decide whether to decode the NALU if it is error free, or bypass the NALU with error concealment. Hence, the forbidden bit in NALU header plays a very important role as error indication in the proposed adaptive NAL packetization scheme.

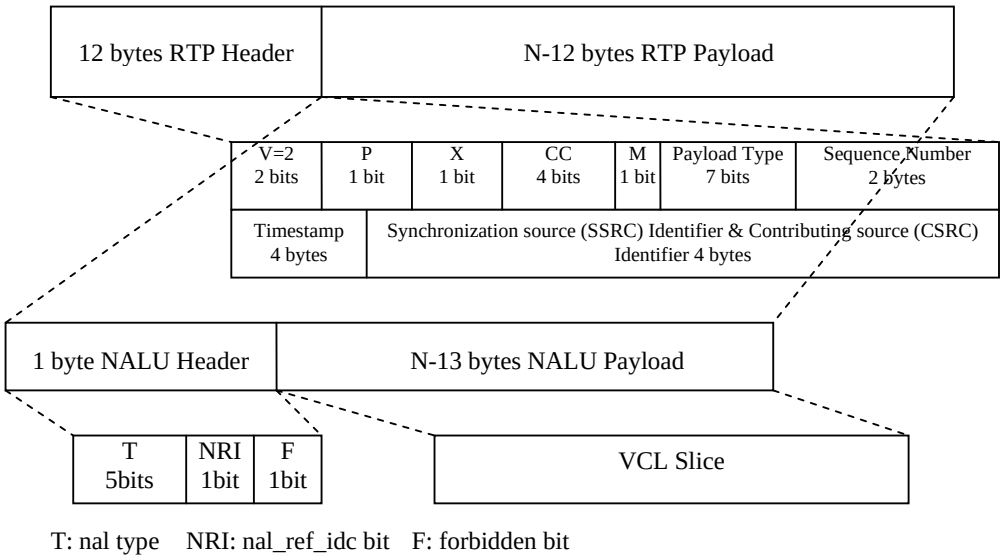


Figure 4.8: "Simple Packetization" format

4.3.3. Adaptive Slice Partition

From higher layer design point of view, NALU with RTP/UDP/IP headers attached is transmitted as blocks at data link layer and as individual bits at physical layer. In Figure 4.7, since BLER follows BER, it can be used as feedback parameter to indicate

the channel status at data link layer [60]. If NALU with RTP/UDP/IP headers has L_U bytes and one block at data link layer has L_L bytes, then the NALU with RTP/UDP/IP headers will be fragmented into N_L blocks, where N_L is given by:

$$N_L = \lceil L_U / L_L \rceil. \tag{4.3}$$

Assuming RLC/RLP at data link layer does not facilitate retransmission, and within a short period of time, assume the channel is stationary with steady state BLER represented by P_{BL}, the loss probability P_U of NALU with RTP/UDP/IP headers is given by:

$$P_U = 1 - (1 - P_{BL})^{N_L}. \tag{4.4}$$

Figure 4.9 shows P_U as a function of P_{BL} for various value of N_L. It is intuitively to see that P_U varies differently in three regions as P_{BL} increases. P_U increases fastest when P_{BL} is bellow 0.1, where it increases moderately when P_{BL} is between 0.1 and 0.2. Finally, it increase slowly when P_{BL} is above 0.2. The observation is clearer in Figure 4.10, which shows P_U as a function of N_L for various value of P_{BL}. In Figure 4.10, the distributions of P_U for P_{BL} below 0.1 are less compact to each other, which show that P_U increases fastest in this region. When P_{BL} is slightly higher than 0.1, the distributions of P_U starts to be compact, and they become very close to each other when P_{BL} is above 0.2. Therefore, the above varying trends suggest that adaptive slice partition can be based on value of P_{BL} such that whether it leads to a fast increase of P_U. Figure 4.9 and Figure 4.10 also show that error control techniques are necessary to protect video packets due to the fact that as P_{BL} increases slightly, P_U increases tremendously. This case is even worse when there are more blocks in the NALU.

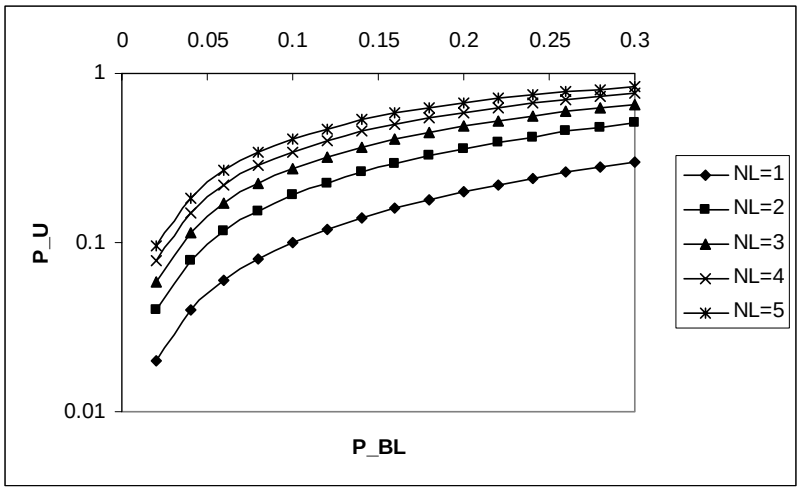


Figure 4.9: P_U **as a function of** P_{BL} **with no RLC/RLP retransmission**

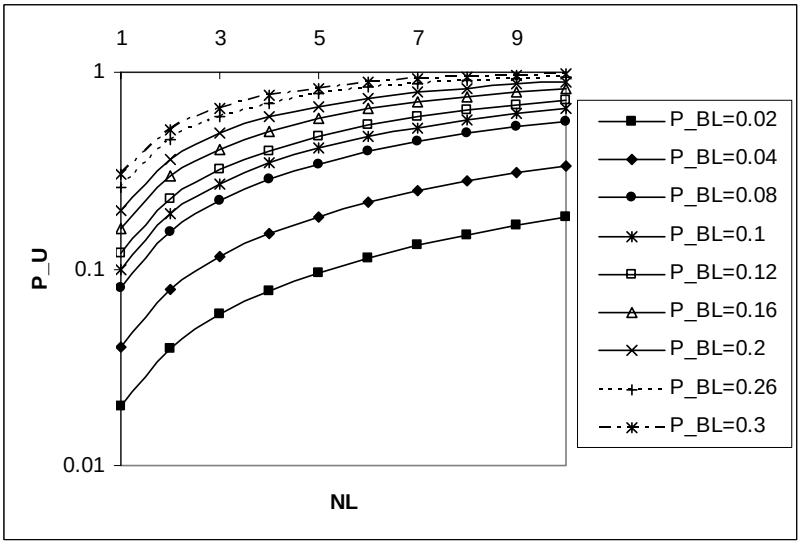


Figure 4.10: P_U **as a function of** N_L **with no RLC/RLP retransmission**

Now assume RLC/RLP at data link layer facilitates retransmissions, Figure 4.11 shows P_U as a function of $N_{\max}$ for various value of P_{BL} when $N_L = 5$, where $N_{\max}$ is the maximum number of RLC/RLP retransmissions allowed at data link layer. The derivation of performance for RLC/RLP retransmissions will be shown in the next chapter. Nevertheless, Figure 4.11 agrees with previous argument drawn by the case of no RLC/RLP retransmission.

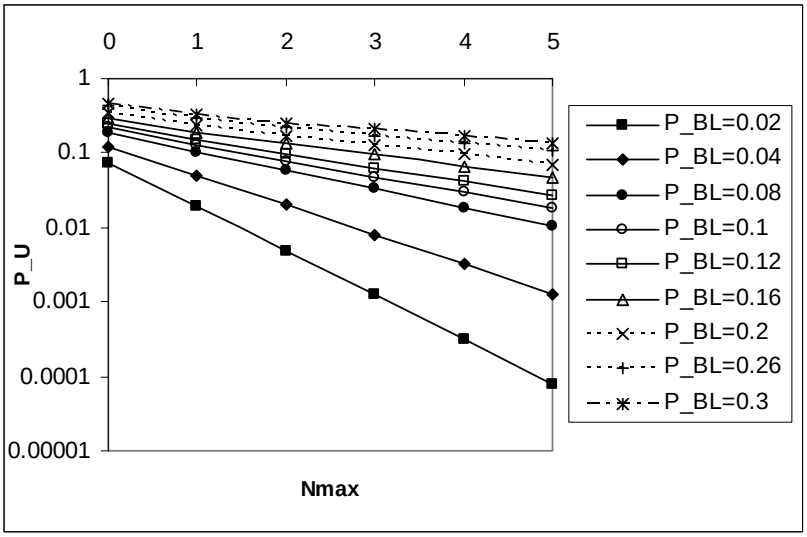


Figure 4.11: P_U **as a function of** $N_{\max}$ **with RLC/RLP retransmission (** $N_L = 5$ **)**

Hence, to represent channel condition and transition, adaptive slice partition defines global adaptive regions as state machines, which are differentiated by the fast and slow change of P_U as function of P_{BL} . The global thresholds are defined at $P_{BL} < 0.1$, $0.1 \le P_{BL} < 0.2$ and $P_{BL} \ge 0.2$, which represent *amiable state*, *noisy state* and *hostile state* respectively. The next step is to determine how to perform slice increment or decrement based on channel variations represented by channel state transitions. *Previous state* and *current state* are defined such that the slice increment or decrement step assignment is designed based on the level of improvement or degradation of BLER between previous state and current state, which represents the channel varying characteristic at that moment.

The adaptive slice partition defines eight slice changing steps for slice increment or decrement step assignment. Step 1 corresponds to increment or decrement the number of slices per video frame by one, and Step 2 corresponds to increment or decrement the number of slices per video frame by two, and so on. The higher the Step number, the more aggressive the adaptive slice partition tracks the channel. Since partitioning video

frame into one or two slices is more risky as shown in Figure 4.1, and more than 12 slices per video frame leads to end-user quality degradation [7], the minimum number of slices per video frame is lower bounded by 3, and the maximum number of slices per video frame is upper bounded by 11.

To associate above slice changing steps with level of improvement or degradation of BLER between previous state and current state, how powerful the error correction is must be investigated in each channel state as FEC is employed at application layer. Assume packet level $RS(n,k)$ is employed to group of NALUs, let P_{NALU} denote the loss probability of NALU, and as before P_U is denoted as loss probability of NALU with RTP/UDP/IP headers before passed to FEC decoder. Since $RS(n,k)$ could correct up to $(n-k)$ error packets, P_{NALU} is given by:

$$P_{NALU} = P_U \cdot [1 - \sum_{i=0}^{n-1-k} \binom{n-1}{i} \cdot P_U^i \cdot (1-P_U)^{n-1-i}]$$

$$= P_U \cdot [\sum_{i=n-k}^{n-1} \binom{n-1}{i} \cdot P_U^i \cdot (1-P_U)^{n-1-i}]$$

$$= \sum_{i=n-k}^{n-1} \binom{n-1}{i} \cdot P_U^{i+1} \cdot (1-P_U)^{n-1-i}$$

$$= \sum_{j=n-k+1}^{n} \binom{n-1}{j-1} \cdot P_U^j \cdot (1-P_U)^{n-j}$$

$$= \sum_{j=n-k+1}^{n} \frac{j}{n} \cdot \binom{n}{j} \cdot P_U^j \cdot (1-P_U)^{n-j} . \quad (4.5)$$

Figure 4.12, Figure 4.13 and Figure 4.14 show the error correction capability of packet level $RS(n,k)$ with code rate 0.5, 0.6, and 0.75 respectively.

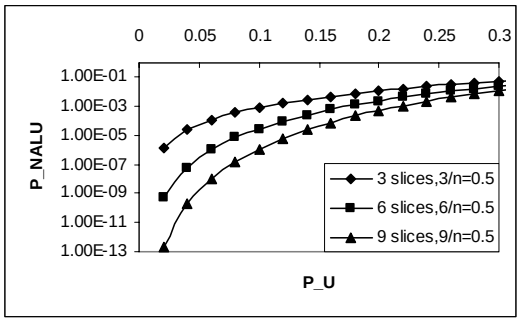


Figure 4.12: Performance of packet level $RS(n,k)$ with code rate 0.5

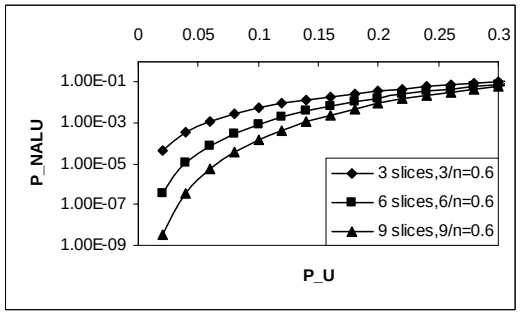


Figure 4.13: Performance of packet level $RS(n,k)$ with code rate 0.6

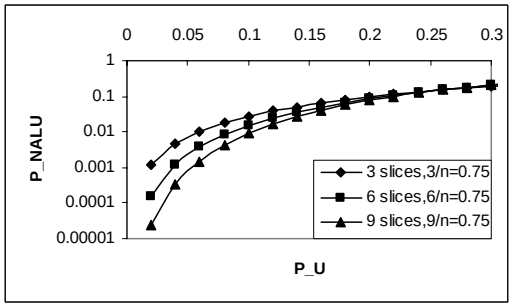


Figure 4.14: Performance of packet level $RS(n,k)$ with code rate 0.75

In each case, $k = 3,6,9$, which can be considered as 3 slices, 6 slices, and 9 slices per video frame. There are 2 observations that can be noticed. Firstly, it is intuitively to see that as the number of slices per video frame increase, the error correction capability of $RS(n,k)$ is also enhanced. This is due to the fact that $RS(n,k)$ is a MDS code. For a fixed code rate k/n, a MDS code achieves better error correction efficiency as k

increases [45]. Secondly, it is obvious that when P_U is low, the error correction is still much powerful even for smaller number of slices per video frame. Since the adaptive H.264/AVC packetization scheme intends to partition more slices per video frame when channel is hostile and fewer slices when channel is amiable, these 2 observations provide hint that the adaptive slice partition should perform slice increment or decrement much faster when it detects channel is changing faster among different states and adjust the number of slices slowly when channel is in amiable state or when channel varies slowly within noisy or hostile state.

With above observations in mind, Figure 4.15 shows the state transition diagram of a channel with slice increment or decrement step assignment. Table 4.1, Table 4.2, and Table 4.3 show the slice increment or decrement step assignment associated with the level of improvement or degradation of BLER from previous state when current state is amiable state, noisy state, and hostile state respectively.

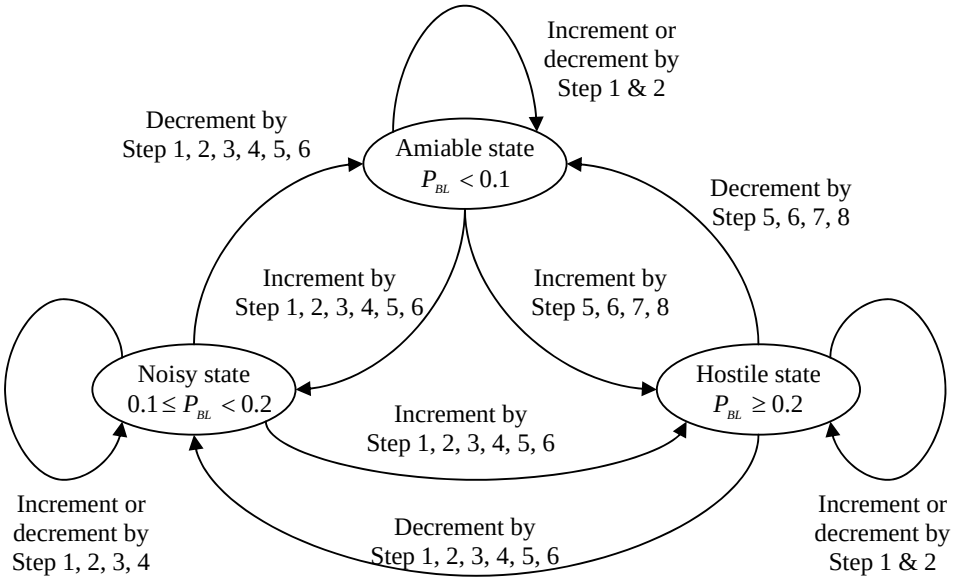


Figure 4.15: Channel state transition diagram with slice increment or decrement step assignment

Table 4.1: Slice adjusting step assignment when current state is amiable state

Previous State \ Current State	Amiable State		
Amiable State	[0, 0.3)	[0.3, 0.6)	[0.6, 1.0)
	No change	Step 1	Step 2
Noisy State	[0, 0.2)	[0.2, 0.4)	[0.4, 0.6)
	Step 1	Step 2	Step 3
Hostile State	[1.0, 1.2)	[1.2, 1.5)	[1.5, 2.0)
	Step 5	Step 6	Step 7

Previous State \ Current State						
Amiable State	[0.6, 0.8)	[0.8, 1.3)	[1.3, 2.0)			
	Step 4	Step 5	Step 6			
Noisy State	[0.8, 1.3)	[1.3, 2.0)				
	Step 5	Step 6				
Hostile State	[2.0, 3.0)					
	Step 8					

Table 4.2: Slice adjusting step assignment when current state is noisy state

Previous State \ Current State	Noisy State					
Amiable State	[0, 0.2)	[0.2, 0.4)	[0.4, 0.6)	[0.6, 0.8)	[0.8, 1.3)	[1.3, 2.0)
	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
Noisy State	[0, 0.2)	[0.2, 0.4)	[0.4, 0.6)	[0.6, 0.8)	[0.8, 1.0)	
	No change	Step 1	Step 2	Step 3	Step 4	
Hostile State	[0, 0.2)	[0.2, 0.4)	[0.4, 0.6)	[0.6, 0.8)	[0.8, 1.3)	[1.3, 2.0)
	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6

Table 4.3: Slice adjustment step assignment when current state is hostile state

Previous State \ Current State	Hostile State			
Amiable State	[1.0, 1.2)	[1.2, 1.5)	[1.5, 2.0)	[2.0, 3.0)
	Step 5	Step 6	Step 7	Step 8
Noisy State	[0, 0.2)	[0.2, 0.4)	[0.4, 0.6)	[0.6, 0.8)
	Step 1	Step 2	Step 3	Step 4
Hostile State	[0, 0.3)	[0.3, 0.6)	[0.6, 1.0)	
	No change	Step 1	Step 2	

Previous State \ Current State			
Amiable State	[0.8, 1.3)	[1.3, 2.0)	
	Step 5	Step 6	
Noisy State	[0.8, 1.3)	[1.3, 2.0)	
	Step 5	Step 6	
Hostile State			

If current state is an amiable state, which occurs commonly in low-error channel, because BLER is low, RLC/RLP retransmission at data link layer is very powerful in amiable state as shown in Figure 4.11, and after FEC at application layer as shown in Figure 4.12 to Figure 4.14, NALUs will be almost error free. Hence, smaller number of slices per video frame is preferred in order to maximize system throughput by improving source coding efficiency and avoiding unnecessary overheads from network protocol headers. If previous state is also an amiable state, the slice increment and decrement are set to be slow varied. For instance, in Table 4.1, there will be only one step increment or decrement until more than 30% and less than 60% change in BLER. If the previous state is a noisy state or a hostile state, then the adaptive slice partition is more aggressive to decrease the number of slices per video frame.

If current state is a noisy state, which is very common in high-error channel, as seen from Figure 4.11, the RLC/RLP retransmission at data link layer is not sufficient in noisy state to guarantee error free transmission. Therefore, more slices per video frame have been assigned to assist application layer FEC to improve end-user quality. Meanwhile, system throughput can be enhanced because heavy channel protection is not necessary since decoding gains can come from error concealment for lost slices. The slice increment or decrement is more sensitive to small changes of BLER. For instance, in Table 4.2, when the previous state is also a noisy state, the slice increment or decrement step is changed with 20% change in BLER. If previous state is either an amiable state or a hostile state, the step change to current noisy state is aggressive to track the fast variation of the channel.

If current state is a hostile state, which occurs quite rarely even in high-error channel, large number of slices per video frame can localize the burst errors into smaller regions in the video frame, and error control mechanisms adopted in the

system can be assisted even further as discuss above. Furthermore, as the system throughput is already quite low in hostile state, the overheads from network protocol headers are less significant compared to loss of NALU payload. Therefore, large number of slices per video frame which approaches upper limit of 11 is preferred. If previous state is also a hostile state, the slice increment and decrement are set to be slowly varied as seen in Table 4.3. If previous state is either an amiable or a noisy state, the slice increment is more aggressive.

4.3.4. Numerical Results

Figure 4.16 shows the performance of adaptive slice partition in transmission of 400 frames "Foreman" sequence encoded with $QP = 36$ based on channel quality measurement of BLER in high-error channel. As expected, the adaptive slice partition can track the channel status when BLER is monotonically increasing or decreasing. And the plot of number of slices per video frame is roughly the right shift version of the plot of BLER. It has been noticed that, there are mismatches from frame 230 to 240. This is reasonable because realistically, the adaptive slice partition is adopted based on current channel condition to estimate channel status for next period of time, which is always behind the channel changes. Whenever the channel changes too fast, it is not able to track the variation immediately. Nevertheless, the proposed adaptive NAL packetization scheme does perform 3-slice to 11-slice partition adaptively to channel conditions for NAL packetization.

Figure 4.17 shows the performance of adaptive slice partition based on channel quality measurement of BLER in low-error channel. Low-error channel condition is improved compared to high-error channel in the sense that noisy state and hostile state are less likely to occur. The proposed adaptive NAL packetization scheme prefers 3-

slice or 4-slice partition for NAL packetization whenever channel is in amiable state, and similar conclusions as shown in the case of high-error channel can be drawn.

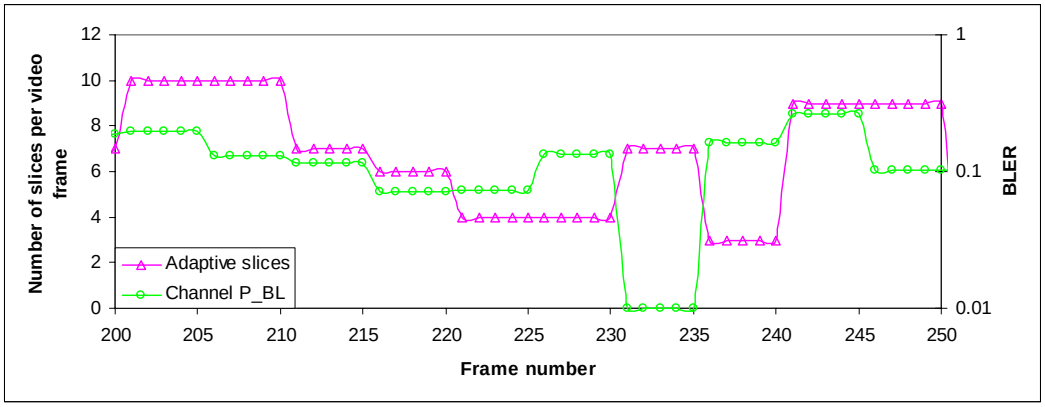


Figure 4.16: Performance of adaptive slice partition in high-error channel

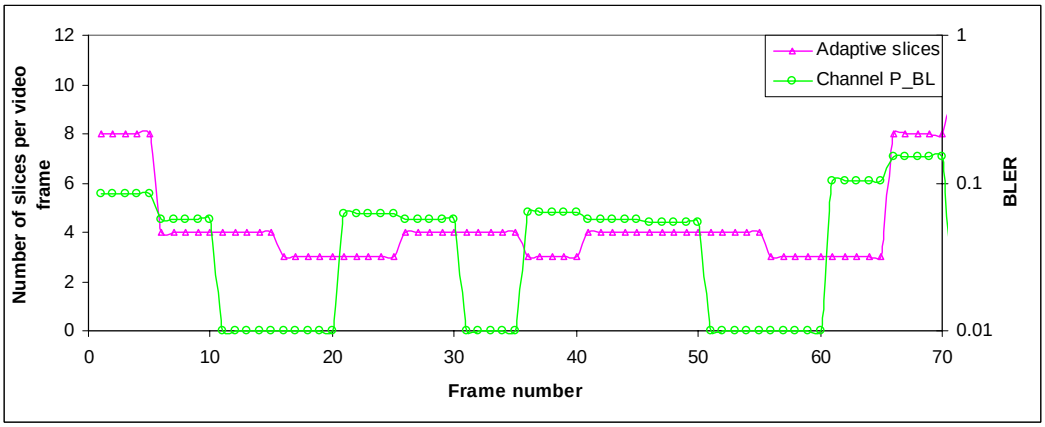


Figure 4.17: Performance of adaptive slice partition in low-error channel

Figure 4.18 shows the PSNR performances of proposed adaptive NAL packetization scheme compared with fixed NAL packetization scheme in high-error channel. As before, FEC is disabled in the simulation to see the advantage of adaptive slice partition in improving end-user quality. Such no error control situation will lead to failure of H.264/AVC video decoder in high-error channel condition. Hence, maximum of one RLC/RLP retransmission is allowed at data link layer in order to avoid early failure of video decoder. It can be seen that the proposed adaptive NAL

packetization scheme tries to combat the channel changes and it performs 2-10dB better than the fixed NAL packetization scheme with fewer slices per video frame, such as 5 slices. And the proposed NAL packetization scheme performs comparable with fixed NAL packetizaiton scheme with larger number of slices per video frame, such as 8 slices. However, fixed 8-slice NAL packetization scheme makes the decoder fail at frame 75, while decoding process still continues in the proposed adaptive NAL packetization scheme.

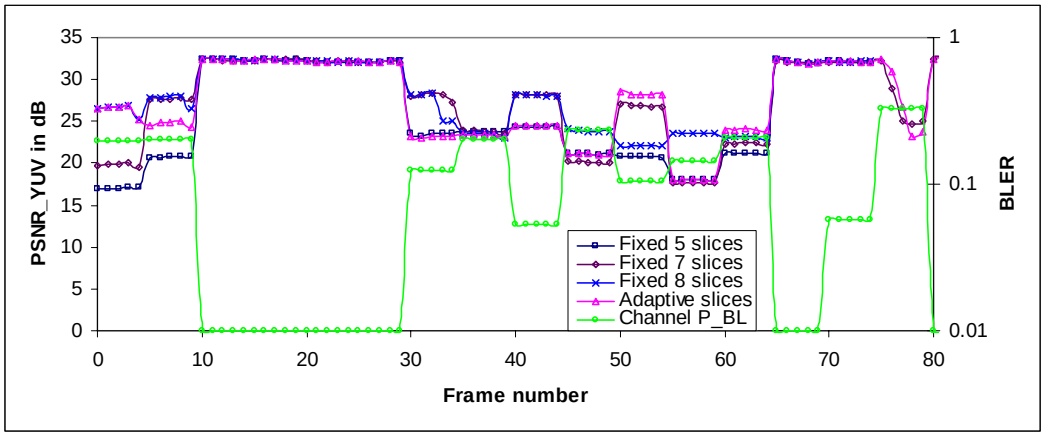


Figure 4.18: PSNR performances of proposed adaptive NAL packetization scheme and fixed NAL packetization scheme in high-error channel

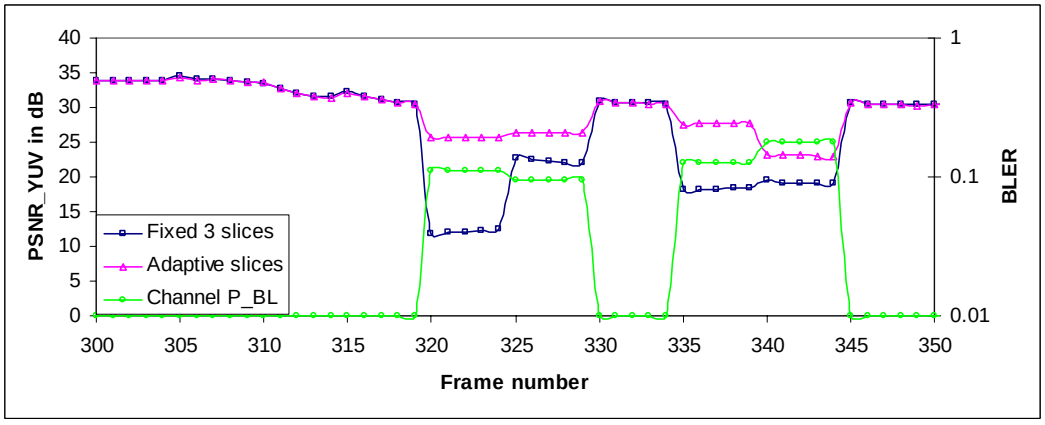


Figure 4.19: PSNR performances of proposed adaptive NAL packetization scheme and fixed NAL packetization scheme in low-error channel

Figure 4.19 shows the PSNR performances of proposed adaptive NAL packetization scheme compared with fixed NAL packetization scheme in low-error channel. It can be seen that the proposed adaptive NAL packetization scheme has 5-12dB gain over fixed NAL packetization scheme with fewer slices per video frame, such as 3 slices. Similar to high-error channel, it can be deduced that the proposed adaptive NAL packetization scheme has similar PSNR performance compared to fixed NAL packetization scheme with larger number of slices per video frame because from improving end-user quality point of view, large number of slices per video frame is always preferred for NAL packetization.

In addition, there are some frames under fixed NAL packetization scheme whose PSNR is far more below the minimum acceptable objective quality of 22dB [29]. For instance, frame 1 to 10 in high-error channel under fixed 5-slice or 7-slice NAL packetization scheme, frame 320 to 330 and frame 335 to 345 in low-error channel under fixed 3-slice NAL packetization scheme have PSNR less than 20dB. Such cases occurred rarely in the proposed adaptive NAL packetization scheme even though it may not track the channel variation immediately.

Therefore, it can be concluded that under such weak or almost no error control conditions, the proposed novel adaptive H.264/AVC NAL packetization scheme has the advantage of improving end-user quality over fixed NAL packetization scheme. Meanwhile, in contrast to fixed overheads introduced by fixed NAL packetization scheme, the overheads induced by the proposed NAL packetization scheme vary from 7.6% to 20.9% for I-slice, and 41.5% to 65.8% for P-slice as shown in columns $QP = 36$ of Table B.1 in Appendix B. This advantage can be used to improve system efficiency because the proposed NAL packetization with adaptive slice partition can assist error control mechanisms adopted in the system in the sense that constant

redundant data for channel protection is reduced when wireless channel is noisy, and the unnecessary overheads from network protocol headers is avoided whenever the wireless channel condition is improved.

4.4. Summary

This chapter presents the novel adaptive H.264/AVC NAL packetization scheme in terms of "Simple Packetization" and adaptive slice partition. The proposed scheme intends to partition video frame into slices adaptively to channel conditions and encapsulate those slices into NALUs with error indication.

Simulation results between proposed adaptive NAL packetization scheme and fixed NAL packetizatin scheme when minimum error control mechanisms are employed in the video transmission system under both high-error and low-error channel conditions show that: i) compared to fixed NAL packetization scheme with a few slices per video frame, the proposed adaptive NAL packetization scheme provides significant decoder gain of 2-12dB. This improvement has a significant impact because video frames which were originally unacceptable objectively are now acceptable; ii) compared to fixed NAL packetization scheme with larger number of slices per video frame, the proposed adaptive NAL packetization scheme offers comparable PSNR performance using fewer slices per video frame. This advantage in improving end-user quality can be used to enhance system efficiency by considering the novel adaptive H.264/AVC NAL packetizaiton scheme as a built-in block in the channel adaptive H.264/AVC video transmission framework proposed in next chapter.

Chapter 5

Channel Adaptive H.264/AVC Video Transmission Framework under Cross Layer Optimization

The novel adaptive H.264/AVC NAL packetization scheme discussed in Chapter 4 has shown that it can be used to partition video frame into slices adaptively by tracking the channel characteristics, and to encapsulate slice into NALU through “Simple Packetization” with error indication. Its ability of improving end-user quality by partitioning video frame into independent slices adaptively to channel conditions can assist error control mechanisms adopted in the video transmission system compared to the fixed NAL packetization scheme. This advantage can be used to improve system efficiency. Since new research direction [7] emphasizes on maximizing system throughput rather than minimizing bit errors, a *channel adaptive H.264/AVC video transmission framework under cross layer optimization* is proposed in this chapter. Unlike the traditional approach that is trying to allocate source and channel resources by minimizing end-to-end video distortion in the design of wireless video transmission system, this framework coordinates the proposed *novel adaptive H.264/AVC NAL packetization scheme* and error control techniques adopted in different layers to work towards global optimal solution to enhance system efficiency such that the system throughput can be adapted to the variation of channel capacity. Furthermore, for completeness and flexibility, the traditional approach of distortion minimization is included in the framework as well.

5.1. Single Layer Approach vs. Cross Layer Approach

An important aspect of wireless mobile communication networks is their dynamic behavior. In order to efficiently utilize limited network resources such as bandwidth, spectrum and energy, the end-to-end system needs to be adaptive to the changes of wireless network conditions by properly configuring error control mechanisms. Currently, each network layer provides a separate solution by providing its own optimized adaptation and protection mechanisms. In [67], Shan and Zakhor presented a novel integrated application layer packetization, scheduling and protection strategies for wireless transmission of non-scalable coded video. In [57], Majumdar *et al.* address the problem of resilient real-time video streaming over IEEE 802.11b *Wireless Local Area Network* (WLAN) for both unicast and multicast transmission. For the unicast scenario, a hybrid ARQ algorithm that combines FEC and ARQ is proposed. For the multicast case, progressive video coding based on MPEG-4 *Fine Granularity Scalability* (FGS) is combined with FEC. Similarly, Hybrid ARQ schemes, where the rate of the associated FEC is adaptively changed based on the underlying channel conditions, have also been presented by Wang and Zhu in [38], and by Ma and Zarki in [68]. However, it should be pointed out that the protection strategies described in these papers are implemented at the application layer, and do not exploit the mechanisms available in the lower layers of the protocol stack. Such research efforts have mainly focused on adaptive error control strategies at the application layer. Meanwhile, the conventional layered protocol stack, where various protocol layers can only communicate with each other in a restricted manner, has also proved to be inefficient and inflexible in adapting to the constantly changing in network conditions [22]. Hence, such *single layered* strategy does not always result in an optimal overall performance for video transmission.

Cross layer design of multimedia transmission aims to improve the system's overall performance by jointly considering multiple protocol layers. In [69], Girod and Färber give an excellent review of the existing solutions for combating wireless transmission errors in cellular networks. Their focus is on channel-adaptive source coding schemes that are useful when real-time channel feedback is available to the encoder. Importantly, joint consideration of network and application layers is mentioned as an interesting area for further research. Therefore, in existing wireless mobile environment, different protection strategies exist at the various layers of the protocol stack, and hence a joint cross layer consideration is desirable in order to provide an optimal overall performance for the transmission of video. In [70], Krishnamachari *et al.* propose a novel adaptive cross-layer protection strategy for enhancing the robustness and efficiency of scalable video transmission over WLANs by performing tradeoffs between throughput, distortion, and delay depending on the application requirements. And in [59-60], Zhang *et al.* propose a channel adaptive resource allocation scheme for transmission of scalable MPEG-4 video over 3G wireless networks. In such framework, bits are optimally distributed among source coding, channel coding, and ARQ based on distortion/power-minimized evaluation metric subject to varying channel/network conditions. The results of above cross layer design show a significantly improved visual performance for the transmitted video over a variety of channel conditions.

5.2. Overview of Channel Adaptive H.264/AVC Video Transmission Framework through Cross Layer Design

For real-time H.264/AVC video transmission, cross layer design involves the video codec and all the underlying network layers, with the emphasis on the interactions among them, so as to improve the performance of video delivery by given the resource

constraints. The main focus is on the end-to-end system design under the assumption that the lower layer provides a set of given adaptation components. Hence, under such cross layer design philosophy, the proposed channel adaptive H.264/AVC video transmission framework consists of five critical channel adaptive blocks as shown in Figure 5.1, namely, i) adaptive H.264/AVC NAL packetization, ii) end-to-end distortion estimation, iii) channel quality measurement, iv) bit rate estimation, and v) error control adaptation.

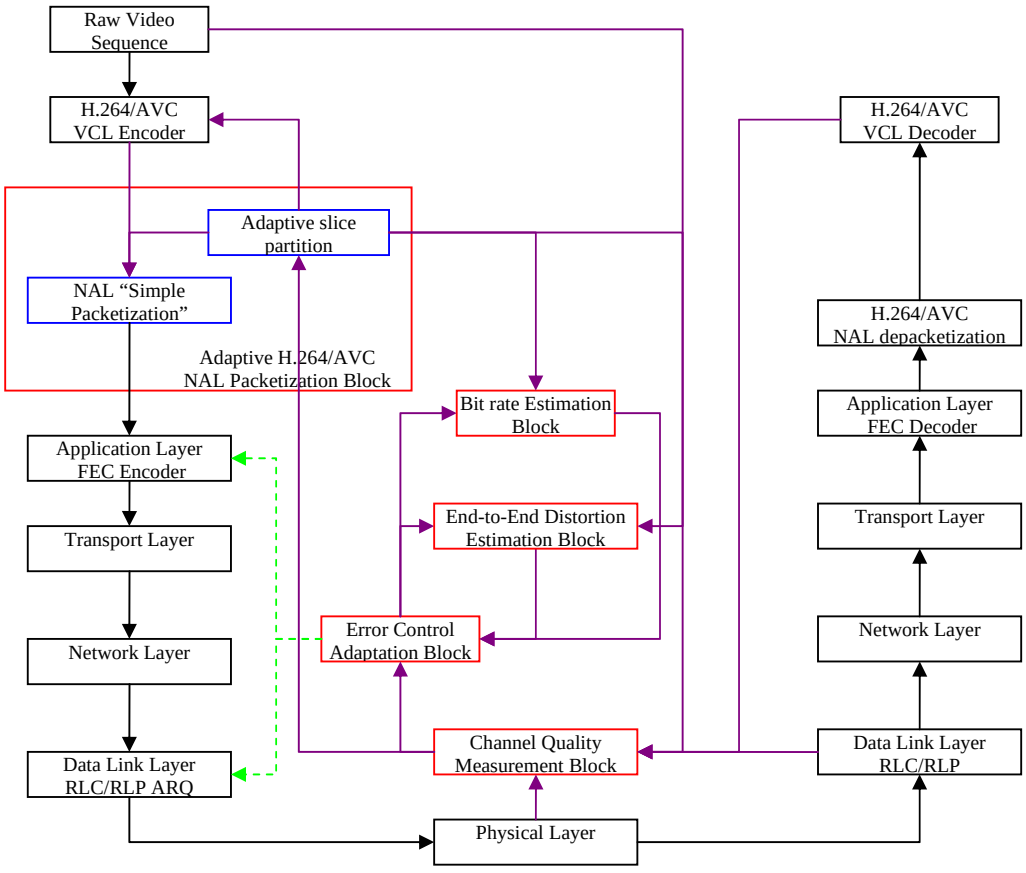


Figure 5.1: Channel adaptive H.264/AVC video transmission framework

Adaptive H.264/AVC NAL packetization block packetizes the slice-coded video stream into NALUs. NALUs are protected by packet level FEC at application layer, and passed to lower layers as RTP packets. RTP packets are further attached with UDP/IP headers at transport layer and network layer, and transmitted as blocks at data

link layer, where RLC/RLP retransmission is performed. channel quality measurement block will process the feedback from physical layer and data link layer and report BLER, BER, channel distortion, actual available bit rate to adaptive H.264/AVC NAL packetization block, error control adaptation block, and end-to-end distortion estimation block. Adaptive H.264/AVC NAL packetization block will pass the slice partition information to bit rate estimation block. Meanwhile, error control adaptation block will also pass the possible FEC and ARQ configuration to bit rate estimation block and end-to-end distortion estimation block. Bit rate estimation block will calculate the require bit rate and estimate system throughput to make sure such FEC and ARQ configuration will not let the bit rate exceed the allocated bandwidth. End-to-end distortion block will estimate the distortion based on such FEC and ARQ configuration and current channel condition for video frames which will be transmitted in next period of time. Finally, bit rate estimation block and end-to-end distortion estimation block will pass their results back to error control adaptation block. Error control adaptation block will record the estimated system throughput and distortion for the current set of FEC and ARQ, and determine the next possible set of FEC and ARQ configuration. Such process will go on iteratively until the optimal level of unequal FEC for I-frame and P-frame at application layer and the number of allowed RLC/RLP retransmission at data link layer are determined to either maximize the system throughput to available channel throughput or minimize end-to-end distortion.

5.3. Analysis of Channel Adaptive H.264/AVC Video Transmission Framework

Since the idea of adaptive H.264/AVC NAL packetization block has been discussed in Chapter 4, in this section, end-to-end distortion estimation, channel quality measurement, bit rate estimation, and error control adaptation will be discussed.

5.3.1. End-to-End Distortion Estimation

5.3.1.1. Statistical Analysis of Distortion

End-to-end distortion, denoted as D, objectively measures the video quality between original video sequence before video encoding and the reconstructed video sequence after video decoding. In an end-to-end wireless video coding and transmission system, there are two types of picture distortions, known as source distortion and channel distortion, denoted by D_s and D_c respectively. *Source distortion* occurs in video encoder, mainly caused by quantization error although DCT and motion estimation also contribute to source distortion. *Channel distortion* occurs when video packets are corrupted by channel errors. Note that D_s is the function of QP, and D_c is the function of number of slices per video frame, FEC, and ARQ. The total distortion is the summation of source and channel distortions, which is given by:

$$D = D_s(QP) + D_c(N_{slice}, FEC, ARQ). \tag{5.1}$$

For fixed QP, the source distortion can be decoupled from overall distortion. This thesis focuses on channel distortion only.

As indicated earlier, motion compensated predictive coding causes inter-frame propagation of channel errors. The complex error propagation in the video decoding loop has to be accurately modeled in channel distortion analysis. The modeling process needs to consider the specific source/channel encoding and decoding schemes, packetization method, patterns of the channel errors, error concealment, and so on. Several approaches for channel-distortion estimation have been proposed in the literature. To analyze the video transmission over lossy channels, a heuristic approach has been introduced in [71], where the channel distortion formula is derived through a leaking filter model. This distortion formula has several control parameters. To

estimate these parameters, one needs to run the codec over the video sequence for a few times to generate some measurement points and match the model to the experimental data. Obviously, this type of estimation scheme is not desired in real-time video coding and communication. A statistical simulation of the video decoder is employed in [72] to estimate the channel distortion with error concealment at the decoder. Using this decoder simulation, the encoder understands how much the video frame at decoder is "corrupted" by the random channel errors. Such estimation scheme involves potentially high computational complexity and implementation cost. In addition, this type of simulation approach does not allow further analysis for global optimization. In [73], a theoretical framework to estimate the channel distortion based on statistical analysis of the error propagation, error concealment, and channel decoding has been proposed. However, this framework treats I-frame and P-frame equally by assuming that intra refreshed MBs are employed in every video frame.

Extended from [73], now denote the loss probability of NALU in I-frame as $P_{NALU,I}$ and the loss probability of NALU in P-frame as $P_{NALU,P}$. Assume that each NALU contains the same number of MBs (or pixels), then the loss ratio of a pixel is also $P_{NALU,I}$ or $P_{NALU,P}$ [71]. Let $F(n,i)$ be the original value of pixel i in the n th video frame, and $\hat{F}(n,i)$ be the corresponding reconstructed value in the feedback loop at encoder for motion compensated predictive coding. At decoder side, the reconstructed value is denoted as $\tilde{F}(n,i)$. For inter coded MBs, let $e(n,i)$ be the motion-compensation difference at the encoder. Let $\hat{e}(n,i)$ and $\tilde{e}(n,i)$ be the corresponding reconstruction values at the encoder and decoder, respectively. Hence, the expected distortion for video frame n at receiver end is given by:

$$D(n) = E\left\{[F(n,i) - \tilde{F}(n,i)]^2\right\}. \tag{5.2}$$

Here $E\{x(n,i)\}$ represents the average (over all pixels) or expected value of random variable $x(n,i)$. According to definitions, the source distortion $D_s(n)$ and channel distortion $D_c(n)$ are given by:

$$D_s(n) = E\left\{[F(n,i) - \hat{F}(n,i)]^2\right\} \tag{5.3}$$

$$D_c(n) = E\left\{[\hat{F}(n,i) - \tilde{F}(n,i)]^2\right\}. \tag{5.4}$$

$D_s(n)$ and $D_c(n)$ are assumed to be uncorrelated [73], then the end-to-end distortion for video frame n is given by:

$$D(n) = D_s(n) + D_c(n). \tag{5.5}$$

At the decoder side, the following error concealment scheme is employed: if a MB is skipped by the decoder, both the motion vectors and the texture information are supposed to be lost since this thesis assumes that data partition is not adopted. Hence, the decoder simply copies the MB at the same location from the previous decoded frame. For a pixel in intra MBs, in case of no channel errors, its reconstruction value should be $\hat{F}(n,i)$ by definition. If the MB is lost, the reconstruction value is $\tilde{F}(n-1,i)$, which is copied from the previous decoded frame. Therefore, the expected channel distortion for pixels in I-frame is given by:

$$\begin{aligned}
D_c^I(n) &= E\left\{[\hat{F}(n,i) - \tilde{F}(n,i)]^2\right\} \\
&= P_{NALU,I} \cdot E\left\{[\hat{F}(n,i) - \tilde{F}(n-1,i)]^2\right\} + (1 - P_{NALU,I}) \cdot E\left\{[\hat{F}(n,i) - \hat{F}(n,i)]^2\right\} \\
&= P_{NALU,I} \cdot E\left\{[\hat{F}(n,i) - \tilde{F}(n-1,i)]^2\right\} \\
&= P_{NALU,I} \cdot E\left\{[\hat{F}(n,i) - \hat{F}(n-1,i) + \hat{F}(n-1,i) - \tilde{F}(n-1,i)]^2\right\} \\
&= P_{NALU,I} \cdot E\left\{[\hat{F}(n,i) - \hat{F}(n-1,i)]^2\right\} + P_{NALU,I} \cdot E\left\{[\hat{F}(n,i) - \tilde{F}(n-1,i)]^2\right\}
\end{aligned}$$

$$= P_{NALU,I} \cdot RFD(n, n-1) + P_{NALU,I} \cdot D_c(n-1) \tag{5.6}$$

where $RFD(n, n-1)$ is the reconstructed frame difference between frame n and $n-1$, and $D_c(n-1)$ is the channel distortion of previous I-frame or P-frame. The above relationship is hold based on the assumption that the frame difference is uncorrelated with channel distortion [73]. Note that the distortion estimator operates before quantization and coding of the current frame. At this point, $\hat{F}(n,i)$ is not available yet. Nevertheless, the encoder does know the absolute difference between consecutive frames. Let us define the original frame difference $FD(n, n-1)$ as:

$$FD(n, n-1) = E\left\{[F(n,i) - F(n-1,i)]^2\right\}. \tag{5.7}$$

Assume that if we regard the video encoder as low-pass filter [71], then the reconstruction frame is the filter output of the original frame. Note that a low-pass filter removes the energy in high frequency band. Hence, the reconstructed frame difference can be linearly proportional to original frame difference [73] as:

$$RFD(n, n-1) = \alpha \cdot FD(n, n-1) \tag{5.8}$$

where α is a constant known as the energy loss ratio of the encoder filter. It mainly depends how much information is discarded by the coding algorithm. More precisely, it is related to the average quantization step size. In this thesis, it is estimated using the statistics from previous frames. Therefore, equation (5.6) becomes:

$$D_c^I(n) = P_{NALU,I} \cdot D_c(n-1) + P_{NALU,I} \cdot \alpha \cdot FD(n, n-1). \tag{5.9}$$

For a pixel in inter MBs of P-frame, in case of no channel errors, its reconstruction value is given by definition:

$$\tilde{F}(n,i) = \tilde{F}(n-1, j) + \hat{e}(n,i). \tag{5.10}$$

where pixel j is the motion compensated prediction of pixel i. And at the encoder, with the motion compensated reference frame $\hat{F}(n-1,i)$, the reconstructed frame n is given by:

$$\hat{F}(n,i) = \hat{F}(n-1,\, j) + \hat{e}(n,i)\,. \tag{5.11}$$

If the MB is lost, the reconstruction value of pixel i is $\tilde{F}(n-1,i)$, which is copied from previous frame. Therefore, the expected channel distortion for P-frame is given by:

$$\begin{aligned} D_c^P(n) &= E\left\{[\hat{F}(n,i) - \tilde{F}(n,i)]^2\right\} \\ &= (1-P_{NALU,P})\cdot E\left\{[\hat{F}(n,i) - \hat{e}(n,i) - \tilde{F}(n-1,\, j)]^2\right\} \\ &\quad + P_{NALU,P}\cdot E\left\{[\hat{F}(n,i) - \tilde{F}(n-1,i)]^2\right\} \\ &= (1-P_{NALU,P})\cdot E\left\{[\hat{F}(n-1,\, j) - \tilde{F}(n-1,\, j)]^2\right\} \\ &\quad + P_{NALU,P}\cdot RFD(n,n-1) + P_{NALU,P}\cdot D_c(n-1)\,. \end{aligned} \tag{5.12}$$

If assume

$$\begin{aligned} E\left\{[\hat{F}(n-1,\, j) - \tilde{F}(n-1,\, j)]^2\right\} &= \beta\cdot E\left\{[\hat{F}(n-1,i) - \tilde{F}(n-1,i)]^2\right\} \\ &= \beta\cdot D_c(n-1) \end{aligned} \tag{5.13}$$

where β is constant describing motion randomness of video scene, then

$$D_c^P(n) = [(1-P_{NALU,P})\cdot\beta + P_{NALU,P}]\cdot D_c(n-1) + P_{NALU,P}\cdot\alpha\cdot FD(n,n-1)\,. \tag{5.14}$$

5.3.1.2. Fast Channel Distortion Estimation for I-frame and P-frame

In wireless video communication, with the feedback information on the channel condition, the encoder can determine the decoded picture quality of frame $(n-\Delta)$ and its previous frames, where Δ is the feedback delay in terms of frames counted at the

beginning of an estimation period. In other words, the channel distortion of $\{D_c(n-\Delta-m) \mid m \geq 0\}$ is available as seen by frame n at encoder. If denote T as maximum number of frames in one estimation period, and $1 \leq \Delta \leq T$, the coded frame over the estimation period has structure of $IPPP...P$, Figure 5.2 shows the frame structures for estimation periods.

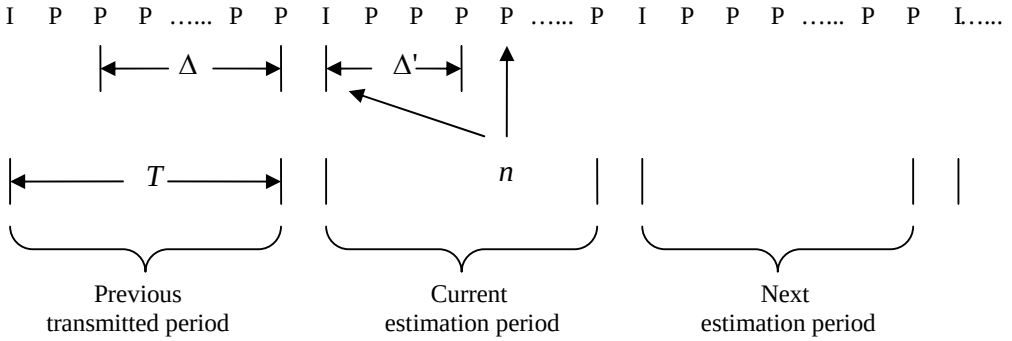


Figure 5.2: Frame structure for distortion estimation periods

If frame n is I-frame, that is the beginning of next estimation period, by iteration, equation (5.9) can be rewritten as:

$$
\begin{aligned}
D_c^I(n) &= P_{NALU,I} \cdot D_c(n-1) + P_{NALU,I} \cdot \alpha \cdot FD(n, n-1) \\
&= P_{NALU,I} \cdot \{[(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}] \cdot D_c(n-2) \\
&\quad + P_{NALU,P} \cdot \alpha \cdot FD(n-1, n-2)\} + P_{NALU,I} \cdot \alpha \cdot FD(n, n-1) \\
&= P_{NALU,I} \cdot [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}] \cdot D_c(n-2) \\
&\quad + P_{NaLU,I} \cdot P_{NALU,P} \cdot \alpha \cdot FD(n-1, n-2) + P_{NALU,P} \cdot \alpha \cdot FD(n, n-1) \\
&= P_{NALU,I} \cdot [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}] \cdot \{[(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}] \\
&\quad \cdot D_c(n-3) + P_{NALU,P} \cdot \alpha \cdot FD(n-2, n-3)\} \\
&\quad + P_{NALU,I} \cdot P_{NALU,P} \cdot \alpha \cdot FD(n-1, n-2) + P_{NALU,I} \cdot \alpha \cdot FD(n, n-1)
\end{aligned}
$$

$$= P_{NALU,I} \cdot [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^2 \cdot D_c(n-3) + P_{NALU,I} \cdot P_{NALU,P}$$

$$\cdot [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}] \cdot \alpha \cdot FD(n-2, n-3)$$

$$+ P_{NALU,I} \cdot P_{NALU,P} \cdot \alpha \cdot FD(n-1, n-2) + P_{NALU,I} \cdot \alpha \cdot FD(n, n-1)$$

$$= P_{NALU,I} \cdot [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta-1} \cdot D_c(n-\Delta)$$

$$+ P_{NALU,I} \cdot P_{NALU,P} \cdot \alpha \cdot \sum_{l=1}^{\Delta-1} \{[(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{l-1}$$

$$\cdot FD(n-l, n-l-1)\} + P_{NALU,I} \cdot \alpha \cdot FD(n, n-1) \quad (5.15)$$

where

$$D_c(n-\Delta) = \begin{cases} D_c^P(n-\Delta) & 1 \le \Delta < T \\ D_c^I(n-\Delta) & \Delta = T \end{cases}. \quad (5.16)$$

If let $\beta = 1$, then equation (5.15) can be simplified as:

$$D_c^I(n) = P_{NALU,I} \cdot D_c(n-\Delta) + P_{NALU,I} \cdot \alpha \cdot FD(n, n-1)$$

$$+ P_{NALU,I} \cdot P_{NALU,P} \cdot \alpha \cdot \sum_{l=1}^{\Delta-1} FD(n-l, n-l-1). \quad (5.17)$$

If frame n is P-frame, introduces Δ' as number of frames just before frame n within current estimation period, $1 \le \Delta' < T$, equation (5.14) can be rewritten as:

$$D_c^P(n) = [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}] \cdot D_c(n-1) + P_{NALU,P} \cdot \alpha \cdot FD(n, n-1)$$

$$= [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}] \cdot \{[(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}] \cdot D_c(n-2)$$

$$+ P_{NALU,P} \cdot \alpha \cdot FD(n-1, n-2)\} + p_{NALU,P} \cdot \alpha \cdot FD(n, n-1)$$

$$= [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^2 \cdot D_c(n-2) + P_{NALU,P} \cdot \alpha \cdot FD(n, n-1)$$

$$+[(1-P_{NALU,P})\cdot\beta+P_{NALU,P}]\cdot\alpha\cdot FD(n-1,n-2)$$

$$=P_{NALU,P}\cdot\alpha\cdot\sum_{l=0}^{\Delta'-1}[(1-P_{NALU,P})\cdot\beta+P_{NALU,P}]^{l}\cdot FD(n-l,n-l-1)$$

$$+[(1-P_{NALU,P})\cdot\beta+P_{NALU,P}]^{\Delta'}\cdot D_c(n-\Delta'). \quad (5.18)$$

Now frame $(n-\Delta')$ is I-frame, $D_c(n-\Delta')=D_c^I(n-\Delta')$, hence:

Case (A) if $\Delta=1$, then:

$$D_c^P(n)=P_{NALU,P}\cdot\alpha\cdot\sum_{l=0}^{\Delta'-1}[(1-P_{NALU,P})\cdot\beta+P_{NALU,P}]^{l}\cdot FD(n-l,n-l-1)$$

$$+[(1-P_{NALU,P})\cdot\beta+P_{NALU,P}]^{\Delta'}\cdot[P_{NALU,I}\cdot D_c^P(n-\Delta'-1)$$

$$+P_{NALU,I}\cdot\alpha\cdot FD(n-\Delta',n-\Delta'-1)]$$

$$=P_{NALU,P}\cdot\alpha\cdot\sum_{l=0}^{\Delta'-1}[(1-P_{NALU,P})\cdot\beta+P_{NALU,P}]^{l}\cdot FD(n-l,n-l-1)$$

$$+P_{NALU,I}\cdot\alpha\cdot[(1-P_{NALU,P})\cdot\beta+P_{NALU,P}]^{\Delta'}\cdot FD(n-\Delta',n-\Delta'-1)$$

$$+P_{NALU,I}\cdot[(1-P_{NALU,P})\cdot\beta+P_{NALU,P}]^{\Delta'}\cdot D_c^P(n-\Delta'-1). \quad (5.19)$$

Case (B) if $\Delta=2$, then:

$$D_c^P(n)=P_{NALU,P}\cdot\alpha\cdot\sum_{l=0}^{\Delta'-1}[(1-P_{NALU,P})\cdot\beta+P_{NALU,P}]^{l}\cdot FD(n-l,n-l-1)$$

$$+P_{NALU,I}\cdot\alpha\cdot[(1-P_{NALU,P})\cdot\beta+P_{NALU,P}]^{\Delta'}\cdot FD(n-\Delta',n-\Delta'-1)$$

$$+P_{NALU,I}\cdot[(1-P_{NALU,P})\cdot\beta+P_{NALU,P}]^{\Delta'}$$

$$\cdot\{P_{NALU,P}\cdot\alpha\cdot FD(n-\Delta'-1,n-\Delta'-2)$$

$$+[(1-P_{NALU,P})\cdot\beta+P_{NALU,P}]\cdot D_c^P(n-\Delta'-2)\}$$

$$= P_{NALU,P} \cdot \alpha \cdot \sum_{l=0}^{\Delta'-1} [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{l} \cdot FD(n-l, n-l-1)$$

$$+ P_{NALU,I} \cdot \alpha \cdot [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta'} \cdot FD(n-\Delta', n-\Delta'-1)$$

$$+ P_{NALU,I} \cdot P_{NALU,P} \cdot \alpha \cdot [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta'}$$

$$\cdot FD(n-\Delta'-1, n-\Delta'-2)$$

$$+ P_{NALU,I} \cdot [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta'+1} \cdot D_c^P(n-\Delta'-2). \quad (5.20)$$

Case (C) if $2 < \Delta \le T$, then:

$$D_c^P(n-\Delta'-2) = P_{NALU,P} \cdot \alpha \cdot \sum_{l=0}^{\Delta-3} \{[(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{l}$$

$$\cdot FD(n-\Delta'-2-l, n-\Delta'-3-l)\}$$

$$+ [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta-2} \cdot D_c(n-\Delta'-2-(\Delta-2))$$

$$= P_{NALU,P} \cdot \alpha \cdot \sum_{l=0}^{\Delta-3} \{[(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{l}$$

$$\cdot FD(n-\Delta'-2-l, n-\Delta'-3-l)\}$$

$$+ [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta-2} \cdot D_c(n-\Delta'-\Delta). \quad (5.21)$$

Hence, substitute equation (5.21) into equation (5.20),

$$D_c^P(n) = P_{NALU,P} \cdot \alpha \cdot \sum_{l=0}^{\Delta'-1} [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{l} \cdot FD(n-l, n-l-1)$$

$$+ P_{NALU,I} \cdot \alpha \cdot [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta'} \cdot FD(n-\Delta', n-\Delta'-1)$$

$$+ P_{NALU,I} \cdot P_{NALU,P} \cdot \alpha \cdot [(1-P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta'}$$

$$\cdot FD(n-\Delta'-1, n-\Delta'-2)$$

$$
\begin{aligned}
&+ P_{NALU,I} \cdot [(1 - P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta'+1} \\
&\cdot \{P_{NALU,P} \cdot \alpha \cdot \sum_{l=0}^{\Delta-3} \{[(1 - P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{l} \\
&\cdot FD(n - \Delta' - 2 - l, n - \Delta' - 3 - l) \\
&+ [(1 - P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta-2} \cdot D_c(n - \Delta' - \Delta)\} \\
&= P_{NALU,P} \cdot \alpha \cdot \sum_{l=0}^{\Delta'-1} [(1 - P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{l} \cdot FD(n - l, n - l - 1) \\
&+ P_{NALU,I} \cdot \alpha \cdot [(1 - P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta'} \cdot FD(n - \Delta', n - \Delta' - 1) \\
&+ P_{NALU,I} \cdot P_{NALU,P} \cdot \alpha \cdot [(1 - P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta'} \\
&\cdot FD(n - \Delta' - 1, n - \Delta' - 2) \\
&+ P_{NALU,I} \cdot P_{NALU,P} \cdot \alpha \cdot [(1 - P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta'+1} \\
&\cdot \sum_{l=0}^{\Delta-3} [(1 - P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{l} \cdot FD(n - \Delta' - 2 - l, n - \Delta' - 3 - l) \\
&+ P_{NALU,I} \cdot [(1 - P_{NALU,P}) \cdot \beta + P_{NALU,P}]^{\Delta'+\Delta-1} \cdot D_c(n - \Delta' - \Delta)
\end{aligned} \tag{5.22}
$$

where

$$
D_c(n - \Delta - \Delta') = \begin{cases} D_c^P(n - \Delta - \Delta') & 1 \le \Delta < T \\ D_c^I(n - \Delta - \Delta') & \Delta = T \end{cases}. \tag{5.23}
$$

Equation (5.22) can be considered as the combination of case (a) (b) and (c) for $1 \le \Delta \le T$. If let $\beta = 1$, then equation (5.22) can be simplified as:

$$
D_c^P(n) = P_{NALU,P} \cdot \alpha \cdot \sum_{l=0}^{\Delta'-1} FD(n - l, n - l - 1) + P_{NALU,I} \cdot \alpha \cdot FD(n - \Delta', n - \Delta' - 1)
$$

$$+ P_{NALU,I} \cdot P_{NALU,P} \cdot \alpha \cdot \cdot \sum_{l=1}^{\Delta-1} FD(n-\Delta'-l, n-\Delta'-1-l)$$

$$+ P_{NALU,I} \cdot D_c(n-\Delta'-\Delta). \tag{5.24}$$

5.3.1.3. Average Channel Distortion

The average channel distortion over current estimation period of T frames is given by:

$$\overline{D_c(T)} = \frac{1}{T}\left\{ D_c^I(n) + \sum_{f=n+1}^{n+T-1} D_c^P(f) \right\} \tag{5.25}$$

where $D_c^I(n)$ is given by equation (5.15), and $D_c^P(f)$ is given by equation (5.22) for $n+1 \le f \le n+T-1$.

5.3.2. Channel Quality Measurement

5.3.2.1. RLC Block Loss at Data Link Layer

The channel quality measurement block makes use of feedback information from data link layer and physical layer. Zorzi *et al.* [54] had investigated the accuracy of a first-order Markov process in modeling transmission on a correlated Rayleigh fading channel [53]. More precisely, as same as in Chapter 3, define p and $1-q$ as the probability that the j^{th} transport block transmission is successful, given that the $(j-1)^{\text{th}}$ transport block transmission was successful or unsuccessful, respectively, then having an analogy to equation (3.7) in Chapter 3, the steady-state BLER P_{BL} is given by:

$$P_{BL} = \frac{1-p}{2-p-q}. \tag{5.26}$$

[54] and [74] have shown that for a Rayleigh fading channel with fading margin F, P_{BL} can be expressed as:

$$P_{BL} = 1 - e^{-1/F}. \tag{5.27}$$

And q can be expressed as:

$$q = 1 - \frac{Q(\theta, \rho \cdot \theta) - Q(\rho \cdot \theta, \theta)}{e^{1/F} - 1} \tag{5.28}$$

where

$$\rho = J_0(2\pi f_d t_{TTI}) \tag{5.29}$$

$$\theta = \sqrt{\frac{2/F}{1-\rho^2}}. \tag{5.30}$$

ρ is the correlation coefficient of two successive samples spaced by transmission time interval t_{TTI} of the complex Gaussian fading channel, f_d is the Doppler frequency which is given by:

$$f_d = v / \lambda \tag{5.31}$$

where v is the mobile velocity and λ is the carrier wave length which is given by:

$$\lambda = c / f_0 \tag{5.32}$$

where c is the speed of light and f_0 is the carrier frequency of the W-CDMA system. $J_0(.)$ is the Bessel function of the first kind and zeroth order, and $Q(.,.)$ is the Marcum-Q function defined as:

$$Q(x, y) = \int_y^{\infty} e^{-\frac{(x^2+w^2)}{2}} \cdot I_0(x \cdot w) \cdot w \cdot dw \tag{5.33}$$

Substitute equation (5.28) into equation (5.26), p is given by:

$$p = \frac{1 - 2 \cdot P_{BL} + q \cdot P_{BL}}{1 - P_{BL}}. \tag{5.34}$$

5.3.2.2. UDP Packet Loss at Transport Layer

With data link layer parameter p and q, [75] proposed novel analysis of transport layer UDP packet loss with data link layer block retransmissions. Without loss generality, if NALU with RTP/UDP/IP headers has average L_{NALU} bytes and one block at data link layer has L_L bytes, then NALU with RTP/UDP/IP headers will be fragmented into N_L blocks. N_L is given by:

$$N_L = \lceil L_{NALU} / L_L \rceil. \tag{5.35}$$

Since this thesis consider NALU for I-frame and P-frame separately, the size of NALU and number of blocks are denoted as $L_{NALU,I}$, $N_{L,I}$, $L_{NALU,P}$, and $N_{L,P}$, respectively. The number of retransmissions allowed for a failed RLC/RLP block is $N_{\max}$. Now similar to [75], let us define:

- p_n: Prob{at least one out of n RLC/RLP blocks fails, given the first RLC/RLP block attempt succeeded};
- $q_n^{(k)}$: Prob{at least one out of n RLC/RLP blocks fails, given first RLC/RLP block already had $k \le N_{\max}$ retransmissions and current RLC/RLP block attempt failed}
- u_s: Prob{current UDP packet failed given that lass RLC/RLP block transmission of previous UDP packet succeeded}
- u_f: Prob{current UDP packet failed given that last RLC/RLP transmission of previous UDP packet failed}

The recursive relation of p_n and $q_n^{(k)}$ are as follows:

$$p_1 = 0 \tag{5.36}$$

$$q_n^{(N_{\max})} = 1 \qquad \text{for } 1 \le n \le N_L \tag{5.37}$$

$$q_1^{(k)} = q^{(N_{\max}-k)} \qquad \text{for } 0 \le k \le N_{\max} - 1 \tag{5.38}$$

$$p_n = p \cdot p_{n-1} + (1-p) \cdot q_{n-1}^{(0)} \qquad \text{for } 2 \le n \le N_L \text{ and } 0 \le k \le N_{\max} - 1 \tag{5.39}$$

$$q_n^{(k)} = (1-q) \cdot p_n + q \cdot q_n^{(k+1)} \qquad \text{for } 2 \le n \le N_L \text{ and } 0 \le k \le N_{\max} - 1. \tag{5.40}$$

Hence, u_s and u_f are given by:

$$u_s = p \cdot p_{N_L} + (1-p) \cdot q_{N_L}^{(0)} \tag{5.41}$$

$$u_f = (1-q) \cdot p_{N_L} + q \cdot q_{N_L}^{(0)}. \tag{5.42}$$

Continue to define:

- $v_{fs}^{(n)}$: Prob{the last RLC/RLP block transmission of current UDP packet of length n RLC/RLP blocks succeed given that current UDP packet failed}
- $v_{ss}^{(n)}$: Prob{the last transmission of current UDP packet of length n RLC/RLP blocks succeed given that current UDP packet succeeded}

Clearly, the following relations hold:

$$v_{ss}^{(n)} = 1 \tag{5.43}$$

$$v_{fs}^{(1)} = 0 \tag{5.44}$$

$$v_{fs}^{(i)} = v_{fs}^{(i-1)} \cdot [p + (1-p) \cdot \sum_{j=0}^{N_{\max}-1} q^j \cdot (1-q)] + (1 - v_{fs}^{(i-1)}) \cdot \sum_{j=0}^{N_{\max}} q^j \cdot (1-q)$$

$$= v_{fs}^{(i-1)} \cdot (p+q-1) \cdot q^{N_{\max}} + (1-q^{N_{\max}+1}) . \quad (5.45)$$

Let $P_{U,ss}$ and $P_{U,fs}$ be the probabilities that the current UDP packet succeed given that previous UDP packet succeeded or failed, respectively. Hence, following relations hold:

$$P_{U,ss} = (1-u_s) \cdot v_{ss}^{(N_L)} + (1-u_f) \cdot (1-v_{ss}^{(N_L)}) \quad (5.46)$$

$$P_{U,fs} = (1-u_s) \cdot v_{fs}^{(N_L)} + (1-u_f) \cdot (1-v_{fs}^{(N_L)}) \quad (5.47)$$

$$P_{U,sf} = 1 - P_{U,ss} \quad (5.48)$$

$$P_{U,ff} = 1 - P_{U,fs} . \quad (5.49)$$

Finally, the steady state UDP packet loss probability is given by:

$$P_U = \frac{1-P_{U,ff}}{2-P_{U,ss}-P_{U,ff}} = \frac{P_{U,fs}}{1+P_{U,fs}-P_{U,ss}} . \quad (5.50)$$

5.3.3. Bit rate Estimation

5.3.3.1. Average NALU Size Modeling

To estimate the bit rate required for the transmission in current estimation period, the bit rate estimator needs to know the NALU size. However, tracking the exact NALU size before encoding is impossible for different sequences. Instead, average NALU size is used for bit rate estimation based on following assumptions:

- Statistically, only a rough bit rate estimation is needed;
- Each slice contains roughly equal number of MBs;
- H.264/AVC source coding generates source bits at very low bit rate by proper setting the QP;

- Slices in I-frame and P-frame are considered separately.

Denote $\overline{S_{NALU,I}}$ and $\overline{S_{NALU,P}}$ as average NALU size for I-frame and P-frame respectively, and $H_{RTP/UDP/IP}$ as overhead size of RTP/UDP/IP headers. Clearly,

$$L_{NALU,I} = H_{RTP/UDP/IP} + \overline{S_{NALU,I}} \tag{5.51}$$

$$L_{NALU,P} = H_{RTP/UDP/IP} + \overline{S_{NALU,P}}\,. \tag{5.52}$$

The source bit rate can be controlled by QPs [29]. And slice coding breaks the compression efficiency across slice boundaries and adds overheads known as slice header [9,11]. This thesis proposes a fast modeling of average NALU size as a linear function of number of slices for different QPs. This modeling is based on the observations of exhaustive experiments over different video sequences as follows.

Figure 5.3 shows the average number of bits per frame for Foreman (400 frames, $QP = 36$, $\overline{PSNR\_Y} = 30.88dB$), Carphone (382 frames, $QP = 36$, $\overline{PSNR\_Y} = 31.88dB$), Suzie (150 frames, $QP = 38$, $\overline{PSNR\_Y} = 31.90dB$) and Claire (494 frames, $QP = 42$, $\overline{PSNR\_Y} = 30.83dB$) video sequences. Figure 5.4 shows the average number of bits per slice in I-frame and Figure 5.5 shows the average number of bits per slice in P-frame for the different video sequences. Appendix A shows the statistical data collection in terms of source bit rate, the number of bits per I-frame and P-frame, and the number of bits per slice in I-frame and P-frame for above 4 video sequences individually at different QPs. Conclusion can be drawn that as the number of slices per frame increases, the number of bits per frame increases linearly.

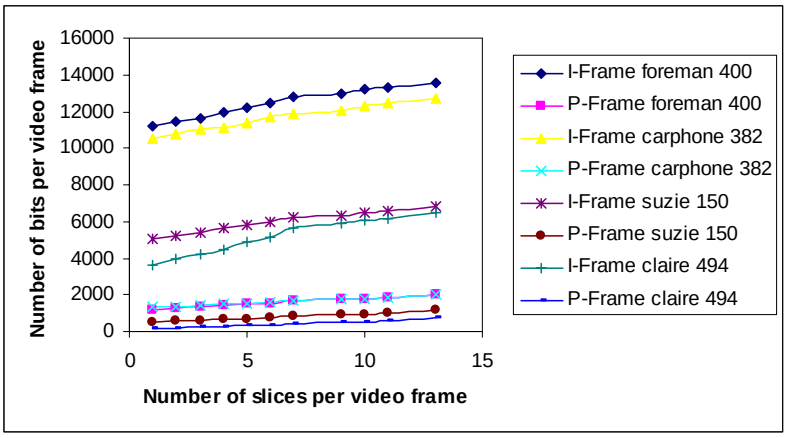


Figure 5.3: Average number of bits per video frame

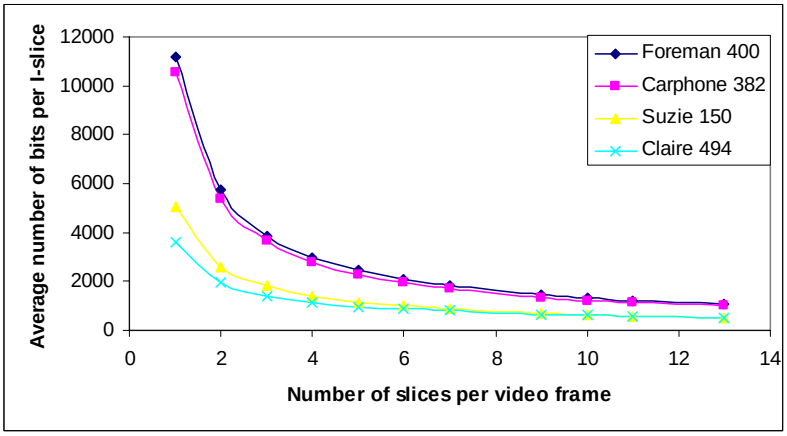


Figure 5.4: Average number of bits per I-slice

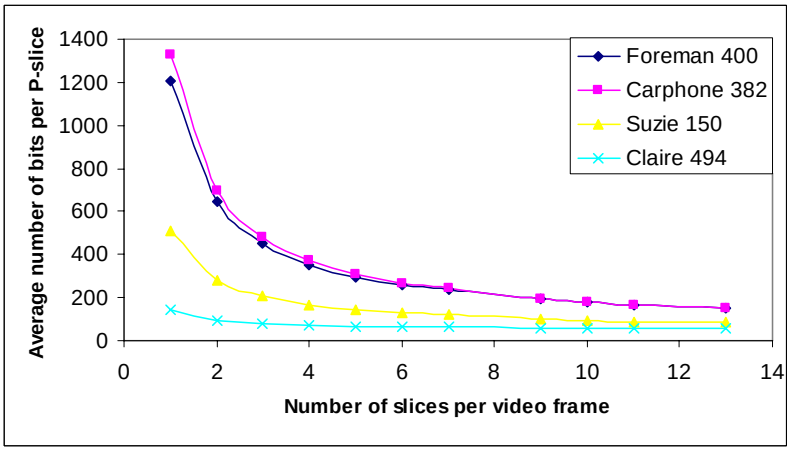


Figure 5.5: Average number of bits per P-slice

Hence if denote N_{slices} as number of slices per video frame, $a_I(QP)$, $b_I(QP)$ and $a_P(QP)$, $b_P(QP)$ as linear function parameters of QP, then the average number of bits per frame $\overline{S_{frame,I}}$ and $\overline{S_{frame,P}}$ can be expressed as follows:

$$\overline{S_{frame,I}} = a_I(QP) \cdot N_{slices} + b_I(QP) \tag{5.53}$$

$$\overline{S_{frame,P}} = a_P(QP) \cdot N_{slices} + b_P(QP). \tag{5.54}$$

In addition, Figure 5.4 and Figure 5.5 show that as number of slices increase, the number of bits per slice will converge. This observation could be verified as follows:

$$\overline{S_{NALU,I}} = \overline{S_{frame,I}} \Big/ N_{slice} = a_I(QP) + b_I(QP) \Big/ N_{slice} \tag{5.55}$$

$$\overline{S_{NALU,P}} = \overline{S_{frame,P}} \Big/ N_{slice} = a_P(QP) + b_P(QP) \Big/ N_{slice}. \tag{5.56}$$

Mathematically, as $N_{slices} \to \infty$, $\overline{S_{NALU,I}} \to a_I(QP)$ and $\overline{S_{NALU,P}} \to a_P(QP)$. For each video sequence, a look up table has been implemented at encoder to facilitate the calculation of above model parameters. Assuming that encoder knows the number of bits per frame in “one slice-one frame” case and “11 slices-one frame” case at different QPs, then the linear model is set up in the run time, and the value of model parameters $a_I(QP)$, $b_I(QP)$ and $a_P(QP)$, $b_P(QP)$ for different QPs in each video sequence is obtained by graph interpolation, and furthermore, the average size of NALU is obtained for I-frame and P-frame accordingly.

5.3.3.2. Bit rate Estimation

UEP is adopted in this thesis to differentiate the importance of NALU from I-frame or P-frame. Assumes that $RS(n_I, k_I)$ is used to protect NALU in I-frame, and $RS(n_P, k_P)$ is used to protect NALU in P-frame. The packet loss probability of NALU

with RTP/UDP/IP headers in I-frame and P-frames are denoted as $P_{U,I}$ and $P_{U,P}$ respectively. Hence the NALU loss probabilities $P_{NALU,I}$ and $P_{NALU,P}$ are given by equation (4.5) in Chapter 4:

$$P_{NALU,I} = \sum_{j=n_I-k_I+1}^{n} \frac{j}{n_I} \cdot \binom{n_I}{j} \cdot P_{U,I}^{j} \cdot (1-P_{U,I})^{n_I-j} \tag{5.57}$$

$$P_{NALU,P} = \sum_{j=n_P-k_P+1}^{n} \frac{j}{n_P} \cdot \binom{n_P}{j} \cdot P_{U,P}^{j} \cdot (1-P_{U,P})^{n_P-j}. \tag{5.58}$$

Now come back to the data link layer, as in Chapter 3, the probability of successful transmission of a RLC/RLP block after j times retransmission is given by:

$$P_{BL,succ}(j) = P_{BL}{}^{j}(1-P_{BL}) \qquad 0 \le j \le N_{\max}. \tag{5.59}$$

Then probability of successful transmission within $N_{\max}$ retransmissions is given by:

$$P_{BL,succ} = \sum_{j=0}^{N_{\max}} P_{BL,succ}(j) = 1 - P_{BL}^{N_{\max}+1}. \tag{5.60}$$

Define D_R as retransmission delay factor. This thesis assumes that selective ARQ is employed and no other factors will cause delays rather than transmission time, then $D_R = 1$. After a radio link block has been declared as corrupt at the receiver, the retransmission request arrives at the transmitter after one additional t_{TTI}. Hence, the minimum time between two successive retransmissions of the same block is exactly the *Round Trip Time* (RTT) of $2 \cdot t_{TTI}$. To extend further, the minimum time between j successive retransmissions of the same block is exactly $j \cdot t_{TTI}$. Therefore, the expected duration for successful transmission of a RLC/RLP block is given by:

$$\overline{t_{BL,succ}} = E[t_{BL,succ}] = \sum_{j=0}^{N_{\max}} \frac{P_{BL,succ}(j)}{P_{BL,succ}} \cdot (j \cdot D_R \cdot t_{TTI} \cdot + t_{TTI}). \tag{5.61}$$

And the expected duration for failure transmission for a RLC/RLP block is given by:

$$\overline{t_{BL,\,fail}} = E[t_{BL,succ}] = (N_{\max} \cdot D_R + 1) \cdot t_{TTI} \,. \tag{5.62}$$

Therefore, the expected transmission duration for a RLC/RLP block is given by:

$$\overline{t_{BL}} = E[t_{BL}] = \overline{t_{BL,succ}} \cdot P_{BL,succ} + \overline{t_{BL,\,fail}} \cdot (1 - P_{BL,succ}) \,. \tag{5.63}$$

For a NALU with RTP/UDP/IP headers in I-frame having average packet length $L_{NALU,I}$ and $N_{L,I}$ RLC/RLP blocks, the expected transmission duration is given by:

$$\overline{t_{NALU,I}} = E[t_{NALU,I}] = \overline{t_{BL}} \cdot N_{L,I} \,. \tag{5.64}$$

And for a NALU with RTP/UDP/IP headers in P-frame having average packet length $L_{NALU,P}$ and $N_{L,P}$ RLC/RLP blocks, the expected transmission duration is given by:

$$\overline{t_{NALU,P}} = E[t_{NALU,P}] = \overline{t_{BL}} \cdot N_{L,P} \,. \tag{5.65}$$

Then, the total transmission duration over an estimation period is given by:

$$\bar{t} = E[t] = (\overline{t_{NALU,I}} \cdot n_I \cdot g_I + \overline{t_{NALU,P}} \cdot n_P \cdot g_P) \cdot Frame\_Rate / T \tag{5.66}$$

where T is maximum number of video frames in one estimation period, g_I and g_P are the number of groups of $RS(n_I, k_I)$ and $RS(n_P, k_P)$ within an estimation period. For MDS codes, $g_I = 1$ and $g_P = 1$. Finally, the estimated bit rate $\overline{B}$ is given by:

$$\overline{B} = \bar{t} \cdot BW \tag{5.67}$$

where BW is the bandwidth dedicated to video transmission services.

The normalized throughput of the video transmission system is defined as the ratio of useful video data, which are received to the total amount of transmitted bits. It is given by:

$$\eta_T = \frac{[\overline{S_{NALU,I}} \cdot k_I \cdot g_I \cdot (1 - P_{NALU,I}) + \overline{S_{NALU,P}} \cdot k_P \cdot g_P \cdot (1 - P_{NALU,P})] \cdot 8}{(\overline{t_{NALU,I}} \cdot n_I \cdot g_I + \overline{t_{NALU,P}} \cdot n_P \cdot g_P) \cdot BW}. \quad (5.68)$$

5.3.4. Error Control Adaptation

The error control adaptation block provides error control configurations of $\{N_{\max}, RS(n_I, k_I), RS(n_P, k_P)\}$ together with adaptive H.264/AVC NAL packetization scheme to enhance the error resilience of the transmission in wireless environment. After receiving the channel information from channel estimator, error control adaptation selects possible $\{N_{\max}, RS(n_I, k_I), RS(n_P, k_P)\}$ configuration, and passes it to end-to-end distortion estimator and bit rate estimator. The end-to-end distortion estimator will feedback the estimated average channel distortion based on the passed error control configuration and channel information from channel estimator. And similarly, bit rate estimator will feedback the estimated bit rate and system normalized throughput. With above feedback information, the error adaptation block will determine the optimal solution to either maximize the throughput to available channel throughput at current channel state or minimize the end-to-end distortion.

As stated above, maximizing system throughput to available channel throughput and minimizing the end-to-end distortion are the two evaluation metrics for the error control adaptation. If maximize system throughput to available channel throughput is the evaluation metric, equation (5.68) is the cost function, which is maximized and is subject to the available throughput at current channel state and dedicated channel bandwidth constraints as follows:

$$\eta = \arg\max_{\{N_{\max}, RS(n_I, k_I), RS(n_P, k_P)\}} \{\eta_T\}$$

$$\text{s.t} \qquad \eta_T \le \eta_{T_C} \cdot \gamma \text{ and } \overline{B} \le BW \quad (5.69)$$

where η_{T_C} is the available throughput at current channel state and γ is the constant such that the throughput optimization can be defined as either Aggressive ($\gamma > 1$), Neutral ($\gamma = 1$), or Conservative ($\gamma < 1$) among channel state transitions according to whether channel has been improved by comparing previous channel state and current channel state. Table 5.1 shows the optimization algorithm settings according to transition of channel states. Within each state transition, whether the setting is aggressive, neutral or aggressive depends on the improvement or degradation level of BLER, which is similar to the adaptive slice partition in Chapter 4.

Table 5.1: Throughput optimization settings according to transition of channel states

Previous State \ Current State	Amiable State	Noisy State	Hostile State
Amiable State	Aggressive Neutral	Neutral Conservative	Conservative
Noisy State	Aggressive Neutral	Neutral	Neutral Conservative
Hostile State	Aggressive	Aggressive Neutral	Neutral

If minimizing end-to-end distortion is the evaluation metric, equation (5.25) is the cost function which is minimized and is subject to the dedicated channel bandwidth as follows:

$$D_c = \underset{\{N_{\max}, RS(n_I,k_I), RS(n_P,k_P)\}}{\arg\min} \{\overline{D_c(T)}\}$$

$$\text{s.t} \qquad \overline{B} \le BW \,. \tag{5.70}$$

To solve equation (5.69) and equation (5.70), the optimal solution can be found by using *Dynamic Programming* (DP) approach. This thesis targets at the low bit rate

H.264/AVC video transmission, due to channel bandwidth constraint, the $\{N_{\max}, RS(n_I, k_I), RS(n_P, k_P)\}$ set has been restricted to a limit set if only finite number of RLC/RLP retransmission is allowed, and RS code rate is determined by an integer number of source and redundant packets. Hence, the DP can be viewed as a shortest path problem in a trellis, where each stage corresponds to the mode selection.

5.4. Summary

This chapter proposes channel adaptive H.264/AVC video transmission framework under cross layer optimization. This framework defines five critical channel adaptive blocks: i) end-to-end distortion estimation block is to estimate the end-user quality for group of video frames which will be transmitted in the next period of time based on current channel condition; ii) Loss Estimation block is to estimate the channel information in terms of BLER, BER, channel distortion, and throughput etc., from physical layer and data link layer for a particular wireless mobile network infrastructure; iii) Bit rate estimation block is to estimate the bit rate as function of average NALU size and system configurations for bandwidth constraint checking. In this thesis, average NALU size has been modeled as a linear function to provide fast solution; iv) Error control adaptation block combines information from other channel adaptive blocks and finds the optimal solution to the level of UEP for I-frame and P-frame NALUs, and the maximum number of allowed RLC/RLP retransmission by using either throughput or distortion as evaluation metric; v) Adaptive H.264/AVC NAL packetization block has been discussed in Chapter 4. While in this chapter, we propose the channel adaptive techniques for H.264/AVC video transmission framework under cross layer optimization, we will, in the next chapter, show the performance of proposed channel adaptive techniques in details.

Chapter 6

Performances of Channel Adaptive H.264/AVC Video Transmission Framework

The channel adaptive H.264/AVC video transmission framework defines the interactions of various channel adaptation blocks by processing the feedback information from various layers, and adjusting error control mechanisms through whole protocol stack to enhance the error resilience of video transmission. Meanwhile, it aims to facilitate throughput adaptation in time-varying wireless environment so that the network or system efficiency can be improved in addition to the traditional approach of minimizing end-to-end distortion. The simulation results over high and low error channel conditions, as expected, demonstrate the benefits of the proposed framework through cross layer design in improving system throughput compared to the video transmission system with fixed configuration [25-27] in terms of fixed NAL packetization and fixed error control configuration. And the proposed framework also shows better end-user quality compared to the system with fixed NAL packetization under channel adaptive error control configuration.

6.1. Simulation Environment

6.1.1. Common Test Conditions for Wireless Video

In H.264/AVC standardization process [9], the importance of mobile video transmission has been recognized by adopting the appropriate common test conditions

for 3G mobile networks. These test conditions permit the selection of appropriate coding features, as well as testing and evaluating error resilience features. In this thesis, the IP-based test conditions are the main concerns. JVT defines the common test conditions [65] with 6 radio channel conditions for packet-switched conversational services as well as streaming services over 3G mobile networks. The 6 radio channel conditions are simulated as bit-error patterns, which were captured in different real or emulated mobile radio channels. The bit-error patterns are captured above the physical layer and below the RLC/RLP layer. Hence, they are used as the physical layer simulation in practice. The properties such as bit-rate, length, BER, and the mobile speed of the bit-error patterns are presented in Table 6.1, as reported in [65].

Table 6.1: Wireless H.264/AVC video transmission bit error patterns

Pattern No.	Physical Layer Bit rate	Error Pattern Length in seconds	BER	RLC PDU Size	Mobile Speed	Application
1	$64kbps$	60s	9.3×10^{-3}	640 bits	$3km/h$	Streaming
2	$64kbps$	60s	2.9×10^{-3}	640 bits	$3km/h$	Streaming
3	$64kbps$	180s	5.1×10^{-4}	640 bits	$3km/h$	Conversational
4	$64kbps$	180s	1.7×10^{-4}	640 bits	$50km/h$	Conversational
5	$128kbps$	180s	5.0×10^{-4}	640 bits	$3km/h$	Conversational
6	$128kbps$	180s	2.0×10^{-4}	640 bits	$50km/h$	Conversational

The bit-errors in the files are statistically dependent, as channel coding and decoding included in 3G systems produces burst errors [4,7,65]. This has been taken into account by evaluating the bit-error pattern files in the following. Patterns 1 and 2 are mostly suited to be used in the simulation of video streaming applications, where RLC/RLP re-transmissions at data link layer can correct many of the block/frame losses. Turbo coding has been applied as channel coding, and power control targeting throughput maximization rather than error minimization. Patterns 3 to 6 are meant to simulate a more reliable, lower error-rate bearer that is required in conversational

applications. [65] defines that only streaming service allows RLC/RLP retransmission at data link layer, conversational service does not allow RLC/RLP retransmission due to strict delay constraints. Nevertheless, since the BERs in patterns 3 to 6 are so low that even without RLC/RLP retransmission, the application layer FEC with error concealment can combat those errors. However, patterns 1 and 2 are unrealistic for conversational service, as an acceptable end-user quality cannot be achieved with such high error rates without retransmissions [7]. In order to demonstrate the advantages of the proposed channel adaptive H.264/AVC video transmission framework, error control mechanisms through different layers should be considered with respect to the overall transmission framework. Therefore, this thesis uses video streaming service in the simulation. In other words, simulations are carried on error patterns 1 and 2 only for worst case performance. Error pattern 1 is referred as high-error channel while error pattern 2 is referred as low-error channel.

6.1.2. Overview of Simulation Testbed

Both video transmitter and receiver are assumed to reside in a private operator's network, which consists of a fixed IP-based core network and a radio access network. A video streaming server is directly connected to the operator's core network or both video terminations are inside the mobile network. The network operator's core network is assumed to be over-provisioned so that the packet loss rate in the core network is negligible and the network resources bottleneck is at the radio interface. Thus any degradation to the video stream results from fading/shadowing errors and background white noise induced in wireless channel. In this thesis, an end-to-end simulation testbed is implemented for channel adaptive H.264/AVC video transmission framework and video transmission system with fixed configurations over 3GPP/3GPP2 air interface. This simulation testbed supports both 3GPP-based W-CDMA and

3GPP2-based CDMA-2000. As W-CDMA and CDMA-2000 are similar in the sense of user plane protocols [65] but with slightly differences in the terminologies at data link layer, W-CDMA system is employed in the simulations since those differences will not affect the final performance. In the simulation testbed, the proposed channel adaptive H.264/AVC video transmission framework is realized according to Figure 5.1 in Chapter 5. The 5 critical channel adaptive blocks are implemented according to the analysis in Chapter 4 and Chapter 5 to facilitate cross layer optimization.

More specifically, assume that the channel adaptive H.264/AVC video transmission framework already works out the optimal error control solution for current period of transmission. Before the application layer, the latest H.264/AVC test model software version JM9.3 [66] is modified by adding the proposed novel adaptive H.264/AVC NAL packetization scheme. The modified video codec is configured using the baseline profile according to [76] such that:

1) Frame structure is *IPPP...IPPP...*;
2) MV search range is ±32;
3) MV resolution is ¼ pixels;
4) Hadamard transform is used;
5) Optimization is enabled;
6) Number of reference frames is 5;
7) UVLC is enabled;

The additional simulation parameters are as follows:

8) Testing sequence is Foreman, QCIF, 400 frames, transmitted at constant frame rate 10fps, estimation period is set to be 5 frames;

9) The minimum number of slices per video frame is 3, the maximum number of slices per video frame is 11;

10) Quantization parameter $QP = 36$ for both I-frame and P-frame, which results $\overline{PSNR} = 32dB$ of luminance and chrominance components;

11) Error corrupt NALUs will be passed to the decoder with error concealment enabled.

The selection of $QP = 36$ is subject to physical layer bandwidth constraints of $64kbps$ as shown in Table 6.1. Under such QP, source bit rate at 11 slices per video frame can be supported, and the final channel bit rate with error control redundancy will not exceed the bandwidth limit. If smaller quantization parameter is used, then the maximum number of slices per video frame supported will be less than 11 subject to bandwidth constraint. In addition, although the QCIF-sized video is employed here, video with higher resolution can also be supported if larger bandwidth is available.

At application layer, NALUs for I-frame and P-frame are protected by packet level $RS(n,k)$ codes under UEP scheme [26-27] since I-frame serving as reference frame is more important than P-frame and heavier channel protection should be assigned earlier rather than later [30]. NALUs from I-frame will be protected with code rate within [0.6,1] range, while NALUs from P-frame will be protected with code rate within [0.8,1] range. Additionally, the scheme of MDS code is employed to maximize the error correction capability of $RS(n,k)$ codes. Figure 6.1 shows the implementation of packet level $RS(n,k)$ codes. Bit stuffing is employed for variable length NALUs to make sure that they have roughly equal length.

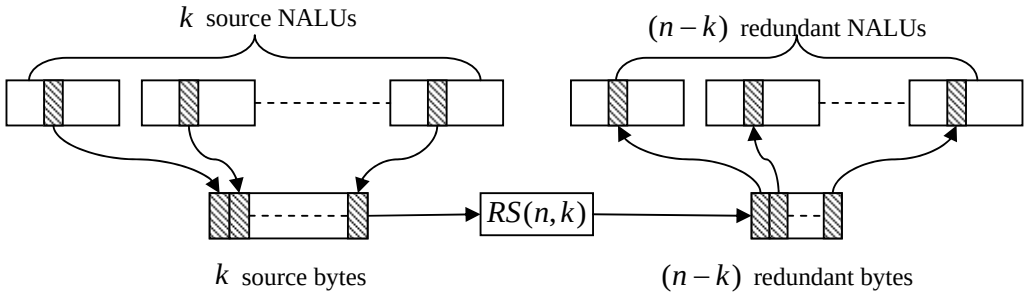


Figure 6.1: $RS(n,k)$ **code implemented for NALUs**

After RS encoding, RTP header is attached to each NALU to form an RTP packet. At transport layer and network layer, UDP/IP headers are attached to each RTP packet. Then the NALUs with RTP/UDP/IP headers are passed to data link layer. The simulation testbed will perform following simulations to model data link layer and physical layer behaviors by assuming that the high-level syntax parameters (eg, sequence and picture parameter sets) have been transmitted in advance and out-of-band:

1) Framing at data link layer, performing RoHC [48] as suggested in [5];
2) Schedule and split NALU with RTP/UDP/IP compressed header to RLC/RLP transport blocks/frames of 80 bytes. Add RLC/RLP frame header (2 or 4 bytes);
3) One link layer transport block/frame is packed into one physical layer l*ogical transmission unit* (LTU). This means that each link layer transport block/frame can have a *frame quality indicator* (FQI) that indicates if there are errors detected by the physical layer CRC. Map bit-error pattern to RLC/RLP transport blocks/frames, discard RLC/RLP blocks/frame if it contains errors;
4) Optional RLC/RLP retransmission, the maximum number of retransmissions is set by optimal solution of framework and is upper bounded by 3 [60];
5) If an RLC/RLP frame is lost, the NALU with RTP/UDP/IP headers that were covered by it (even if only partially) are declared as corrupt;

6) Physical layer and data link layer statistics, such as BER, BLER, actual available bit rate, etc are fed back to the respective channel adaptation blocks;

7) The received NALUs with RTP/UDP/IP headers no matter corrupt or not are passed to upper layer for de-packetization and decoding;

8) Finally, the channel adaptive blocks will work out the optimal number of slices per video frame for NAL packetization and error control solution for the next transmission period.

6.1.3. Evaluation Criteria

As stated before, the new directions in the design of wireless systems do not necessarily attempt to minimize the error rate but to maximize the throughput. The most important motivation of proposing the novel adaptive H.264/AVC NAL packetization scheme with channel adaptive H.264/AVC video transmission framework is to facilitate throughput adaptation in time-varying wireless environment so that the network or system efficiency can be improved. Hence, with similar end-user quality, normalized throughput defined in equation (5.68) in Chapter 5 is the performance evaluation criteria. Moreover, as end-user quality is usually the direct indication of system performance, the framework also includes average PSNR as another evaluation criterion. In [76], the average PSNR of luminance component (Y) and chrominance components (U, V) is computed as follows:

$$\overline{PSNR} = 10\log_{10}\left(255^2 \Big/ \overline{MSE}\right) \tag{6.1}$$

where the $\overline{MSE}$ is given by:

$$\overline{MSE} = \frac{4 \times MSE_Y + MSE_U + MSE_V}{6}. \tag{6.2}$$

6.2. Performances of Channel Adaptive H.264/AVC Video Transmission Framework

6.2.1. Performances under Throughput Metric

The proposed channel adaptive H.264/AVC video transmission framework using throughput metric as cost function in the cross layer optimization aims to adapt system throughput to variation of channel capacity with acceptable end-user quality. Figure 6.2 shows the PSNR performance, and Figure 6.3 shows the throughput performance of proposed framework using throughput metric as cost function in high-error channel. Note that the discontinuity of discrete BLER curve indicates that the channel is error free at that moment, which cannot be plotted in logarithmic scale.

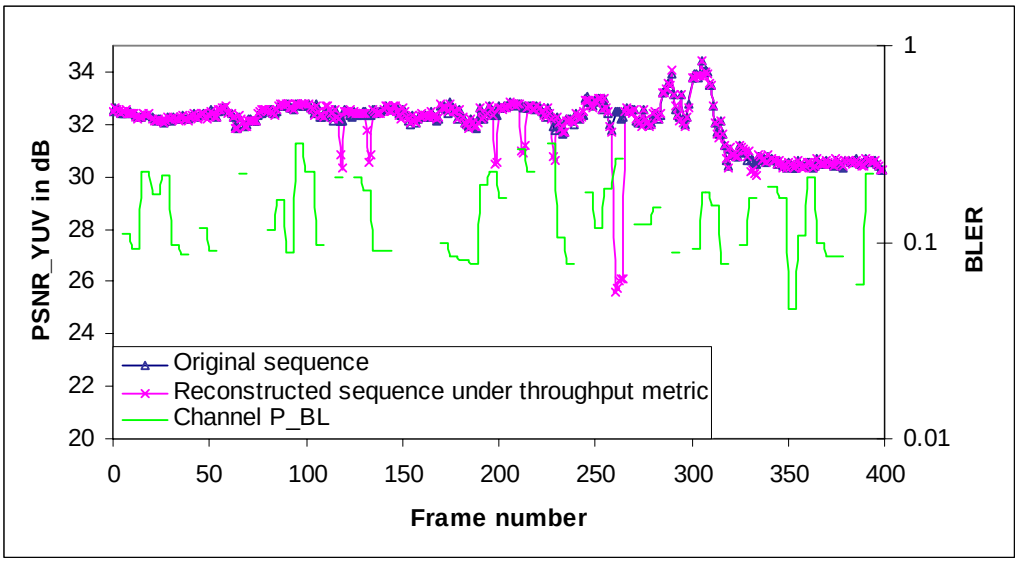


Figure 6.2: PSNR performance of proposed framework using throughput metric as cost function in high-error channel

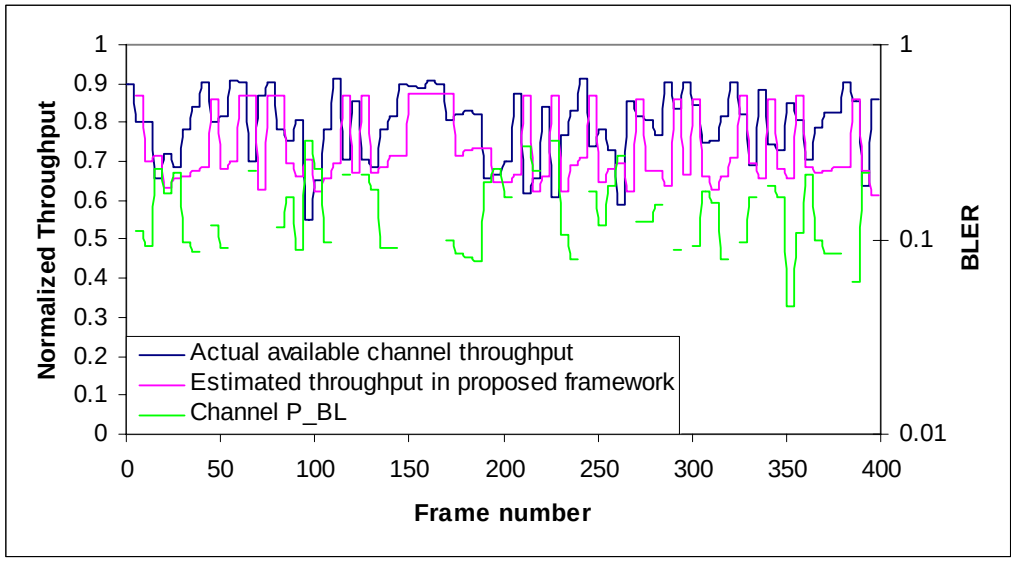


Figure 6.3: Throughput performance of proposed framework using throughput metric as cost function in high-error channel

PSNR performance shows that the end-user quality is guaranteed in the proposed framework using throughput metric in high-error channel. Although PSNR drops a little bit when channel is very hostile, the end-user quality is still acceptable both objectively (above 22dB) and subjectively due to the reservation of channel protection by conservative approach defined in throughput metric in Chapter 5. Indeed, it is the trade-off between system efficiency and quality distortion. Since adapting throughput to the variations of channel capacity is the objective, minor quality loss is definitely unavoidable under such high error rate channel condition.

The throughput performance shows that in such high error rate channel conditions, the proposed framework using throughput metric is able to adapt system throughput to the variations of channel capacity. There are discrepancies between the estimated channel throughput and actual available channel throughput, which are due to following reasons. Firstly, the throughput adaptation aims to track the variations of channel available throughput rather than the exact value because accurate estimation is

not always possible due to the fact that the channel quality measurement is always behind channel variations. The proposed framework intends to improve system throughput whenever the channel is not hostile and slow varying. Secondly, the channel adaptation solution $\{N_{slice}, N_{\max}, RS(n_I, k_I), RS(n_P, k_P)\}$ is in discrete value format. The calculations with discrete values lead to discontinuous estimated system throughput, which make differences with actual available channel throughput. Thirdly, conservative approach defined in throughput metric is employed to minimize quality loss in the case that under channel protection may still occur when channel varies too fast. This is to guarantee end-user quality as long as there are enough channel resources. After all, end-user quality is always the most direct indication of system performance. Nevertheless, the overall system efficiency has been improved when channel is not hostile and slow varying. In such cases, the system throughput has been tracked and adapted closely to actual available channel throughput, which can be as high as 0.9 when channel is error free.

Figure 6.4 shows the PSNR performance, and Figure 6.5 shows the throughput performance of proposed framework using throughput metric as cost function in low-error channel. It can be seen that the observations in the case of high-error channel are also valid in the case of low-error channel. The proposed framework using throughput metric is able to adapt the system throughput to the variations of channel capacity. It is obvious that error-free video transmission can be achieved with overall high system throughput in low-error channel condition. Hence, it can be deduced that the proposed channel adaptive H.264/AVC video transmission framework using throughput metric will have better PSNR and system throughput performances whenever the channel condition improves.

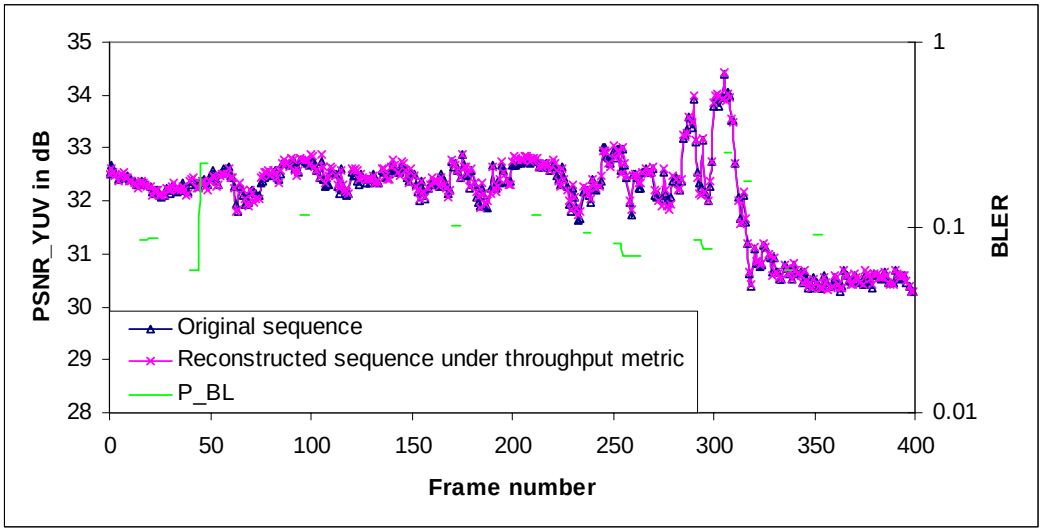


Figure 6.4: PSNR performance of proposed framework using throughput metric as cost function in low-error channel

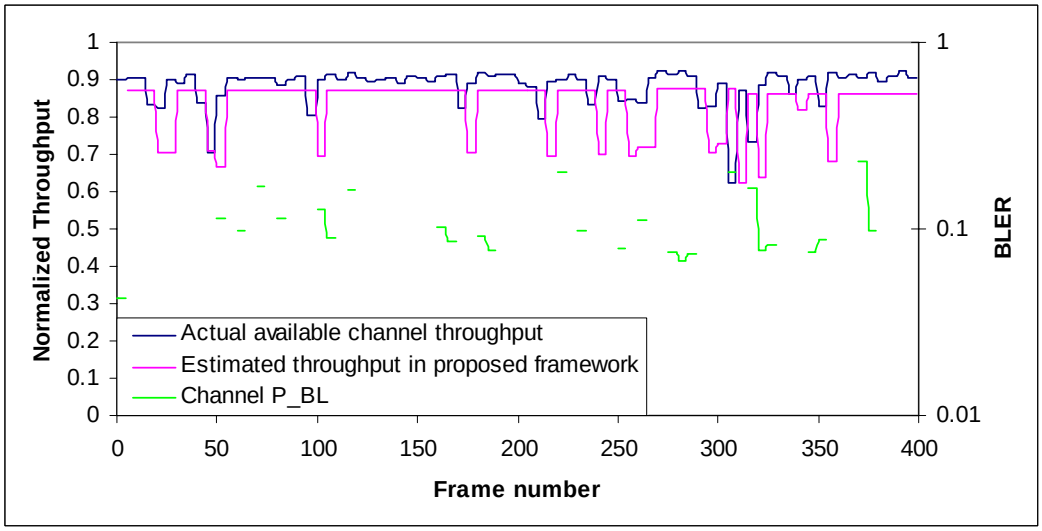


Figure 6.5: Throughput performance of proposed framework using throughput metric as cost function in low-error channel

6.2.2. Performances under Distortion Metric

The proposed channel adaptive H.264/AVC video transmission framework using distortion metric as cost function in the cross layer optimization aims to achieve nearly error free transmission with minimum end-user quality loss. Like the traditional

resource allocation and joint source-channel coding problem, the objective is to minimize end-to-end distortion such that the channel resources are fully used for heavy protection on the video stream. In the proposed framework using distortion metric, the novel adaptive H.264/AVC NAL packetization scheme plays an important role to assist error control mechanisms in such a way that unnecessary source video data overheads can be reduced and data used for channel protection can be increased. Recall that large number of slices per video frame is usually preferred to localize channel errors. However, large number of slices per video frame also means less source coding efficiencies. Within bandwidth constraint, an increase of source coding bits means an equal decrease of bits used for channel protection. Nevertheless, the proposed framework using distortion metric is able to find an optimal solution to this dilemma.

Figure 6.6 shows the PSNR performance, and Figure 6.7 shows the throughput performance of proposed framework using distortion metric as cost function in high-error channel.

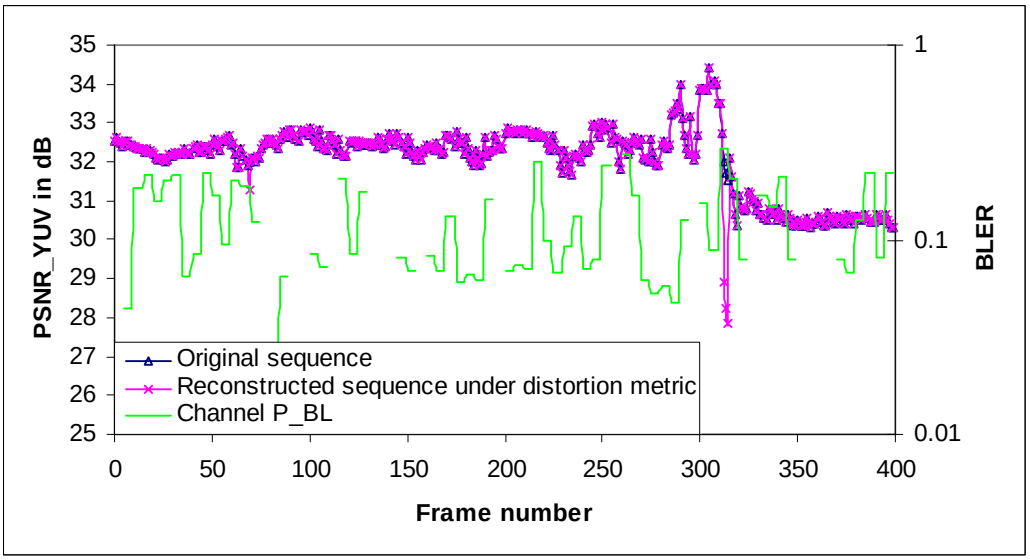


Figure 6.6: PSNR performance of proposed framework using distortion metric as cost function in high-error channel

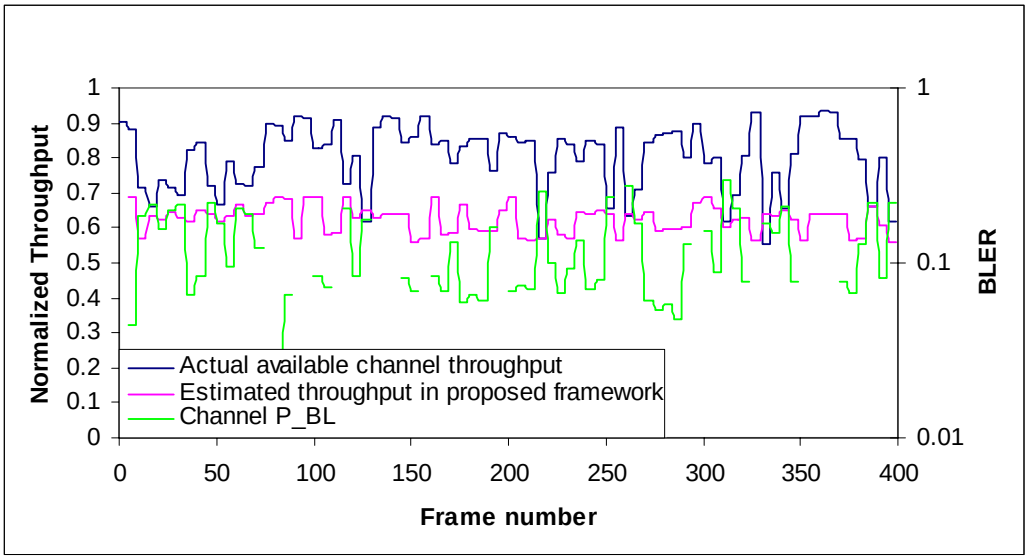


Figure 6.7: Throughput performance of proposed framework using distortion metric as cost function in high-error channel

It can be seen that in the proposed framework using distortion metric, the estimated throughput will not be adapted to the actual available channel throughput any more but rather stay below it in order to achieve nearly error-free transmission. As shown in Figure 6.6, error-free transmission is almost achieved except at frame number 314, where the channel BLER is 0.294 as shown in Figure 6.7. That is why PSNR at frame 314 slightly dropped from 31dB to 28dB due to suddenly worse channel condition. Nevertheless, such high channel BLER does not harm the end-user quality significantly because the video frame 314 has already been partitioned into 7 slices for NAL packetization before the channel becomes hostile, which is able to localize the channel errors and improve end-user quality with the assistant of error concealment.

Figure 6.8 shows the PSNR performance, and Figure 6.9 shows the throughput performance of proposed framework using distortion metric as cost function in low-error channel. As shown in Figure 6.8, error-free transmission is guaranteed since over protection is always the choice. In such case, the estimated system throughput is all the

way below the actual available channel throughput. Therefore, the proposed channel adaptive H.264/AVC video transmission framework can achieve error-free transmission but with sacrificed system efficiency under distortion metric.

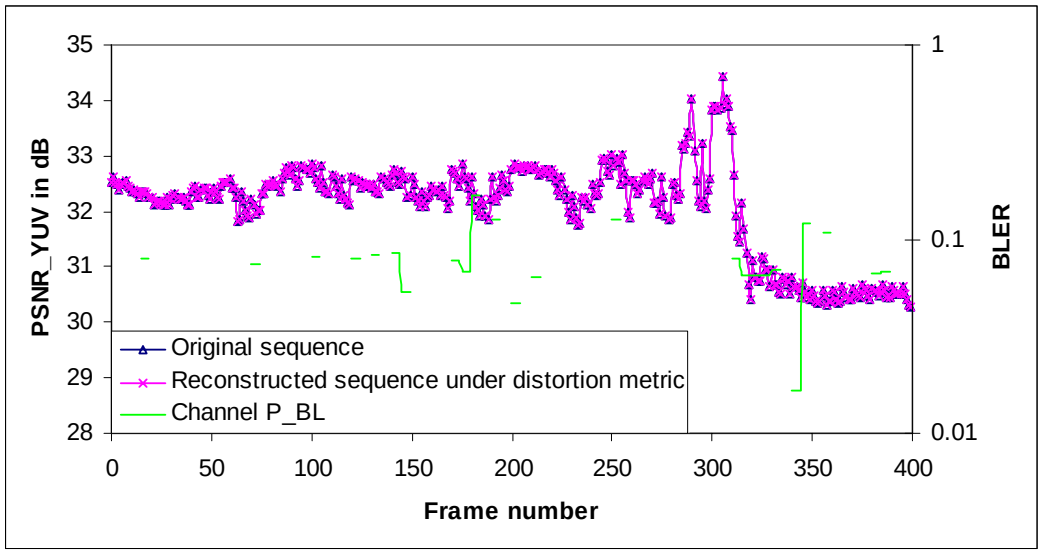


Figure 6.8: PSNR performance of proposed framework using distortion metric as cost function in low-error channel

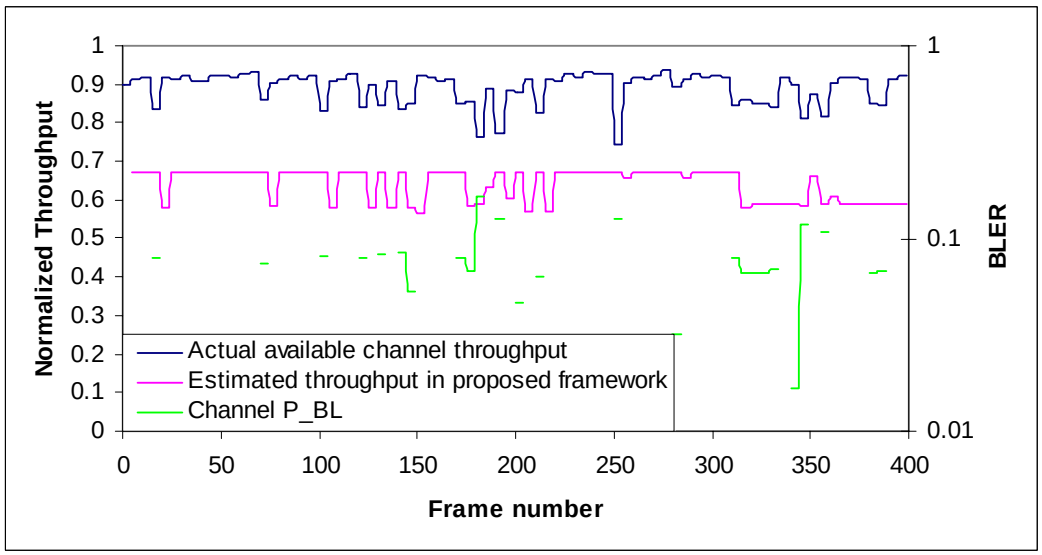


Figure 6.9: Throughput performance of proposed framework using distortion metric as cost function in low-error channel

6.3. Performances between Channel Adaptive H.264/AVC Video Transmission Framework using Throughput Adaptation and System with Fixed NAL Packetization under Fixed Error Control Configuration

6.3.1. Performances in High-Error Channel

Figure 6.10 shows the PSNR performances between proposed framework using throughput adaptation and video transmission system with fixed 4-slice NAL packetization under fixed error control configurations in high-error channel. For systems with fixed 6-slice and 9-slice NAL packetization under fixed error control configurations, the PSNR performances compared with proposed framework using throughput adaptation in high-error channel are shown in Figure 6.11 and Figure 6.12 respectively.

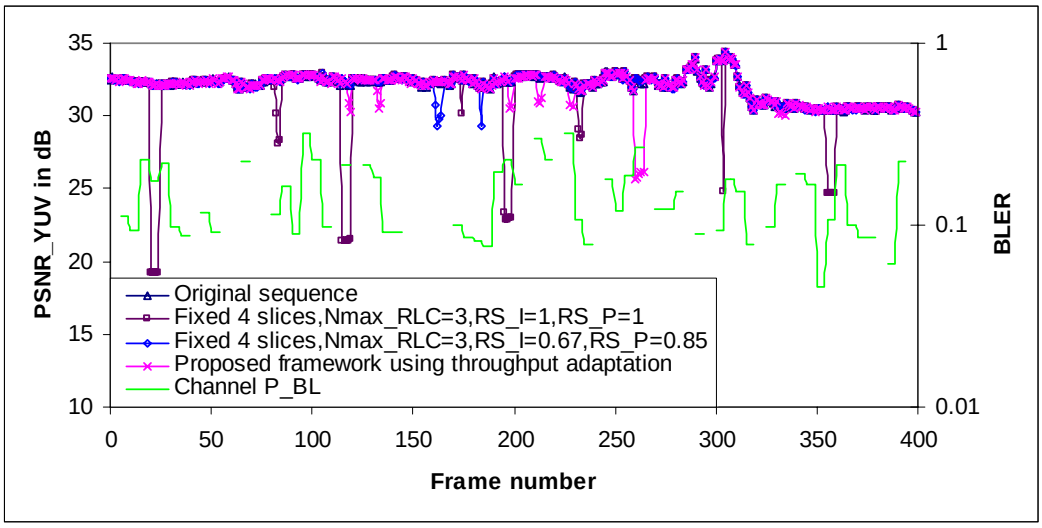


Figure 6.10: PSNR performances between proposed framework using throughput adaptation and system with fixed 4-slice NAL packetization under fixed error control configurations in high-error channel

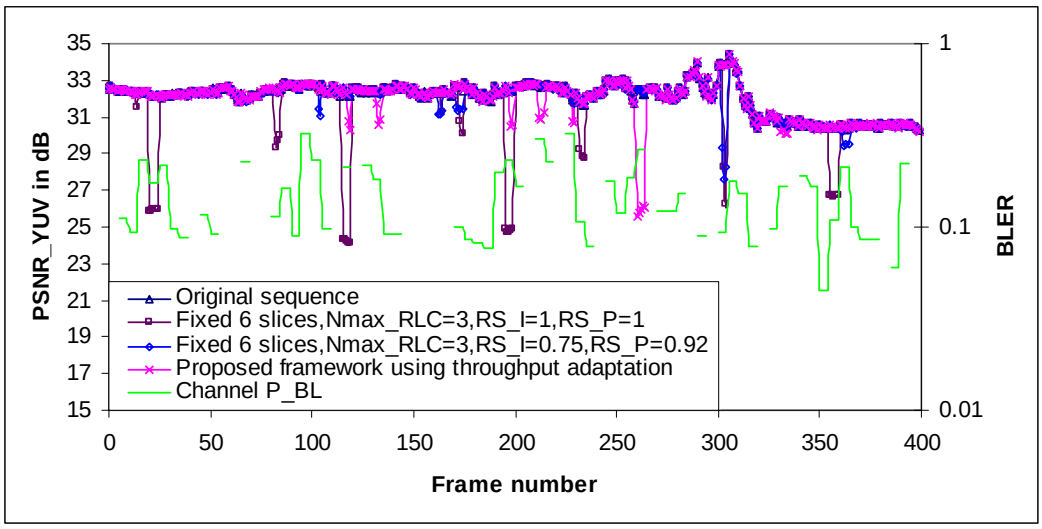


Figure 6.11: PSNR performances between proposed framework using throughput adaptation and system with fixed 6-slice NAL packetization under fixed error control configurations in high-error channel

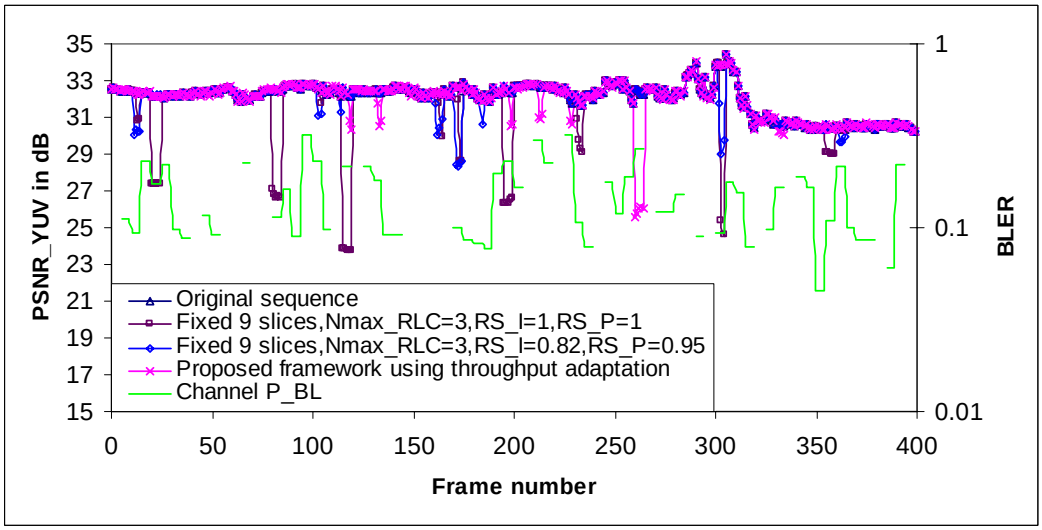


Figure 6.12: PSNR performances between proposed framework using throughput adaptation and system with fixed 9-slice NAL packetization under fixed error control configurations in high-error channel

Table 6.2: Average PSNR performances among the proposed framework using throughput adaptation and video transmission systems with fixed NAL packetization under fixed error control configurations in high-error channel

Fixed no. Slices in NAL Packetization	N_{max}_RLC	RS Code Rate of I-frame NALUs	RS Code Rate of P-frame NALUs	Average PSNR in dB in System with Fixed Configuration	Original Sequence Average PSNR in dB	Average PSNR in dB in Proposed Framework	Improvement in dB
4	3	0.67	0.85	32.05	32.08	31.95	-0.1
		1	1	31.52			0.43
6	3	0.75	0.92	32.01			-0.06
		1	1	31.67			0.28
9	3	0.82	0.95	31.95			0
		1	1	31.64			0.31

Table 6.3: Average throughput performances among the proposed framework using throughput adaptation and video transmission systems with fixed NAL packetization under fixed error control configurations in high-error channel

Fixed No. Slices in NAL Packetization	N_{max}_RLC	RS Code Rate of I-frame NALUs	RS Code Rate of P-frame NALUs	Average Normalized Throughput in System with Fixed Configuration	Average Channel Normalized Throughput	Average Normalized Throughput in Proposed Framework	Improvement %
4	3	0.67	0.85	0.591	0.794	0.731	23.7
		1	1	0.759			N/A
6	3	0.75	0.92	0.639			14.4
		1	1	0.755			N/A
9	3	0.82	0.95	0.671			8.94
		1	1	0.747			N/A

Table 6.2 shows the average PSNR performances over 400 frames, and Table 6.3 shows the average throughput performances among the proposed framework using

throughput adaptation and video transmission systems with fixed NAL packetization under fixed error control configurations in high-error channel.

It is worth to note that in Table 6.2, only the systems with fixed configurations when FEC is enabled (RS code rates for I-frame NALUs and P-frame NALUs are less than 1) have similar PSNR performances to the proposed framework using throughput adaptation. Here, the "systems with fixed configurations" are used as a short representation for "systems with fixed NAL packetization under fixed error control configurations". Although the systems with fixed configurations when FEC is disabled (RS code rates for I-frame NALUs and P-frame NALUs are equal to 1) have average PSNR only 0.5dB less than the proposed framework, they actually perform badly because there are frequent large drops in instantaneous PSNR as shown in Figure 6.10, Figure 6.11, and Figure 6.12. Such difference has been hidden by the statistical average over 400 frames. Therefore, for throughput comparison in Table 6.3, only the systems with fixed configurations when FEC is enabled are compared with proposed framework using throughput adaptation.

Following observations can be drawn from Table 6.3. First of all, the proposed framework using throughput adaptation is able to adapt system throughput to variations of channel capacity, which ensures that the system operates efficiently with acceptable end-user quality. In the systems with fixed configurations, the NAL packetization and error control mechanisms are configured under the assumption that channel is static with constant throughput. For instance, in high-error channel condition, the systems with fixed configurations are configured with constant system throughput ranging from 0.59 and 0.759 as shown in Table 6.3. Hence, such fixed configurations lead to less efficient channel utilization when channel is not hostile compared to the proposed framework using throughput adaptation.

Secondly, for similar PSNR performances, the proposed framework using throughput adaptation has higher system throughput than the systems with fixed configurations. It can be seen in Table 6.3, with similar PSNR performances, the proposed framework using throughput adaptation has system throughput improvement in the range of 8.94% to 23.7% compared to the systems with fixed configurations.

Thirdly, in the proposed framework using throughput adaptation, the novel adaptive H.264/AVC NAL packetization scheme can take the advantage of adaptive slice partition to assist error control mechanisms for throughput enhancement. In the system with fixed 4-slice NAL packetization, the RS code rates for I-frame NALUs and P-frame NALUs have to be set to 0.67 and 0.85 respectively to guarantee end-user quality, and the resulted system throughput is only 0.59. As number of slices per video frame for NAL packetization increases, weaker channel protection has been employed in the systems with fixed 6-slice and 9-slice NAL packetization. In these two systems, average PSNR only drops from 32.05dB to 31.95dB as shown in Table 6.2, but the system throughput has been improved from 0.59 to 0.67 due to less redundant data allocated for channel protection. However, such improvement from 0.59 to 0.67 is still marginal compared to average system throughput of 0.731 in the proposed framework. This is because in the proposed framework using throughput adaptation, when channel is noisy, more slices per video frame for NAL packetization is employed to trade off heavy channel protection, whereas fewer slices per video frame for NAL packetization and light channel protection are employed when channel is amiable.

Therefore, the proposed framework using throughput adaptation can improve channel usage and network or system efficiency even further compared to the system with fixed configuration in the sense that not only the heavy channel protection can be reduced when wireless channel is noisy, but also the unnecessary overheads from

network protocol headers can be avoided whenever wireless channel does not behave hostilely.

6.3.2. Performances in Low-Error Channel

Figure 6.13 shows the PSNR performances between proposed framework using throughput adaptation and video transmission system with fixed 4-slice NAL packetization under fixed error control configurations in low-error channel. For systems with fixed 6-slice and 9-slice NAL packetization under fixed error control configurations, the PSNR performances compared with proposed framework using throughput adaptation in low-error channel are shown in Figure 6.14 and Figure 6.15 respectively.

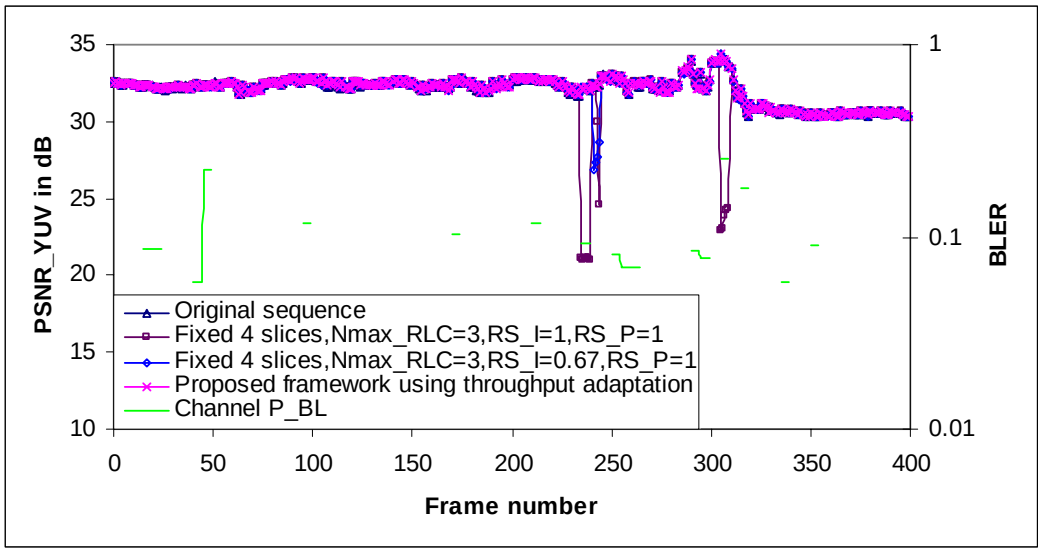


Figure 6.13: PSNR performances between proposed framework using throughput adaptation and system with fixed 4-slice NAL packetization under fixed error control configurations in low-error channel

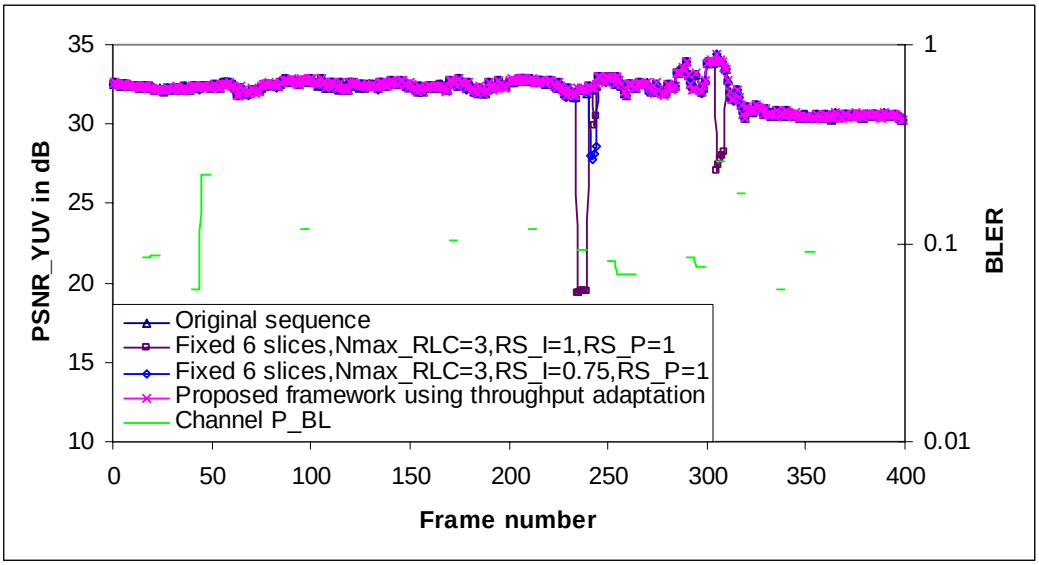


Figure 6.14: PSNR performances between proposed framework using throughput adaptation and system with fixed 6-slice NAL packetization under fixed error control configurations in low-error channel

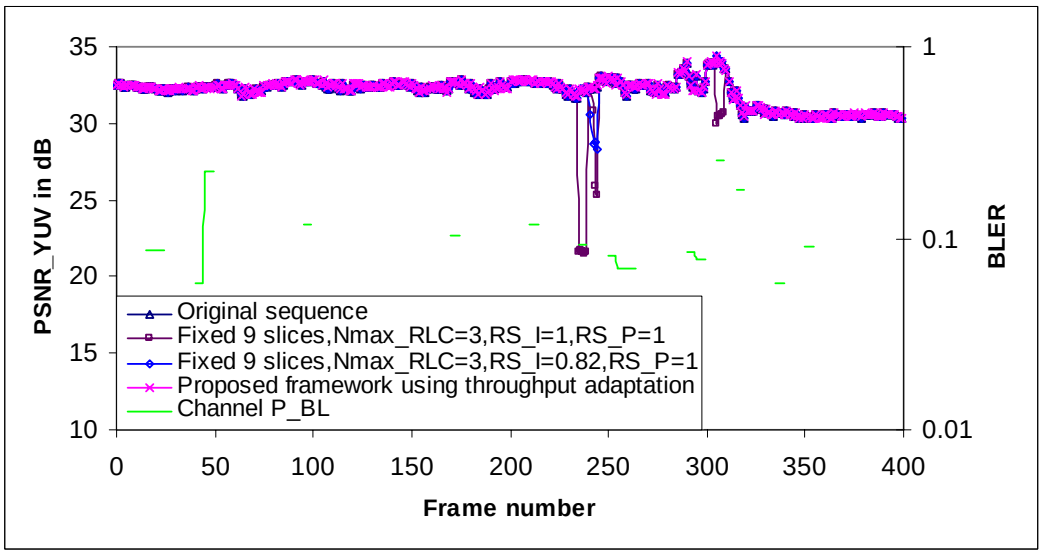


Figure 6.15: PSNR performances between proposed framework using throughput adaptation and system with fixed 9-slice NAL packetization under fixed error control configurations in low-error channel

Table 6.4: Average PSNR performances among the proposed framework using throughput adaptation and video transmission systems with fixed NAL packetization under fixed error control configurations in low-error channel

Fixed No. Slices in NAL packetization	N_{max}_RLC	RS Code Rate of I-frame NALUs	RS Code Rate of P-frame NALUs	Average PSNR in dB in System with Fixed Configuration	Original Sequence Average PSNR in dB	Average PSNR in dB in Proposed Framework	Improvement in dB
4	3	0.67	1	32.04	32.08	32.06	0.02
		1	1	31.80			0.26
6	3	0.75	1	32.04			0.02
		1	1	31.84			0.22
9	3	0.82	1	32.05			0.01
		1	1	31.87			0.19

Table 6.5: Average throughput performances among the proposed framework using throughput adaptation and video transmission systems with fixed NAL packetization under fixed error control configurations in low-error channel

Fixed No. Slices in NAL Packetization	N_{max}_RLC	RS Code Rate of I-frame NALUs	RS Code Rate of P-frame NALUs	Average Normalized Throughput in System with Fixed Configuration	Average Channel Normalized Throughput	Average Normalized Throughput in Proposed Framework	Improvement %
4	3	0.67	1	0.661	0.883	0.833	26
		1	1	0.858			N/A
6	3	0.75	1	0.711			17.2
		1	1	0.846			N/A
9	3	0.82	1	0.742			12.3
		1	1	0.831			N/A

Table 6.4 shows the average PSNR performances over 400 frames, and Table 6.5 shows the average throughput performances among the proposed framework using throughput adaptation and video transmission systems with fixed NAL packetization

under fixed error control configurations in low-error channel. Similarly, for system throughput comparison in Table 6.5, only the systems with fixed configurations when FEC is enabled are compared with proposed framework under similar PSNR performances. The PSNR and throughput performances confirm the observations in the case of high-error channel such that the proposed framework using throughput adaptation has system throughput improvement in the range of 12.3% to 26% compared to the systems with fixed configurations in low-error channel.

There is one more additional observation. That is, even thought channel condition in low-error channel is much better than high-error channel in terms of BER, the hostile channel environment may still occur. Since the system with fixed configuration is unable to response to channel changes, sudden quality drop due to under channel protection is unavoidable unless the system is configured such that the constant system throughput is always below available channel throughput. If that is the case, video data must be always transmitted as over protected, which means that system throughput has to be sacrificed even further. Nevertheless, in the proposed framework using throughput adaptation, the level of channel protection has been adapted to variations of channel capacity so that over channel protection and under channel protection can be minimized. Even when the channel condition suddenly becomes very hostile, the proposed novel adaptive H.264/AVC NAL packetization scheme can improve end-user quality by assigning more slices per video frame.

To summarize, video transmission system with fixed NAL packetization under fixed error control configuration has low system throughput because heavy channel protection has to be employed in order to combat hostile channel conditions for guaranteed end-user quality. On the other hand, the proposed channel adaptive H.264/AVC video transmission framework using throughput adaptation can coordinate

error control mechanisms at different protocol layers and the novel adaptive H.264/AVC NAL packetization scheme to enhance network or system efficiency under cross layer optimization.

6.4. Performances between Channel Adaptive H.264/AVC Video Transmission Framework and System with Fixed NAL Packetization under Channel Adaptive Error Control Configuration

In traditional channel adaptive video transmission [15-21,55] and resource allocation problems [12,56,58-60], the error control configurations are adapted to channel conditions with end-to-end distortion minimization as cost function. The number of slices or packets per video frame is fixed in the system through the whole video transmission. In the proposed channel adaptive H.264/AVC video transmission framework, one important advantage of employing the novel adaptive H.264/AVC NAL packetization scheme working as build-in block is to assist error control mechanisms by localizing the burst errors to improve end-user quality with assistance of error concealment. To show this advantage even further, simulation on the video transmission system with fixed NAL packetizaton under channel adaptive error control configuration is carried out and compared with the proposed framework. Both systems are under throughput metric to adapt the system throughput to variations of channel capacity, hence the average PSNR is the evaluation criteria.

Figure 6.16 and Figure 6.17 show the PSNR performances among the proposed framework with adaptive NAL packetization and the systems with fixed 3-slice, 6-slice, and 9-slice NAL packetization in high-error channel and in low-error channel respectively.

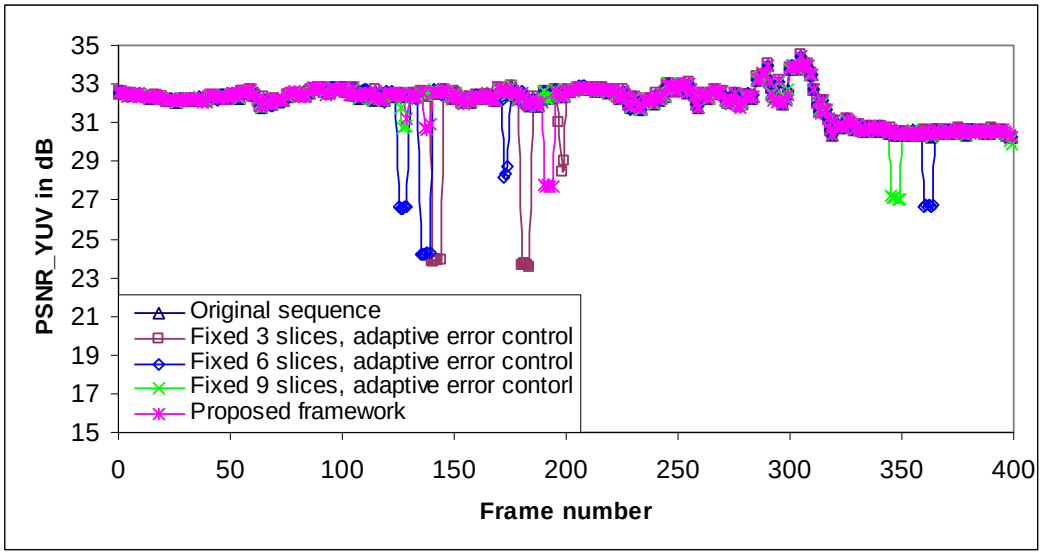


Figure 6.16: PSNR performances among the proposed framework with adaptive NAL packetization and the systems with fixed 3-slice, 6-slice, and 9-slice NAL packetization using throughput metric as cost function in high-error channel

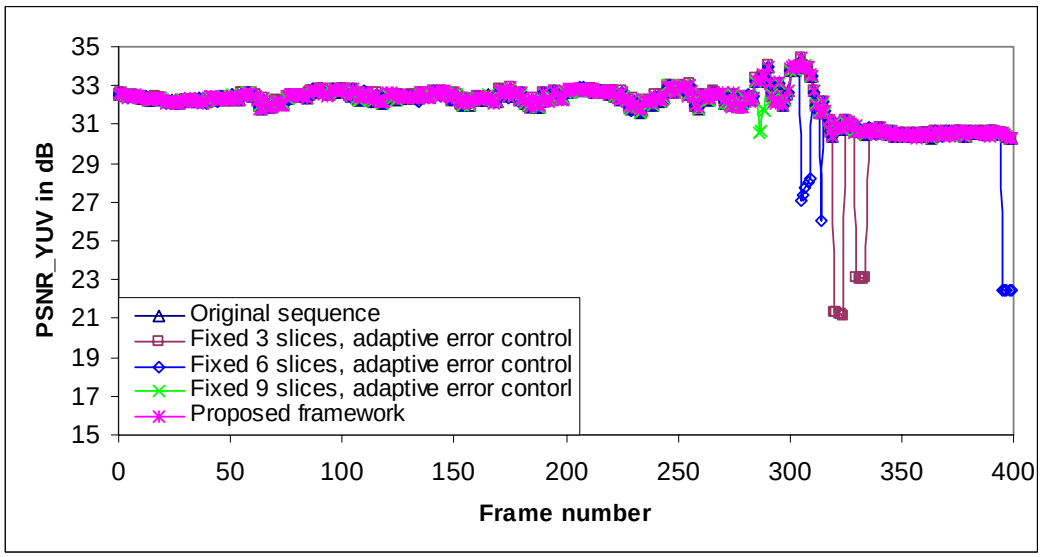


Figure 6.17: PSNR performances among the proposed framework with adaptive NAL packetization and the systems with fixed 3-slice, 6-slice, and 9-slice NAL packetization using throughput metric as cost function in low-error channel

It can be seen that under throughput metric, for high-error channel, in the system with fixed NAL packetization under channel adaptive error control configuration, as expected, with the increase of number of slices per video frame for NAL packetization, the PSNR performance improves. If a large number of slices per video frame is employed for NAL packetiztion, for example, 9 slices per video frame, the PSNR performance is similar to the proposed framework with adaptive NAL packetization. However, as stated in previously, it is not necessary to always partition video frames into large number slices for NAL packetization when the channel is not hostile, because source coding efficiency and codec complexity are sacrificed. For low-error channel, the proposed framework has better PSNR performance than the system with fixed NAL packetization under channel adaptive error control configuration.

6.5. Summary

This chapter presents the performances of the proposed channel adaptive H.264/AVC video transmission framework under cross layer optimization. The framework can coordinate the novel adaptive H.264/AVC packetization scheme and error control mechanisms at different protocol layers to work towards global optimal solution to either system throughput adaptation or end-user quality distortion minimization.

Under throughput metric, simulation results over both high-error and low-error channel conditions show that: i) in both channel conditions, the proposed framework can adapt system throughput to actual available channel throughput such that with acceptable end-user quality, the overall system efficiency can be improved whenever the channel is not hostile and slow varying; ii) in high error channel, although minor PSNR drops suddenly at certain moment when the channel varies too fast, the end-user quality is still acceptable both objectively and subjectively; iii) in low-error channel,

error-free video transmission can be achieved with overall high system throughput. On the other hand, under distortion metric, simulation results show that the proposed framework can achieve error-free transmission but with sacrificed system efficiency in both channel conditions.

The performance of the proposed channel adaptive H.264/AVC video transmission framework using throughput adaptation is also compared to video transmission system with fixed NAL packetization under fixed error control configuration. Simulation results over both high-error and low-error channel conditions show that: i) the video transmission system with fixed NAL packetization under fixed error control configuration cannot improve system efficiency and end-user quality due to the inability to response to wireless channel variations; ii) with similar PSNR performances, the proposed framework using throughput adaptation has system throughput improvement in the range of 8.94% to 23.7% in high-error channel, and in the range of 12.3% to 26% in low-error channel over the system with fixed NAL packetization under fixed error control configuration; iii) the novel adaptive H.264/AVC NAL packetization scheme has been confirmed that it can enhance system throughput in addition to improve end-user quality in the proposed framework using throughput adaptation.

Finally, with similar system throughput performances, the performance of the proposed channel adaptive H.264/AVC video transmission framework is compared to the system with fixed NAL packetization under channel adaptive error control configuration. Simulation results under throughput metric over both high-error and low-error channel conditions show that the proposed framework has better end-user quality, which confirms the advantage of proposed novel adaptive H.264/AVC NAL packetization scheme in improving end-user quality.

Chapter 7

Conclusion and Directions for Future Research

7.1. Concluding Remarks

Multimedia communications in terms of text, voice, audio, and image over wireless mobile networks have been popular in the past decade, and video is the next achievable and attractive area for the realization of wireless transmission of multimedia contents in current mobile computing era. With the development of 3G cellular mobile technology, video transmission over wireless mobile network becomes possible since 3G systems have included video conferencing, video streaming and MMS as dedicated services with guaranteed QoS. However, currently real-time wireless video applications still remain challenges due to the incompatibility between the time-varying and highly error-prone nature of the wireless channel conditions, and the QoS requirements in terms of bandwidth, delay, and packet loss of video applications.

Compared to wired network, bandwidth is always the scarce resource in wireless mobile networks, that's why video data should be compressed in order to meet the bandwidth constraint. Most recent video coding standards adopt predictive coding to remove spatial and temporal redundancies within frame itself or among consecutive frames. And variable length coding is employed to achieve high compression efficiency even further. Predictive and variable length coding make the compressed video data sensitive to channel errors such that even single bit error can cause the loss of synchronization between encoder and decoder, and error propagation among video

frames. Therefore, compressed video data should be protected under error control mechanisms such as FEC and ARQ.

H.264/AVC is the latest and emerging video coding standard which is designed for low bit rate video transmission over wireless mobile networks. Its VCL provides slice-coded video streams with high compression efficiency, and its NAL provides network-friendly capability by packeizing the slice-coded video stream into independent and adaptive network packets known as NALUs. The concept of slice-coding within video frame is introduced to reduce error propagation among video frames because each slice is encoded and decoded independently. It has been shown that partitioning video frame into many slices is helpful to localize channel errors to smaller regions in the video frame, and enhance error resilience of video data with error concealment if the slices representing small regions in the video frame are lost. However, large number of slices within a video frame will reduce source coding efficiency and system throughput due to increased source data, overheads from network protocol headers, and redundant channel data needed to protect them.

The above channel and source approaches could be jointly considered to design a high-quality and efficient video transmission system over error-prone wireless environment. Here, an efficient system is defined as system can transmit video with acceptable end-user quality by using less source, channel and network resources. Current H.264/AVC wireless video transmission system has fixed configurations in terms of fixed NAL packetizaton and fixed error control configuration for less computation and implementation complexities. However, it has low system throughput and quality degradation due to most likely occurred over and under channel protections, which is less efficient because wireless channel is also time-varying and such system is not able to response to channel variations. Since new research directions in the design

of wireless systems do not necessarily attempt to minimize the error rate but to maximize the throughput, an efficient system should be able to adapt throughput to the variation of channel capacity so that the source, channel, and network resources are allocated subject to channel conditions to avoid under and over channel protections.

As recent research on H.264/AVC does not address the issues on the NAL packetization to enhance error resilience and system efficiency, this thesis proposes the novel adaptive H.264/AVC NAL packetization scheme aiming to assist error control mechanisms for end-user quality improvement and system throughput enhancement. With the presumption that channel errors occur in burst, which is usually the case in wireless environment due to fading, the proposed scheme partitions video frames into slices adaptively to channel conditions and encapsulates slices into NALUs by “simple packetization”. The channel conditions have been classified as amiable state, noisy state, and hostile state depending on the BLER feedback from data link layer. When channel is in noisy or hostile state, larger number of slices per video frame for NAL packetization is preferred so that heavy channel protection may not be necessary. When channel is in amiable state, smaller number of slices per video frame for NAL packetization is the choice. In order to track the fast and slow variations of channel precisely, the increment and decrement steps in adaptive slice partition are adapted to the level of improvement or degradation of BLER between current channel state and previous channel state. Simulation results between proposed adaptive NAL packetization scheme and fixed NAL packetizatin scheme when minimum error control mechanisms are employed in the video transmission system show that: i) compared to fixed NAL packetization scheme with a few slices per video frame, the proposed adaptive NAL packetization scheme provides significant decoder gain of 2-12dB. This improvement has a significant impact because video frames which were

originally unacceptable objectively are now acceptable; ii) compared to fixed NAL packetization scheme with larger number of slices per video frame, the proposed adaptive NAL packetization scheme offers comparable PSNR performance using fewer slices per video frame.

This advantage in improving end-user quality is used to enhance system efficiency by considering the novel adaptive H.264/AVC NAL packetizaiton scheme as a built-in block in the channel adaptive H.264/AVC video transmission framework proposed later on due to the following observations at different channel conditions. When channel is in amiable state, the BLER and BER are so low that channel errors can be eliminated by data link layer transport block ARQ. In this case, smaller number of slices per video frame for NAL packetization will lead to high source coding efficiency and less overheads from network protocol headers. Hence, system throughput can be improved because data link layer transport block ARQ is sufficient to combat channel errors and redundant data from application layer FEC may not be necessary.

When channel is in noisy state, more slices are generated per video frame for NAL packetization. If the data link layer transport block ARQ cannot clear errors within limited number of retransmissions, the application layer FEC will correct the errors. In this case, weak FEC is sufficient in the noisy state. If unfortunately, there are errors left after application layer FEC, the NALUs containing errors will be dropped and the error concealment mechanism at video decoder will conceal the loss. Since the degradation of end-user quality is only restricted to error-corrupt slices, the error propagation has been restricted to smaller regions in the video frame so that the end-user quality can be improved. The system throughput can be improved due to less redundant data used for channel protection.

When channel is in hostile state, large number of slices per video frame for NAL packetization is preferred for 3 reasons. Firstly, as larger number of slices per video frame can localize the burst errors into smaller regions in the video frame, error control mechanisms adopted in the system can be assisted even further. Secondly, since the actual available channel throughput in the hostile state is already low, the overheads from network protocol headers are less significant compared to loss of NALU payload. Thirdly, from MDS coding point of view, large number of slices per video frame means large number of source packets per coding group. With the same error correct capability (same number of redundant parity packets), the code rate is higher than the case of less source packets per coding group. Hence, system throughput can be enhanced by avoiding heavy channel protection due to high MDS coding rate if certain loss of end-user quality is acceptable.

With above observations, this thesis incorporates the novel adaptive H.264/AVC NAL packetization scheme to video transmission system and proposes channel adaptive H.264/AVC video transmission framework. As discussed above, the novel adaptive H.264/AVC NAL packetization scheme has to work with lower layer FEC and ARQ for throughput adaptation. Since single layer approach has been proved to be inefficient and inflexible in adapting to the constantly changes in network conditions, cross layer design is a natural approach to facilitate throughput adaptation in proposed channel adaptive H.264/AVC video transmission framework.

The proposed channel adaptive H.264/AVC video transmission framework focuses on end-to-end system design, which consists of five channel adaptive components, namely i) adaptive H.264/AVC NAL packetization, ii) end-to-end distortion estimation, iii) channel quality measurement, iv) bit rate estimation, and v) error control adaptation. The framework is designed through the interaction between the video codec and the

underlying layers. We first assume that lower layers provide a set of given adaptation components to application layer and our H.264/AVC encoder can access, and specify those adaptation components. At application layer, this framework performs adaptive NAL packetization and assigns FEC according to the importance of the NALUs. The channel protected NALUs are attached with RTP/UDP/IP headers and processed by lower layers. At data link layer, RLC/RLP retransmission is performed for each transport block. Here, unlike the traditional approach in the design of wireless video transmission system that is trying to allocate source and channel resources by minimizing end-to-end distortion, throughput metric with aggressive, neutral, and conservative adjustments is used as cost function in the optimization. More specifically, the slice partition for NAL packetization, level of FEC, and the number of allowed RLC/RLP retransmission within adaptation period are selected based on channel conditions so that the system throughput is adapted to the variations of channel capacity with bandwidth as constraint. Furthermore, for completeness and flexibility, the proposed framework includes traditional distortion metric as well.

In order to show the performances of the proposed channel adaptive H.264/AVC video transmission framework under cross layer optimization, average reconstructed PSNR at the receiver and normalized system throughput are used as evaluation criteria. The JVT test conditions error patterns 1 and 2 are used to simulate 3G wireless mobile environment as high-error channel and low-error channel respectively. Under throughput metric, simulation results show that: i) in both channel conditions, the proposed framework can adapt system throughput to actual available channel throughput such that with acceptable end-user quality, the overall system efficiency can be improved whenever the channel is not hostile and slow varying; ii) in high error channel, although minor PSNR drops suddenly at certain moment when the channel

varies too fast, the end-user quality is still acceptable both objectively and subjectively; iii) in low-error channel, error-free video transmission can be achieved with overall high system throughput. Hence, we conclude that the proposed framework using throughput metric will have better PSNR and system throughput performances whenever the channel condition improves. On the other hand, under distortion metric, simulation results show that the proposed framework can achieve error-free transmission but with sacrificed system efficiency in both channel conditions.

The performance of the proposed channel adaptive H.264/AVC video transmission framework using throughput adaptation is also compared to video transmission system with fixed NAL packetization under fixed error control configuration. Simulation results over both high-error and low-error channel conditions show that: i) the system with fixed NAL packetization under fixed error control configuration cannot improve system efficiency and end-user quality due to the inability to response to wireless channel variations; ii) with similar end-user quality, the proposed framework using throughput adaptation has system throughput improvement over the system with fixed NAL packetization under fixed error control configuration in the range of 8.94% to 23.7% in high-error channel and in the range of 12.3% to 26% in low-error channel; iii) the novel adaptive H.264/AVC NAL packetization scheme has been confirmed that it can enhance system throughput in addition to improve end-user quality in the proposed framework using throughput adaptation. Hence, we conclude that the proposed H.264/AVC video transmission framework using throughput adaptation can improve the average channel usage and system efficiency in the sense that not only the heavy channel protection can be reduced when wireless channel is noisy, but also the unnecessary overheads from network protocol headers can be avoided whenever wireless channel does not behave hostilely.

Finally, with similar system throughput, the performance of the proposed channel adaptive H.264/AVC video transmission framework is compared to the system with fixed NAL packetization under channel adaptive error control configuration. Simulation results under throughput metric over both high-error and low-error channel conditions show that the proposed framework has better end-user quality, which confirms the advantage of proposed novel adaptive H.264/AVC NAL packetization scheme in improving end-user quality for wireless H.264/AVC video applications.

7.2. Directions of Future Research

There are a few areas that future research can explore:

First of all, in this thesis, only channel distortion is under consideration by decoupling source distortion. Indeed, as stated in Chapter 5, distortion due to source coding also contributes significant to overall distortions. In some literatures, such as [30], [59] and [60], source coding is jointly considered with channel coding. Source coding is a meaningful research area that could be explored for throughput maximization. This is due to the fact that decreasing the QP will result in improved video quality prior to transmission, and channel distortion can be tolerated to even large extend. Hence, by joint considering source coding, channel coding, ARQ, and slice partition, either the system throughput or end-user quality may be improved further compared to current approach of excluding source coding.

Secondly, the source rate as a function of QP and number of slices per video frame is modeled as linear by exhaustive experiments on various video testing sequences in this thesis. Although such approach provides a fast estimation, it will be better if accurate bit rate could be found. However, as stated in the thesis, it is almost impossible in real time because encoding always happens after the estimation unless

the video sequences have been encoded and stored into multiple versions before hand. Future research could find a method to solve this problem.

Thirdly, although H.264/AVC standard introduces data partition to enhance error resilience, this thesis does not explore the data partition due to the concern of overheads. Under data partition scheme, one NALU will be split into three NALUs. It is definitely a problem for low bit rate video transmission. However, if larger bandwidth is available, it may not be a problem. Hence, UEP could be extended to syntax element level in addition to NALU level in the future research.

Fourthly, regarding error control mechanisms, packet-level FEC and local transport block ARQ are employed in this thesis. In future research, byte-level FEC and packet-level ARQ may be employed. However, the combination of those error control techniques needs more intelligent way to find the optimal solutions. Searching algorithm is time consuming, which results in unacceptably computational complexity.

Finally, in the resource allocation problem, considering end-to-end distortion as cost function in the optimization is the traditional approach. This thesis employs system throughput as optimization metric. Furthermore, energy or power consumption can also be the cost function as well [60]. In [77], the H.264/AVC codec -designed particularly to improve coding efficiency and network friendliness – was simulated and profiled at the instruction-level so that complexity and power consumption can be derived. Different cost functions in an optimization problem will lead the system to have different performances. I have always been interested in whether it is possible to combine different cost functions for global solutions. It is not necessary that the combination of different cost functions is practical in the actual system. I just bring about the idea that transpired during my research work, hoping that it may give some hints to people who would be interested in this research area.

Appendix A

Experiments on Source Coding

A.1. "Foreman" Sequence

Figure A.1 shows the source coding bit rate as a function of number of slices per frame at different QPs. Figure A.2 shows the average number of bits per I-frame as function of number of slices per frame at different QPs. Figure A.3 shows the average number of bits per I-slices as function of number of slices per frame at different QPs. Figure A.4 shows the average number of bits per P-frame as function of number of slices per frame at different QPs. Figure A.5 shows the average number of bits per P-slices as function of number of slices per frame at different QPs.

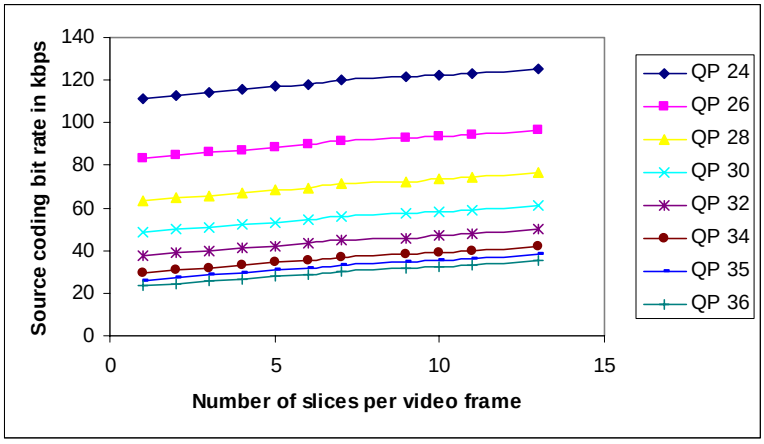


Figure A.1: "Foreman" sequence source coding bit rate

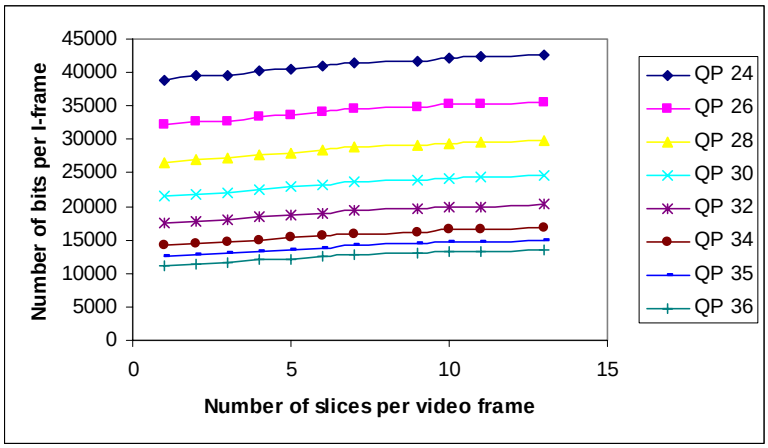


Figure A.2: "Foreman" sequence average number of bits per I-frame

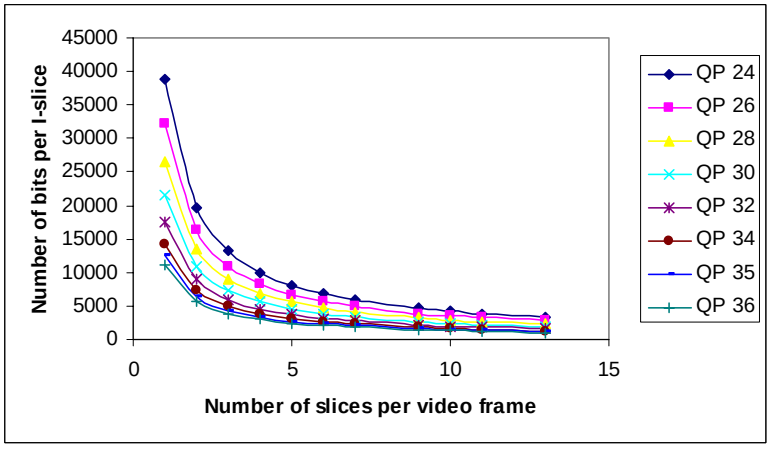


Figure A.3: "Foreman" sequence average number of bits per I-slice

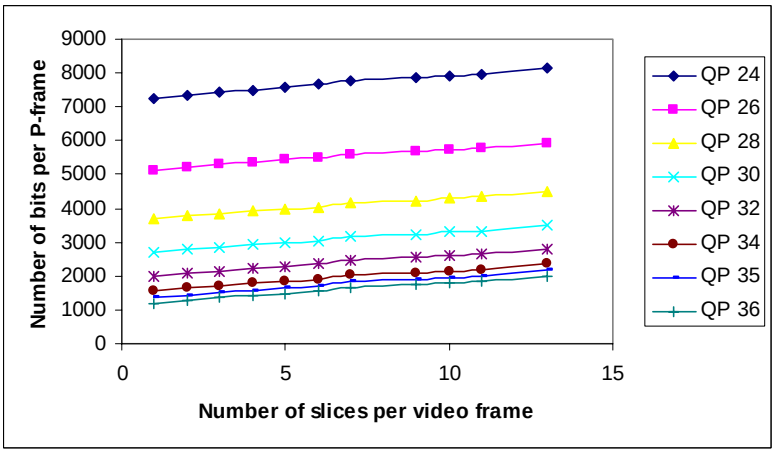


Figure A.4: "Foreman" sequence average number of bits per P-frame

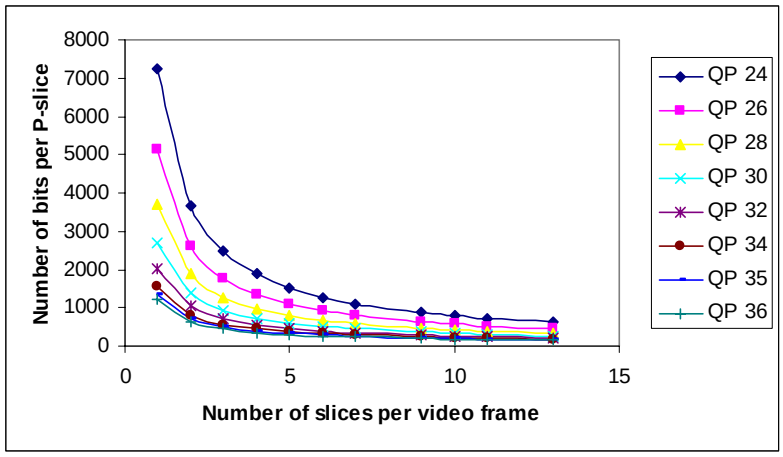


Figure A.5: "Foreman" sequence average number of bits per P-slice

A.2. "Carphone" Sequence

Figure A.6 shows the source coding bit rate as a function of number of slices per frame at different QPs. Figure A.7 shows the average number of bits per I-frame as function of number of slices per frame at different QPs. Figure A.8 shows the average number of bits per I-slices as function of number of slices per frame at different QPs. Figure A.9 shows the average number of bits per P-frame as function of number of slices per frame at different QPs. Figure A.10 shows the average number of bits per P-slices as function of number of slices per frame at different QPs.

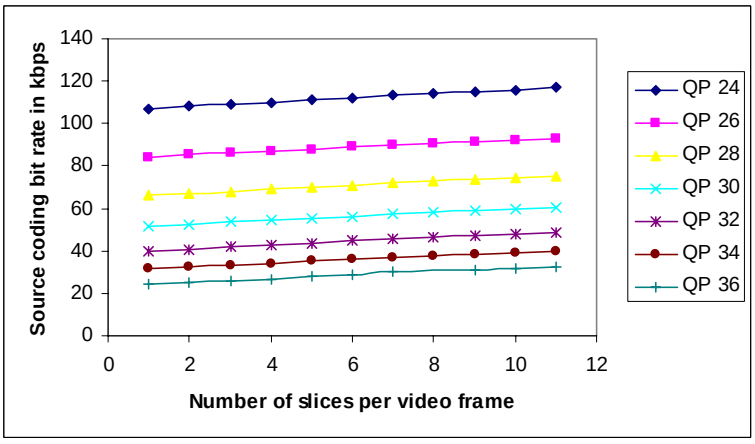


Figure A.6: "Carphone" sequence source coding bit rate

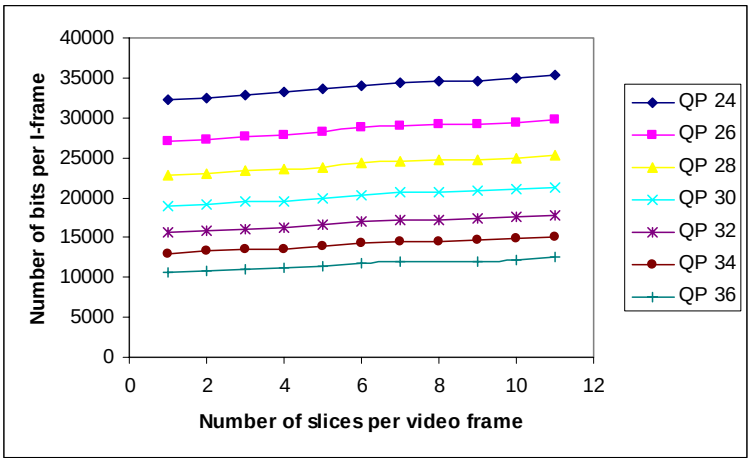


Figure A.7: "Carphone" sequence average number of bits per I-frame

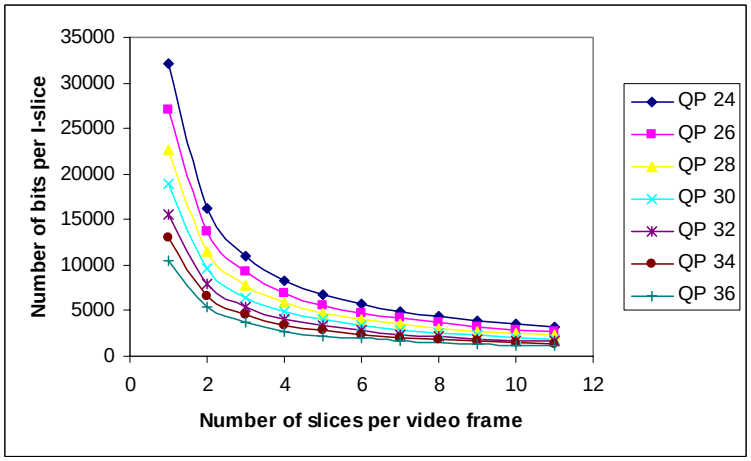


Figure A.8: "Carphone" sequence average number of bits per I-slice

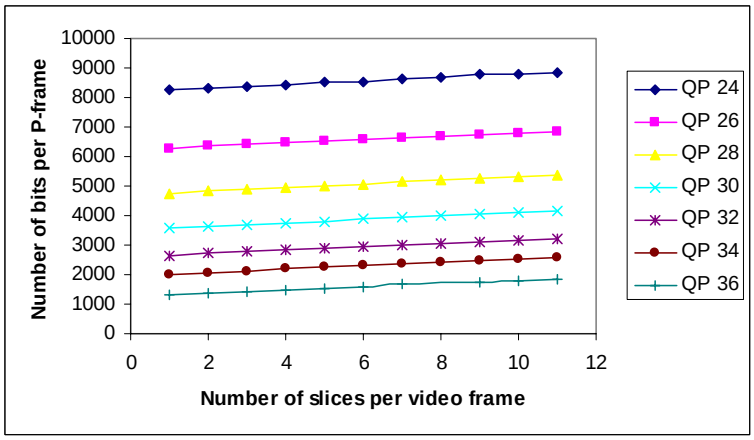


Figure A.9: "Carphone" sequence average number of bits per P-frame

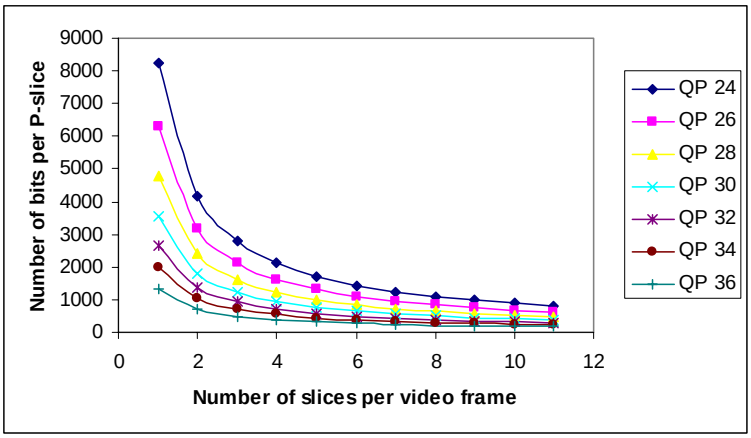


Figure A.10: "Carphone" sequence average number of bits per P-slice

A.3. "Suzie" Sequence

Figure A.11 shows the source coding bit rate as a function of number of slices per frame at different QPs. Figure A.12 shows the average number of bits per I-frame as function of number of slices per frame at different QPs. Figure A.13 shows the average number of bits per I-slices as function of number of slices per frame at different QPs. Figure A.14 shows the average number of bits per P-frame as function of number of slices per frame at different QPs. Figure A15 shows the average number of bits per P-slices as function of number of slices per frame at different QPs.

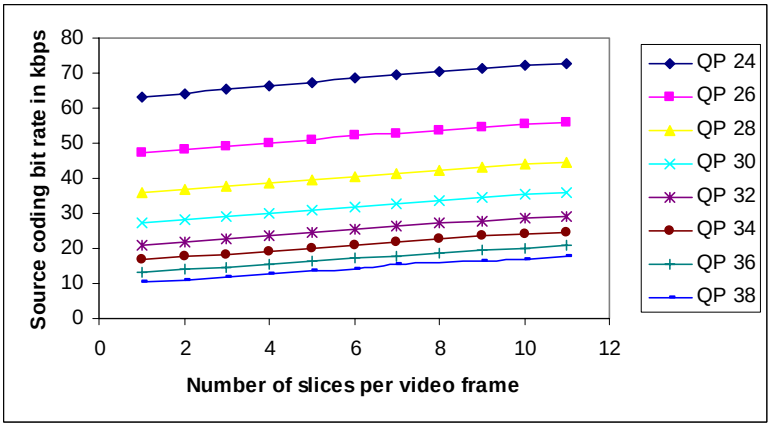


Figure A.11: "Suzie" sequence source coding bit rate

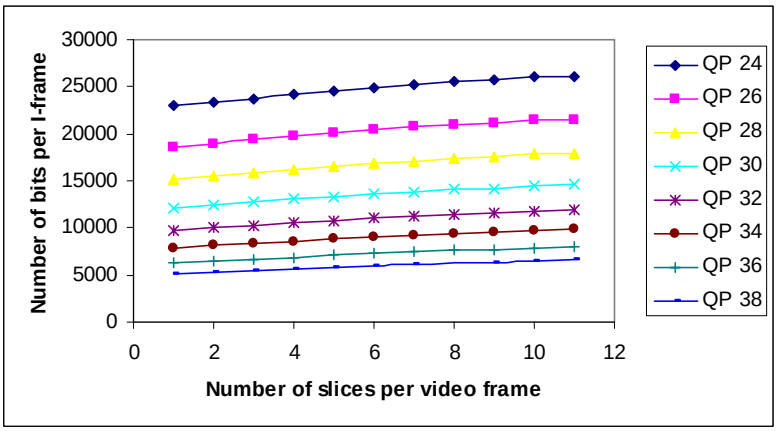


Figure A.12: "Suzie" sequence average number of bits per I-frame

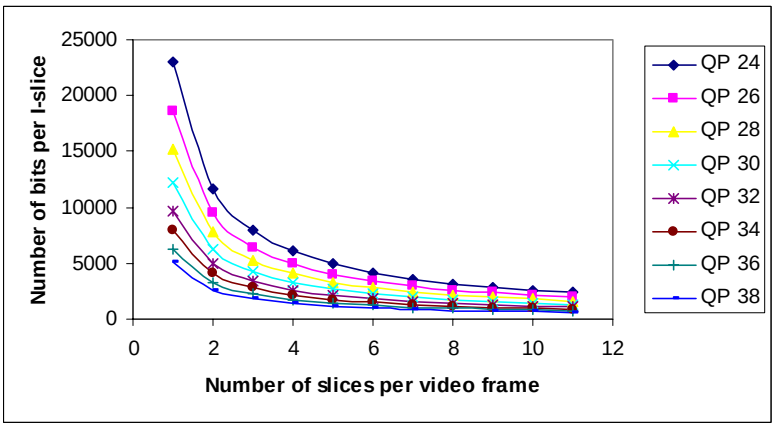


Figure A.13: "Suzie" sequence average number of bits per I-slice

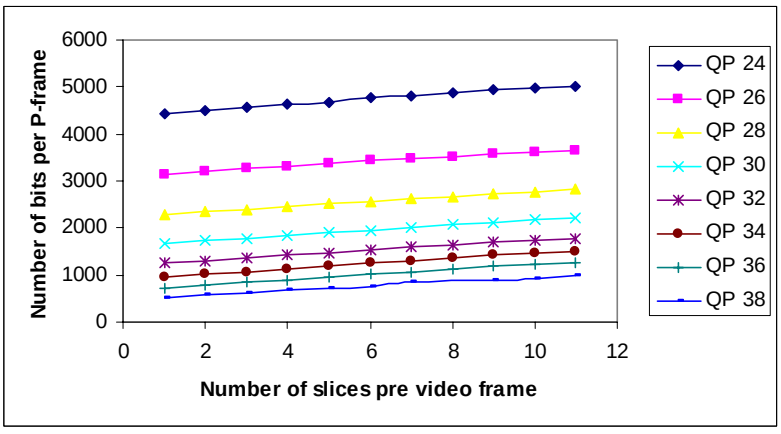


Figure A.14: "Suzie" sequence average number of bits per P-frame

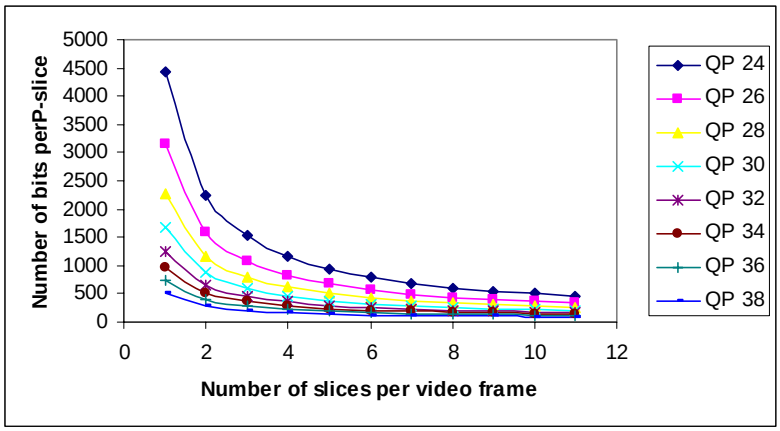


Figure A.15: "Suzie" sequence average number of bits per P-slice

A.4. "Claire" Sequence

Figure A.16 shows the source coding bit rate as a function of number of slices per frame at different QPs. Figure A.17 shows the average number of bits per I-frame as function of number of slices per frame at different QPs. Figure A.18 shows the average number of bits per I-slices as function of number of slices per frame at different QPs. Figure A.19 shows the average number of bits per P-frame as function of number of slices per frame at different QPs. Figure A.20 shows the average number of bits per P-slices as function of number of slices per frame at different QPs.

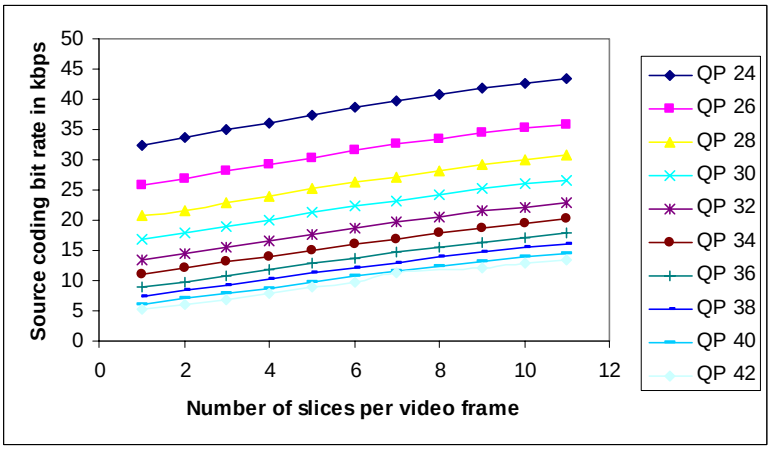


Figure A.16: "Claire" sequence source coding bit rate

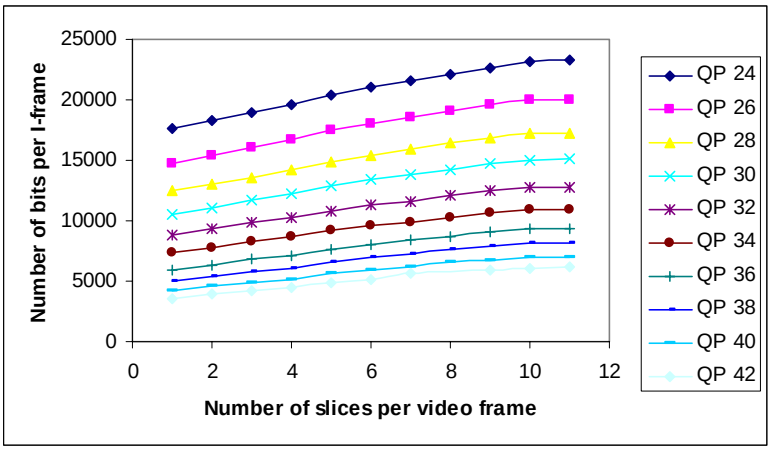


Figure A.17: "Claire" sequence average number of bits per I-frame

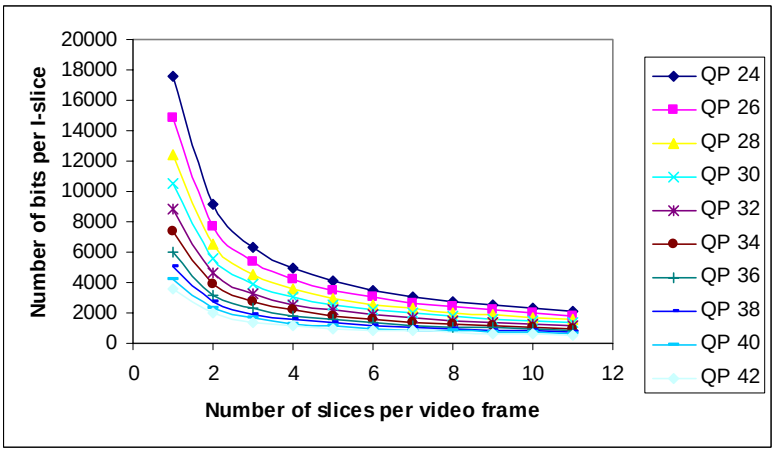


Figure A.18: "Claire" sequence average number of bits per I-slice

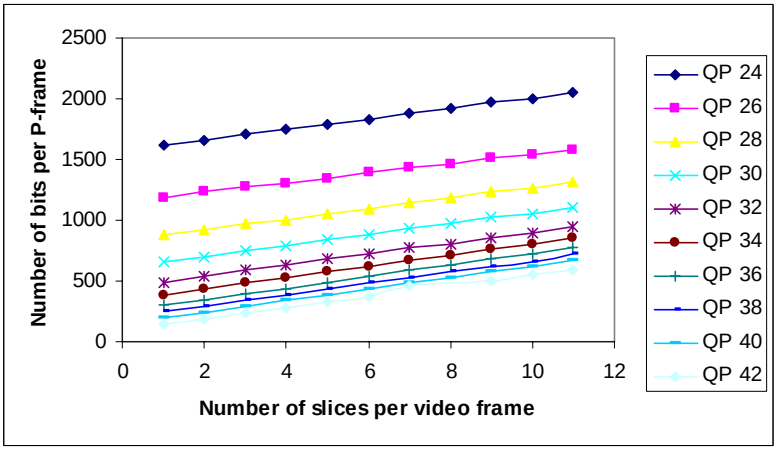


Figure A.19: "Claire" sequence average number of bits per P-frame

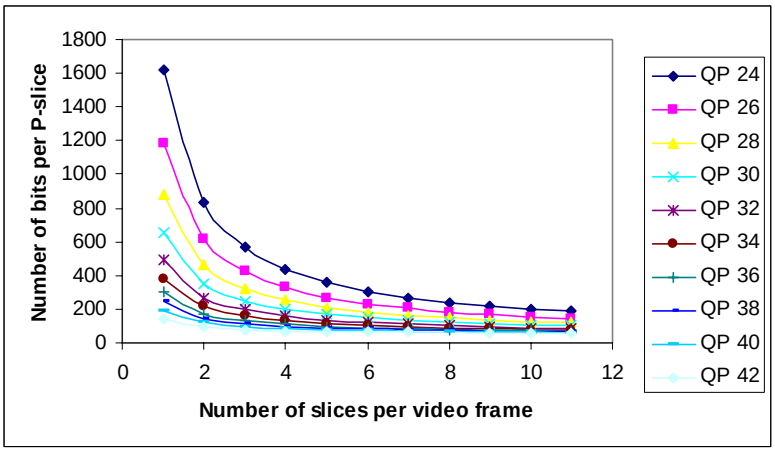


Figure A.20: "Claire" sequence average number of bits per P-slice

Appendix B

Overheads in Slice-Coding

In H.264/AVC video transmission, the overhead is defined as packet header over packet size in equation (4.1). Packet header is 40-byte RTP/UDP/IP header for IPv4, while packet payload is H.264/AVC NALU. Table B.1 shows the overheads in "Foreman" sequence due to slice coding for different QPs. Table B.2 shows the overheads in "Carphone" sequence due to slice coding for different QPs. Table B.3 shows the overheads in "Suzie" sequence due to slice coding for different QPs. Table B.4 shows the overheads in "Claire" sequence due to slice coding for different QPs. It is obvious that for each QP, the larger number of slices per video frame, the higher the overheads.

Table B.1: Overheads in "Foreman" sequence due to slice coding for different QPs

Number of slices per video frame	QP = 30		QP = 32		QP = 34		QP = 36	
	I	P	I	P	I	P	I	P
1	1.5%	10.7%	1.8%	13.8%	2.2%	17.1%	2.7%	21%
2	2.8%	18.7%	3.5%	23.4%	4.2%	28.1%	5.2%	33.2%
3	4.2%	25.3%	5.1%	30.9%	6.2%	36.1%	7.6%	41.5%
4	5.4%	30.5%	6.5%	36.5%	7.8%	41.8%	9.6%	47.4%
5	6.5%	34.9%	7.9%	41.1%	9.4%	46.4%	11.6%	51.8%
6	7.7%	38.6%	9.2%	44.8%	11%	50.1%	13.4%	55.3%
7	8.7%	41.5%	10.4%	47.6%	12.3%	52.6%	14.9%	57.4%
8	9.7%	44%	11.5%	50%	13.8%	54.9%	16.7%	61.9%
9	10.8%	47.1%	12.8%	53.2%	15.1%	58%	18.2%	62.4%
10	11.7%	49.3%	13.8%	55.2%	16.2%	59.8%	19.5%	64.2%
11	12.6%	51.4%	14.9%	57.2%	17.5%	61.6%	20.9%	65.8%

Table B.2: Overheads in "Carphone" sequence due to slice coding for different QPs

Number of slices per video frame	QP = 30		QP = 32		QP = 34		QP = 36	
	I	P	I	P	I	P	I	P
1	1.7%	8.2%	2%	10.8%	2.4%	13.8%	2.9%	19.4%
2	3.2%	15%	3.9%	19.1%	4.6%	23.6%	5.6%	31.5%
3	4.7%	20.6%	5.6%	25.7%	6.6%	31.1%	8%	39.9%
4	6.1%	25.4%	7.3%	31%	8.6%	36.7%	10.3%	46.2%
5	7.4%	29.6%	8.8%	35.6%	10.3%	41.6%	12.3%	50.9%
6	8.6%	33.1%	10.2%	39.4%	11.9%	45.5%	14.1%	54.7%
7	9.8%	36.2%	11.6%	42.5%	13.4%	48.6%	15.8%	57%
8	11%	39%	13%	45.4%	15%	51.3%	17.3%	59.2%
9	12.2%	41.5%	14.2%	48%	16.4%	53.7%	19.3%	62.2%
10	13.2%	43.8%	15.4%	50.2%	17.7%	56%	20.7%	64.2%
11	14.2%	45.9%	16.5%	52.2%	18.9%	57.7%	22%	65.7%

Table B.3: Overheads in "Suzie" sequence due to slice coding for different QPs

Number of slices per video frame	QP = 26		QP = 28		QP = 30		QP = 32	
	I	P	I	P	I	P	I	P
1	1.7%	9.2%	2.1%	12.3%	2.6%	16.1%	3.2%	20.3%
2	3.3%	16.7%	4%	21.5%	4.9%	27%	6%	32.9%
3	4.7%	22.7%	5.7%	28.6%	7%	35%	8.6%	41.3%
4	6.1%	28%	7.3%	34.3%	8.9%	40.9%	10.9%	47.4%
5	7.4%	32.2%	8.8%	38.9%	10.7%	45.7%	13%	51.9%
6	8.6%	35.9%	10.2%	42.8%	12.3%	49.5%	14.9%	55.7%
7	9.8%	39.2%	11.6%	46.2%	13.8%	52.7%	16.6%	58.5%
8	10.9%	42.1%	12.9%	48.9%	15.4%	55.3%	18.3%	61%
9	12%	44.6%	14.1%	51.4%	16.8%	57.5%	19.9%	63.1%
10	13%	47%	15.2%	53.6%	18%	60%	21.3%	64.9%
11	14%	49.1%	16.5%	55.5%	19.4%	61.4%	22.9%	66.3%

Table B.4: Overheads in "Claire" sequence due to slice coding for different QPs

Number of slices per video frame	QP = 24		QP = 26		QP = 28		QP = 30	
	I	P	I	P	I	P	I	P
1	1.8%	16.5%	2.1%	2.1%	2.5%	26.7%	2.9%	32.7%
2	3.4%	27.8%	4%	34.2%	4.7%	41%	5.5%	47.8%
3	4.8%	36%	5.6%	43%	6.6%	49.7%	7.6%	56.2%
4	6.1%	42.2%	7.1%	49.5%	8.3%	56%	9.5%	61.8%
5	7.3%	47.2%	8.4%	54.3%	9.7%	60.3%	11.1%	65.5%
6	8.3%	51.1%	9.6%	58%	11.1%	63.6%	12.5%	68.5%
7	9.4%	54.4%	10.8%	61%	12.4%	66.3%	14%	70.6%
8	10.4%	57.1%	11.8%	63.6%	13.5%	68.5%	15.2%	72.4%
9	11.3%	59.4%	12.8%	65.4%	14.6%	70%	16.4%	73.8%
10	12.1%	61.5%	13.8%	67.5%	15.7%	71.7%	17.5%	75.1%
11	13.2%	63.2%	15%	69%	16.9%	72.8%	18.9%	76.1%

Publication Lists

[1]. H. D. Le, M. Zhao and M. T. Le, "Performance of Transceiving JPEG2000 Image over Peer-to-Peer Wireless Link," *Proc. IEEE The Ninth International Conference on Communication Systems (ICCS 2004)*, pp.255-259, Singapore, Sep 2004.

[2]. M. Zhao and T. M. Le, "Robust Transmission of JPEG-2000 Images over Peer-to-Peer Mixed Wired and Wireless Links," to appear in *Proc. IEEE Wireless Communications and Network Conference (WCNC 2005)*, New Orleans, LA, USA, March 2005.

[3]. M. Zhao and T. M. Le, "A Novel Channel Adaptive H.264/AVC Network Abstraction Layer Packetization Scheme for Video Transmission in Wireless Environment," submitted to *IEEE Trans. Circuits Syst. Video Technol*., 2005.

[4]. M. Zhao and T. M. Le, "Channel Adaptive H.264/AVC Video Transmission over Wireless Mobile Networks under Cross Layer Optimization," submitted to *IEEE Trans. Wireless Comms*., 2005.

Bibliography

[1]. S. Wenger, M. Hannuksela, and T. Stockhammer, “Identified H.26L Applications,” ITU-T SG 16, Doc. VCEG-L34, Eibsee, Germany, 2001.

[2]. “Codec for Circuit Switched Multimedia Telephony Service: General Discription,” 3GPP Technical Specification 3GPP TS 26.110, V5.0.0, Jun. 2002.

[3]. “Packet Switched Conversational Multimedia Applications: Default Codecs,” 3GPP Technical Specification 3GPP TS 26.235, V6.0.0, Jun. 2003.

[4]. S. Wenger, “H.264/AVC over IP,” *IEEE Trans. Circuits Syst. Video technol.*, vol.13, pp. 645-656, Jul. 2003.

[5]. “Transparent End-to-End Packet Switched Streaming Services (PSS): RTP Usage Model,” 3GPP Technical Specification 3GPP TS 26.937, V6.0.0, Mar. 2004.

[6]. “Multimedia Messaging Services (MMS): Media Formats and Codecs,” 3GPP Technical Specification 3GPP TS 26.140, V5.2.0, Dec. 2002.

[7]. T. Stockhammer, M. M. Hannuksela and T. Wiegand, “H.264/AVC in wireless environments,” *IEEE Trans. Circuits Syst. Video technol.*, vol.13, pp. 657-673, Jul. 2003.

[8]. ITU-T/SG 16/VCEG (formerly Q.15 now Q.6), H.26L Test Model Long-Term Number 7 (TML-7), Doc. VCEG-M81, Apr. 2001.

[9]. "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Recommendation H.264/ISO/IEC 14496-10 AVC)," in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, Mar. 2003.

[10]. "Information Technology—Multimedia Content Description Interface—Part 3: Visual," ISO/IEC FDIS 15938-3, Dec. 2004.

[11]. T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video technol.*, vol.13, pp. 560-576, Jul. 2003.

[12]. C. C. Liu and S. C. S. Chen, "Providing Unequal Reliability for Transmitting Layered Video Streams over Wireless Networks by Multi-ARQ Schemes," *Proc. of Int Conf. Image Processing*, vol 3, pp. 100-104, Oct. 1999.

[13]. T. S. Rappaport, *Wireless communications: Principles and Practice*, Prentice Hall, 1996.

[14]. J. Postel, "User Datagram Protocol," Request for Comments, RFC 768, ISI, Aug. 1980.

[15]. C. T. Wang, C. H. Fang and G. L. Chen, "Low-Delay and Error Robust Wireless Video Transmission for Video Communications," *IEEE Trans. Circuits Syst. Video Technol*, vol 12, pp. 1049-1058, Dec. 2002.

[16]. S. Dogan, A. Celatoglu, M. Uyguroglu, A. H. sadka and A. M. Kondoz, "Error-Resilient Video Transcoding for Robust Internetwork Communications Using GPRS," *IEEE Trans. Circuit Syst. Video Technol.*, vol 12, pp. 453-464, Jun. 2002.

[17]. B. Yan and K. W. Ng, “Mode-Based Error-Resilient Techniques for the Robust Communication of MPEG-4 Video,” *IEEE Trans. Circuits Syst. Video Technol*, vol 14, pp. 874-879, Jun. 2004.

[18]. H. Nguyen, J. Brouet and P. Duhamel, “Robust and Adaptive Transmission of Compressed Video Streams over EGPRS,” *Proc. of Consumer Communication and Networking Conf (CCNC).*, pp. 320-324, Jan. 2004.

[19]. B. Zheng and M. Atiquzzaman, “A Novel Scheme for Streaming Multimedia to Personal Wireless Handheld Devices,” *IEEE Trans. Consumer Electronics*, vol. 49, pp. 32-40, Feb. 2003.

[20]. D. P. Wu, T. Hou and Y. -Q. Zhang, “Scalable Video Coding and Transport over Broad-Band Wireless Networks,” *Proc. IEEE*, vol. 89, pp. 6-20, Jan. 2001.

[21]. D. P. Wu, T. Hou, W. Zhu, Y. -Q. Zhang and J. M. Peha, “Streaming Video over the Internet: Approaches and Directions,” *IEEE Trans. Circuits Syst. Video Technol*, vol 11, pp. 282-300, Mar. 2001.

[22]. H. Zheng, “Optimizing Wireless Multimedia Transmissions through Cross Layer Design,” *Proc. IEEE*, vol. 1, pp. 185-188, Baltimore, MD, Jul. 2003.

[23]. A. Ortega and K. Ramchandran, “Rate-Distortion Methods for Image and Video Compression,” *IEEE Signal Processing Mag.*, pp. 23-50, Nov. 1998.

[24]. C. E. Shannon, “A Mathematical Theory of Communication,” Bell Syst. Tech. J. 27, pp. 379-423 and 623-656, Jul. 1948.

[25]. T. Stockhammer, M. M. Hannuksela, and S. Wenger, “H.26L/JVT Coding Network Abstraction Layer and IP-based Transport,” *Proc. IEEE Int. Conf. Image Processing.*, vol. 2, pp. 485-488, Sept. 2002.

[26]. D. N. Dao, and W. A. C. Fernando, "Unequal Error Protection for H.26L Video Transmission," *Proc. Int. Symposium. Wireless Personal Multimedia Communications*, vol.2, pp. 821-825, Oct. 2002.

[27]. N. D. Dao, and W. A. C. Fernando, "Channel Coding for H.264 Video in Constant Bit Rate Transmission Context over 3G Mobile Systems," *Proc. Int. Symposium. Circuits and Systems (ISCAS)*, vol.2, pp. 896-899, May. 2003.

[28]. "Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios," ITU-R Recommendation BT.601-5 (10/95).

[29]. A. H. Sadaka, *Compressed Video Communications*, New York, Wiley, pp. 75-214, 2002.

[30]. M. Bystrom and T. Stockhammer, "Dependent Source and Channel Rate Allocation for Video Transmission," *IEEE Trans. Wireless Communication*, vol.3, pp. 258-268, Jan. 2004.

[31]. J. Mitchell, W. Pennebaker, C. Fogg and D. LeGall, *MPEG Video Compression Standard, London*, U.K.: chapman & Hall, 1996.

[32]. R. Talluri, I. Moccagatta, T. Nag, and G. Cheung, "Error Concealment by Data Partition," *Signal Processing: Image Commun.Mag.*, vol. 14, pp. 505-518, May. 1999.

[33]. G. Gote, 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, Jun. 2000.

[34]. Y. Takishima, M. Wada, and H. Murakami, "Reversible Variable Length Codes," *IEEE Trans. Commun.*, vol 43, pp. 158-162, Apr. 1995.

[35]. J. Wen and J. D. Villasensor, “Reversible Variable Length Codes for Efficient and Robust Image and Video Coding,” *Proc. Data Compression Conf.,* pp. 471-480, 1998.

[36]. D. W. Redmill, and N. G. Kingsbury, “The EREC: An Error-Resilient Technique for Coding Variable-Length Blocks of Data,” *IEEE Trans. Image Processing.,* vol 5, pp. 565-574, Apr. 1996.

[37]. A. A. El-Gamal, and T. M. Cover, “Achievable Rates for Multiple Descriptions,” *IEEE Trans. Inform. Theory,* vol. IT-28, pp. 851-857, 1982.

[38]. Y. Wang, and Q. F. Zhu, “Error Control and Concealment for Video Communications: A Review,” *Proc. IEEE,* vol. 86, pp. 974-997, May. 1998.

[39]. S. Wenger, T. Stockhammer, and M. M. Hannuksela, “RTP Payload Format for H.264 Video,” *in Internet Draft,* Draft-ietf-avt-rtp-h264-10.txt, Jul. 2004.

[40]. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC1889, IETF, 1996.

[41]. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC3550, IETF, Jul. 2003.

[42]. H. Schulzrinne, A. Rao, and R. Lanphier, “Real time streaming protocol RTSP,” RFC 2326, IETF, Apr. 1998.

[43]. T. Bova and T. Krivoruchk, “Reliable UDP Protocol,” Internet Draft, Network Working Group. [Online]. Available: draft-ietf-sigtran-reliable-udp-00.txt.

[44]. L. Larzon, M. Degermark, and S. Pink, “Efficient Use of Wireless Bandwidth for Multimedia Applications,” *Proc. MoMuc’99,* San Diego, CA, pp. 187–193, Nov. 1999.

[45]. H. T. Zheng and J. Boyee, “An Improved UDP Protocol for Video Transmission Over Internet-to-Wireless Networks,” *IEEE Trans. Multimedia*, vol. 3, pp. 356-365, Sept. 2001.

[46]. D. Lindberg, “The H.324 Multimedia Communication Standard,” *IEEE Commun. Mag.*, vol. 34, pp. 46-51, Dec. 1996.

[47]. “IP Multimedia Call Control Protocol Based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP),” 3GPP Technical Specification 3GPP TS 24.229, V6.5.1, Jan. 2005.

[48]. Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T. and H. Zheng, “RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed,” RFC 3095, Jul. 2001.

[49]. “Radio Link Control (RLC) protocol specification,” 3GPP Technical Specification 3GPP TS 25.322, V6.2.0, Dec. 2004.

[50]. H. W. Bai, “Error Modeling Schemes for Fading Channels in Wireless Communications: A Survey,” *IEEE Communication Surveys & Tutorials.*, vol. 5, pp. 2-9, 4th Quarter. 2003.

[51]. E. N. Gillbert, “Capacity of a Bursty-Noise Channel,” *Bell Syst. Tech. J.*, vol. 39, pp. 1253-1265, Sept. 1960.

[52]. E. O. Ellliott, “A Model of the Switched Telephone Network for Data Communications,” *Bell Syst. Tech. J.*, vol. 44, pp. 89-109, Jan. 1965.

[53]. Jr W.C.Jakes, Multipath interference, in Jr W. C. Jakes, editor, *Microwave Mobile Communications*, IEEE Piscataway, NY 1974.

[54]. M. Zorzi, R. R. Rao, and L. B. Milstein, “On the Accuracy of A First-Order Markov Model for Data Block Transmission on the Fading Channels,” *Proc. IEEE Int. Conf. Universal Personal Communications*, pp. 211-215, Nov. 1995.

[55]. P. Frossard, “FEC Performance in Multimedia Streaming,” *IEEE Commun. Lett.*, vol. 5, pp. 122-124, Mar. 2001.

[56]. P. Frossard, and O. Verscheure, “Joint Source/FEC Rate Selection for Quality-Optimal MPEG-2 Video Delivery,” *IEEE Trans. Image Processing*., vol. 10, pp. 1815-1825, Dec. 2001.

[57]. A. Majumdar, D. G. Sachs, I. V. Kozintsev, K. R. Ramchandran, and M. M. Yeung, “Multicast and Unicast Real-Time Video Streaming over Wireless LANs,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 524-534, Jun. 2002.

[58]. T-W. A. Lee, G. Chan, Q. Zhang, W. Zhu, and Y.-Q. Zhang, “Optimal Allocation of Packet-Level and Byte-Level FEC in Video Multicasting over Wired and Wireless Networks,” *Proc. IEEE Globecom.*, pp. 1994-1998, Nov. 2001.

[59]. Q. Zhang, W. Zhu, and Y.-Q. Zhang, “Network-Adaptive Scalable Video Streaming over 3G Wireless Network,” *Proc. IEEE Int. Conf. Image Processing*., vol. 3, pp. 579-582, Oct. 2001.

[60]. Q. Zhang, W. Zhu, and Y.-Q. Zhang, “Channel-Adaptive Resource Allocation for Scalable Video Transmssion over 3G Wireless Network,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, pp. 1049-1063, Aug. 2004.

[61]. T. Stockhammer, H. Jankac, and C. Weib, “Feedback and Error Protection Strategies for Wireless Progressive Video Transmission,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 465-482, Jun. 2002.

[62]. S. Aramvith, C. W. Lin, S. Roy, and M. T. Sun, “Wireless Video Transport Using Conditional Retransmission and Low-Delay Inerleaving,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 558-565, Jun. 2002.

[63]. R. Blahut, *Theory and Practice of Error Control Codes*, Reading, MA: Addison-Wesley, 1983.

[64]. I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *J. Soc. Indust. Appl. Math.*, Jun. 1960.

[65]. G. Roth, R. Sjöberg, G. Liebl, T. Stockhammer, V. Varsa, and M. Karczewicz, “Common Test Conditions for RTP/IP over 3GPP/3GPP2,” ITU-T SG16 Doc. VCEG-M77, Austin, TX, USA, May. 2001.

[66]. H.264/AVC Codec Software Archive. [Online]. Available: ftp://ftp.imtc-files.org/jvt-experts/reference_software

[67]. Y. Shan, A. Zakhor, “Cross Layer Techniques for Adaptive Video Streaming over Wireless Networks,” *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, pp.277-280, Lausanne, Switzerland, Aug. 2002.

[68]. H. Ma and M. El Zarki, “MPEG-2 Video Transmission over Wireless Access Networks using Type-I Hybrid ARQ Schemes for VoD Services,” *Proc. Packet Video Workshop’99*, Apr. 1999.

[69]. B. Girod and N. Farber, “Wireless Video,” a chapter in *Compressed Video over Networks*, Marcel Dekker 2001.

[70]. S. Krishnamachari, M. V. D. schaar, S. H. Choi, and X. F. Xu, "Video Streaming over Wireless LANs: A Cross Layer Approach," *Proc. Packet Video*, Nantes, France, Apr. 2003.

[71]. K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of Video Transmission over Lossy Channels," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1012-1032, Jun. 2000.

[72]. 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, Jun. 2000.

[73]. Z. H. He, J. F. Cai, and C. W. Chen, "Joint Source Channel Rate-Distortion Analysis for Adaptive Mode Selection and Rate Control in Wireless Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 511-523, Jun. 2002.

[74]. A. Chochalingam and G. Bao, "Performance of TCP/RLP Protocol Stack on Correlated Fading DS-CDMA Wireless Links," *IEEE Trans. Veh. Technol.*, vol. 49, pp. 28–33, Jan. 2000.

[75]. P. M. Soni and A. Chockalingam, "Performance Analysis of UDP with Energy Efficient Link Layer on Markov Fading Channels," *IEEE Trans. Wireless Commun.*, vol. 1, pp. 769–780, Oct. 2002.

[76]. Gary Sullivan, "Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low Resolution Progressive Scan Source Material," VCEG-N81, 14th meeting: Santa Barbara, USA. Sept. 2001.

[77]. C. Xu, T. M. Le, and T. T. Tay, "H.264/AVC Codec: Instruction Level Complexity Analysis," *Proc. IMSA*, Hawaii, Aug. 2005.