10

Error Resilient Video Coding using Cross-Layer Optimization Approach

Cheolhong An[1]*and Truong Q. Nguyen[2]
[1]*Qualcomm Incorporated*
[2]*University of California San Diego*
U.S.A.

1. Introduction

A communication system can be modeled as Open Systems Interconnection (OSI)-7 layers. Generally, each layer solves its own optimization problem. For example, video coding of the application layer minimizes distortion with or without considering transmission errors for given bit rate constraints. The Transmission Control Protocol (TCP) layer solves fair resource allocation problems given link capacities, and the Internet Protocol (IP) layer minimizes the path cost (e.g. minimum hop or link price). In the Media Access Control (MAC) layer, throughput is maximized for given bit error probability, and the Physical (PHY) layer minimizes the bit error probability or maximizes the bit rate (e.g. diversity and multiplexing of Multiple Input Multiple Output (MIMO)).

However, all the other layers except the application layer are not directly observed by end users. It means that the users evaluate a communication system based on quality of the application layer. Therefore, we need to maximize quality of the application layer cooperating with the other layers since one layer's optimization can affect performance of the other layers. In this chapter, we focus on trade-offs between rate and reliability for given information bit energy per noise power spectral density $\frac{E_b}{N_0}$ (i.e. Signal-to-Noise Ratio (SNR)) with consideration to error resilient video coding feature. Especially, the application oriented cross-layer optimization is considered for transmission of compressed video streams.

For the cross-layer optimization, the basic framework of Network Utility Maximization (NUM) (Kelly et al. (1998)) or extend framework of NUM (Chiang et al. (2007)) can be used. Especially, Lee et al. (2006) incorporate trade-offs between rate and reliability to the extend NUM framework. This framework is applied to decide the number of slices, source code rate, channel code rate, MAC frame length and channel time allocation for multiple access among

*Portions reprinted, with permission, from (C. An and T. Q. Nguyen, "**Resource Allocation for Error Resilient Video Coding over AWGN using Optimization Approach**", *the IEEE Transactions on Image Processing*, vol. 17 , pp. 2347-2355, Dec., 2008) ©[2008] IEEE, (C. An and T. Q. Nguyen, "**Resource Allocation for TDMA Video Communication over AWGN using Cross-Layer Optimization Approach**", *the IEEE Transactions on Multimedia*, vol. 10, pp. 1406-1418, Nov., 2008) ©[2008] IEEE, and (C. An and T. Q. Nguyen, "**Analysis of Utility Functions for Video**", in *Proceedings of the IEEE International Conference on Image Processing*, Sep., 2007) ©[2007] IEEE the utility functions. Figure 1 represents the procedure of cross-layer optimization with an objective function of the application layer. Trade-offs between rate and reliability of the other layers are mathematically modeled as constraint functions. Therefore, mathematical models to represent each layer's features are important. We formulate the end-to-end distortion function of error resilient video coding as a utility function and trade-offs relation between rate and reliability of the other layers as constraint functions. If mathematical models are available, an optimization problem can be formulated and convex property of the problem needs to be analyzed for convex optimization problem whose solution is the global optimum (Boyd & Vandenberghe (2004)).

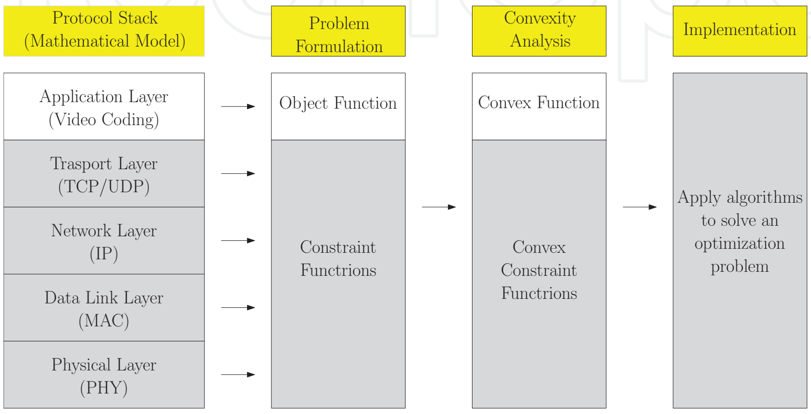


Fig. 1. Procedure of Application-Oriented (AO) cross-layer optimization.

For the convex optimization problem in (1), the objective function $f_0(x)$ must be concave for the maximization, inequality constraint functions $f_i(x)$ must be convex functions and equality functions $h_j(x)$ must be affine in the optimization problem (Boyd & Vandenberghe (2004)):

$$\begin{aligned} \max_{x} \quad & f_0(x) \\ s.t. \quad & f_i(x) \leq 0, \;\; h_j(x) = 0 \quad i,j = 1,...,p \end{aligned} \tag{1}$$

If the problem is not a convex optimization problem, it can be transformed into a convex optimization problem after transformation of optimization variables since convexity and concavity are not intrinsic features of a function. One of simple examples is geometric programming which is not a convex optimization problem. It can be transformed into a convex problem as in Boyd & Vandenberghe (2004). Chiang et al. (2007) showed several other techniques to decouple constraint functions and to transform optimization variables for convex optimization. If the problem is convex, Boyd & Vandenberghe (2004); Palomar & Chiang (2007) presented several algorithms that can be used to solve a convex optimization problem. In this chapter, the primal-dual decomposition method is applied to obtain optimal solutions using the Lagrangian dual decomposition and the (sub)gradient projection method in D.P.Bertsekar (2003); Johansson & Johansson (2005).

In this chapter, the NUM framework is applied to solve resource allocation problems for video communication with elaborate mathematical models and other optimization variables with Automatic Repeat reQuest (ARQ) and Time Division Multiple Access (TDMA). First, we are interested in allocating source code rate, channel code rate, MAC frame length and channel time allocation for multiple access among the utility functions. Many previous researches in Bystrom & Modestino (2000); Cheung & Zakhor (2000); Hochwald & Zeger (1997); K et al.

(2000); Z.He et al. (2002) only focused on finding optimal source code rate and channel code rate without considering MAC functionality such as ARQ and multiple access. Qiao & Choi (2001) adapted MAC frame size and modulation and channel code rate of the PHY layer for maximization of goodput without consideration of error effects of the application layer. Izzat et al. (2005) analyzed importance of MAC frame size to video streaming and Chou & Miao (2006); Haratcherev et al. (2005) addressed source code rate with delay constraint. Haratcherev et al. (2006) applied cross-layer signal method which directly signals the throughput of the link layer to video coding layer to reduce rate control time of video coder, but it does not consider error effects of video coding. In Kalman & Girod (2005), channel time allocation is performed experimentally between two utility functions.

Next, we further extend the framework with consideration of error resilient video coding, especially, the raster scan order picture segmentation known as a slice of a picture (An & Nguyen (2008a)). In H.264 (2009), a slice can contain any number of Macro Blocks (MBs) in a picture with raster scan order unless Flexible MB Ordering (FMO) is used. Segmentation of a picture is a common method for error resilient video coding without limitation of profile (Richardson (2003)). However, it is not well known regarding effects of multiple slices in video coding. Cote et al. (2000); Harmanci & Tekalp (2005) divided slices based on the Rate-Distortion (RD) optimization without considering network packetization and slice error probability. Masala et al. (2004) rearranged some portions of a slice to fit the network packet size in order to increase throughput. As a result, these methods can induce multiple slices loss from one packet loss. Wang et al. (2006) only showed channel-induced distortion with respect to (w.r.t.) slice error probability without considering video coding efficiency. Chiew et al. (2005) used the intra-refresh, multiple reference frames and sliced-coding to prevent error propagation using feedback information from decoder. In Wu & Boyce (2007), the redundant slice feature was proposed to replace corrupted primary slices. The proposed methods mainly focus on usage of slices without considering how many slices are adequate for given network status.

Here, we jointly optimize the number of slices with constraints of the MAC and PHY layers to answer a question how many slices of a picture are sufficient for error protection for given channel error probability. For this work, we analyze source coding efficiency w.r.t. the number of slices because it decreases the error probability of slice but increases the source-coded bit rate if a picture is composed of too many slices. Detail discussion on this trade-off will be presented in subsection 2.2 General guidelines were suggested in Wenger (2003) as follows: a coded slice size is as close as Maximum Transfer Unit (MTU) size of MAC but never bigger than MTU size. This constraint prevents fragmentation of the IP layer. Consequently, one MAC frame carries one IP datagram which contains one slice. It is reasonable to the wired MAC such as Ethernet. However, one MAC frame can be fragmented into smaller frames in order to increase reliability in the wireless MAC such as 802.11 (1999). Therefore, the constraint, that is, a coded slice is as close as MTU size does not prevent MAC fragmentation. In this chapter, we directly decide the optimal MAC frame size which is the length of the fragmented MAC frame. Then a slice is coded as close to the optimal MAC frame length as possible. These constraints are considered as a joint optimization problem with constraints from the PHY and MAC layers.

Previous works do not consider all the protocols from the application layer to the PHY layer. In this chapter, we build all the protocol stacks explicitly or implicitly. The application, the MAC and the PHY layers are explicitly formulated as an optimization problem with the number of slices, source code rate, channel code rate, MAC frame size and channel time allocation as optimization variables. IP, User Datagram Protocol (UDP) and Real-Time Transport Protocol (RTP) are implicitly considered as protocol overheads. The delay and buffering issues of video streaming are implicitly considered with assumption that maximum delay and jitter are guaranteed by the functionality of TDMA MAC. In order to consider RD characteristics of video sequences as well as distortion of channel errors, negative of end-to-end distortion is modeled as a utility function. In this chapter, we consider sum of utility functions as an objective function of an optimization problem for fair resource allocation within the same subscription policy.

2. Mathematical models of protocol layers

In this chapter, a communication system with Additive White Gaussian Noise (AWGN) channel in Figure 2 is considered and formulated as an optimization problem. We jointly optimize three layers of the protocol stack: the application, the data link and the physical layers to decide the optimal number of slices, source code rate, channel code rate, MAC frame size and channel time allocation for a given SNR. The data link layer uses 802.11a-like MAC with Automatic Repeat reQuest (ARQ) and Time Division Multiple Access (TDMA). Acknowledge packets (ACKs) are assumed to be received without any errors because the length of packets is relatively short. In the PHY layer, Binary Phase Shift Keying (BPSK) modulation and high resolution soft decision demodulation are assumed with the perfect bit interleave and deinterleave. For Forward Error Correction (FEC), Rate Compatible Punctured Convolutional code (RCPC) (Hagenauer (1988)) and Viterbi decoder are used. In the application layer, H.264 video encoder and decoder are considered with error resilient video coding. For error resilient video coding, multiple slice coding is used, and previous decoded frame is considered for simple error concealment. We model negative of Mean Square Error (MSE) $-E[(X-\tilde{X})^2]$[1] as a utility function, because the end-to-end distortion $E[(X-\tilde{X})^2]$ is generally used as an objective measure to evaluate quality in video compression. In order to circumvent delay and jitter issues of video streaming, TDMA method instead of Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is considered. For multiple access, joint optimization among coordinator and utility functions is performed distributively for the optimal channel time allocation in a coordinated network.

2.1 Analysis of Utility function for Video

video coding is considered for the application layer. The framework of NUM has the maximization of an objective function. In order to match this framework, the maximization of $-E[(X-\tilde{X})^2]$ is equivalent to the minimization of the end-to-end distortion $E[(X-\tilde{X})^2]$ which induces the maximization of Peak Signal-to-Noise Ratio (PSNR)[2]. Let the end-to-end distortion D_t be $E[(X-\tilde{X})^2]$. D_t in (2) can be decomposed into source-induced distortion D_e and channel-induced distortion D_c with assumption that quantization errors and channel errors are uncorrelated with zero mean. Z.He et al. (2002) showed experimentally that they

[1] Original samples X are input data of the video encoder in the transmit side, and reconstructed samples $\tilde{X}$ are output data of the video decoder in the receiver side which are shown in Figure 2.

[2] $PSNR = 10\log_{10}\frac{255^2}{D_t}$ where $D_t = E[(X-\tilde{X})^2]$

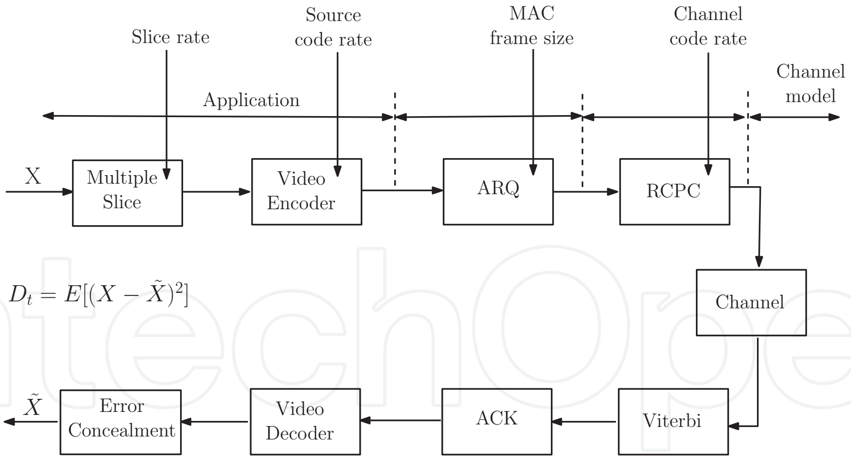


Fig. 2. A communication system model over AWGN.

are uncorrelated:

$$\begin{aligned} D_t &= E[(X-\tilde{X})^2] = E[(X-\hat{X}+\hat{X}-\tilde{X})^2] \\ &= E[(X-\hat{X})^2] + E[(\hat{X}-\tilde{X})^2] + 2E[(X-\hat{X})(\hat{X}-\tilde{X})] \\ &\approx E[(X-\hat{X})^2] + E[(\hat{X}-\tilde{X})^2] = D_e + D_c \end{aligned} \tag{2}$$

where X are original samples, $\hat{X}$ are reconstructed samples at the encoder, and $\tilde{X}$ are reconstructed samples at the decoder. In An & Nguyen (2007), we analyzed utility functions for video. From the information theory, a D-R model [3] in (3) is induced from the Independent Identically Distributed (IID) gaussian process with variance σ^2 in Taubman & Marcellin (2002)

$$D_e(R) = \sigma^2 2^{-2R} \tag{3}$$

where R is the source bit per pixel, and σ^2 is variance of a Discrete Cosine Transform (DCT) coefficient. According to different distributions and quantization methods, the above D-R model can be generalized into (4) by Taubman & Marcellin (2002)

$$D_e(R) = \epsilon^2\sigma^2 2^{-2R} = \beta e^{-\alpha R} \ \ (\beta, \alpha > 0) \tag{4}$$

where $\epsilon^2 \approx 1.2$ for the Laplacian distribution. It is generally well known that a D-R model (4) only matches well with experimental results in a high bit rate region. A P-R function $PSNR(R)$ from (4) makes it clear, since $PSNR(R)$ has a linear relation with R as follows :

$$\begin{aligned} PSNR(R) &= 10\log_{10}\frac{255^2}{D_e} \\ &= 10log_{10}\frac{255^2}{\beta e^{-\alpha R}} = a_1 R + a_2 \end{aligned} \tag{5}$$

[3] we use the bit per pixel R instead of the bit per second x_s without index s of each source for the simplicity : $R_s = \frac{x_s}{f_r \times f_w \times f_h}$ where f_r is the number of frames per second and $f_w \times f_h$ is the number of samples per frame.

Figure 3 shows that the linear model (5) does not match well with the experimental $PSNR(R)$ which is highly nonlinear especially in a low bit rate region. Moreover, the video quality of many applications is between 28dB and 45dB which is a highly nonlinear area.

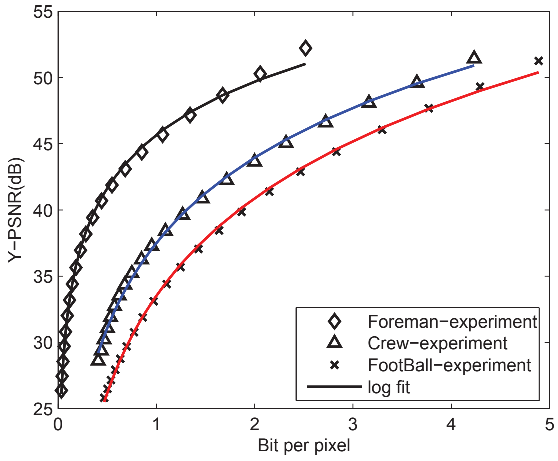


Fig. 3. PSNR vs. bpp for video sequences from An & Nguyen (2008a) (©[2008] IEEE).

A D-R model (6) is an variation of (4) shown in Wu et al. (2006)

$$D_e(R) = \beta e^{-\alpha R^{\gamma}} \quad (\beta > 0, \ \ 0 < \gamma, \alpha < 1) \tag{6}$$

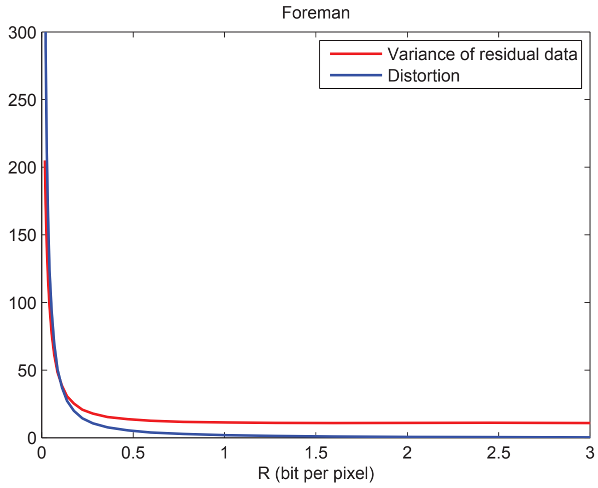


Fig. 4. Variance and distortion with respect to bit per pixel.

The main reason of the mismatch between mathematical models and experimental results is that the distortion in the mathematical models is obtained for a given variance of DCT coefficients. In the image compression techniques such as JPEG and JPEG2000 Taubman & Marcellin (2002), input data of a quantizer are DCT coefficients of natural image pixels. Therefore, variance of input data does not depend on the quantization step size. However, in the video coding techniques such as H.264 (2009), residual data of a current frame, which are difference between original samples and predicted samples from inter or intra prediction, are transformed and quantized. Inter or intra predicted samples are obtained from the neighboring reconstructed samples or previous decoded picture which the sum of predicted samples and quantized residual data. Therefore, residual data have different variance according to the quantization step size which controls bit per pixel R as shown in Figure 4. k. Jain (1989) showed that variance of residual data is highly correlated to variance of DCT coefficients. Consequently, variance of residual data relates to distortion as shown in Figure 4. In a high bit rate region, variance of residual is almost same such that experimental results match well with (4) but in a low rate region variance changes rapidly such that the mathematical models are different from experimental results. Therefore, input variance of a quantizer changes with respect to R such that a D-R model (4) needs to be modified as follows:

$$\begin{aligned} D_e(R) &= \epsilon^2\sigma^2(R)e^{-\alpha R} = \epsilon^2(a_1 e^{-a_2 R} + a_3)e^{-\alpha R} \\ &= ae^{-bR} + ce^{-dR} \ \ (a, b, c, d > 0) \end{aligned} \tag{7}$$

PSNR can be considered as a utility function in addition to the distortion. Figure 3 shows that the linear model (5) does not match well with the experimental $PSNR(R)$ of H.264 reference software model (JM (2007)) since it is highly nonlinear especially in the low bit rate region. Therefore, we propose $PSNR(R)$ and its distortion in reference An & Nguyen (2007) as follows:

$$PSNR(R) = m_1 \log(R) + m_2 \ \ (m_1, m_2 > 0) \tag{8}$$

$$D_e = a_1 x^{-a_2} \ \ (a_1, a_2 > 0) \tag{9}$$

Figure 3 represents that (8) matches well with experimental results of H.264 reference software model (JM (2007)) which is configured with high complexity rate distortion optimization, Universal Variable Length Coding (UVLC), 7 B frames and fast motion search of Enhanced Predictive Zonal Search (EPZS) at main profile. The proposed distortion model is a convex function w.r.t. source code rate x and $\log x$ which is necessary for convex optimization in transform domain. If an original problem is not convex, the problem can be transformed into a convex optimization problem after transformation of optimization variables.

A P-R model (8) is fitted to experimental results (H.264 reference software model JM (2007) is used for this experiment) as shown in Figure 3. They match well with experimental results in the usual operating bit rate $(R < 2)$. A D-R model (10), which is induced from (8), is

$$D_e(R) = hR^{-j} \ \ (h, j > 0) \tag{10}$$

Channel-induced distortion D_c is generated from channel errors because if there are no errors during the transmission, D_c vanishes since reconstructed samples $\tilde{X}$ in the video decoder is equal to reconstructed samples $\hat{X}$ in the video encoder. Wang et al. (2006) proposed that the channel-induced distortion D_c is

$$D_c = \frac{p}{I_r(1-p)} D_{ECP} \tag{11}$$

where D_{ECP}, I_r and p denote average distortion after error concealment, average intra Macro Block (MB) ratio in a picture and slice error probability of one picture, respectively. Wang et al. (2006) claimed that (11) is approximately valid with sub-pixel motion vectors, deblocking filtering and constrained intra prediction of H.264 (2009). Each video frame is coded as $\frac{x}{V_r}$ bits where V_r is video frame rate. Therefore, slice error probability p of a video frame comprising of one slice is

$$p = 1 - \left(1 - \frac{P_{fr}^{d_{max}}}{8(L - h_{ov})}\right)^{\frac{x}{V_r}} \tag{12}$$

$$\approx \frac{x}{V_r} \cdot \frac{P_{fr}^{d_{max}}}{8(L - h_{ov})} \tag{13}$$

after considering average bit error probability of ARQ in the MAC layer $P_{fr}^{d_{max}}$, which will be explained in subsection 2.5 Here, we assume $\frac{P_{fr}^{d_{max}}}{8(L-h_{ov})} \ll 1$ for an convex optimization problem. This assumption is usually satisfied with ARQ function (d_{max}), adequate frame error probability and MAC frame size.

2.2 Application layer with error resilient video coding

There are many methods to protect a coded video stream in H.264 (Richardson (2003); Wenger (2003)). For example, slice grouping which allocates MBs to a slice group by a slice group map is known as FMO, and redundant slices carry the same MBs with different quality in the base profile of H.264. In the extended profile, one slice can be separated into three partitions according to the importance of MBs, and each partition can be decoded independently. The most basic method for error resilient video coding without limitation of profiles is a picture segment which is known as a slice. A slice consists of any number of MBs in raster scan order within a picture, that is, a slice can include from one MB to maximum MBs of a picture. It also means that a picture is composed of a single slice or multiple slices. The main purpose of the picture segmentation is to enable independent decoding of slices because each slice has its own start code, and it is separately coded. Therefore, loss of some slices of a picture does not affect decoding of the other slices. Consequently, some portions of a picture could still be reconstructed, and they can be used for error concealment of loss parts in a picture. However, Motion Vector (MV) prediction, intra prediction, MB mode prediction and context of entropy coding are restricted for independent decoding which will be discussed later in detail. As a result, multiple slices of a picture reduce video coding efficiency with error resiliency increased.

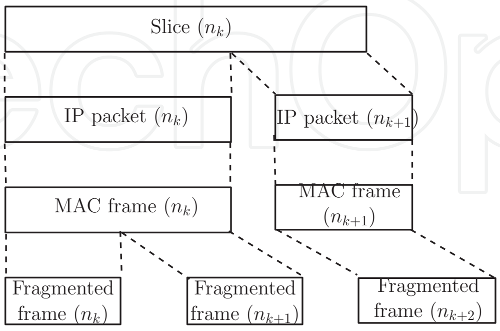


Fig. 5. IP layer fragmentation because a slice length is larger than MTU.

Thus, an essential issue of picture segmentation is how to decide the number of slices of a picture since it has a trade-off between a source-coded rate and error probability of a slice. If the number of slices increases, a source-coded rate increases, but error probability of a slice decreases because coded bits of a slice decrease. However, error probability of a slice is also

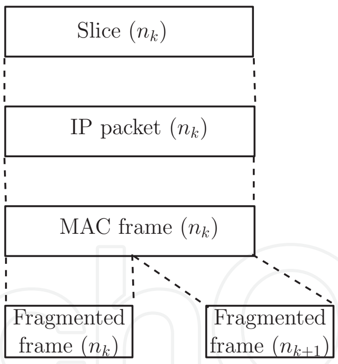


Fig. 6. MAC layer fragmentation even though a slice length is smaller than MTU.

highly related with network packetization. Figure 5 shows that if a slice length is larger than MTU, the IP layer fragments a slice into several IP packets. Thus, error probability of one slice increases because any loss of IP packets induces loss of a slice. Even if a slice length is smaller than MTU, the slice can be partitioned into smaller MAC frames due to the MAC fragmentation in wireless environment which is illustrated in Figure 6. Therefore, we jointly optimize a MAC frame length and slice length with source and channel-coded rate in order to satisfy the following constraint.

Constraint 1: A coded slice is as close as the optimal MAC frame size so that there is no fragmentation of the MAC frame, since the optimal MAC frame is optimal to minimize end-to-end distortion.

Equation (11) is applied to quantify distortion from error probability of a slice, and source-induced distortion D_e is derived from (9). However, (9) does not consider effects of the number of slices, that is, it is modeled as one slice per video frame.

Here, we analyze effects of the number of slices to the source-induced distortion and its bit rate. H.264 reference software model JM (2007) can segment a picture based on the number of MBs or the number of byte. If we choose the former option, each slice can be coded by various bits. Therefore, a picture is segmented by the number of byte to satisfy the Constraint 1. Three parts are mainly affected from multiple slice coding.

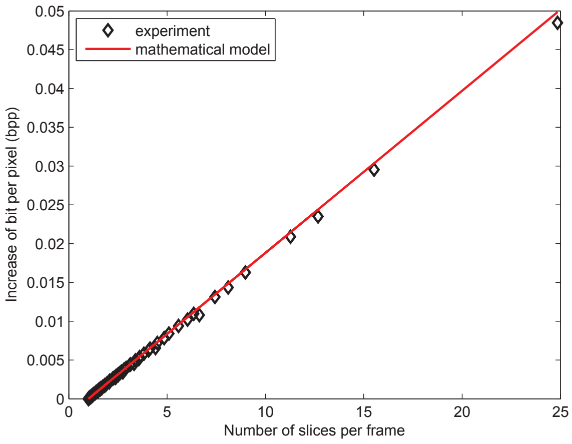


Fig. 7. Slice header bits increase w.r.t. the number of slices from An & Nguyen (2008a) (©[2008] IEEE).

First, coded bits for slice header information increase along with the number of slices because every slice of a picture is needed to be decoded independently. Figure 7 shows that slice header bits increase w.r.t. the number of slices at each different bit rate. Thus, increments of slice header bits X_{SH} do not depend on coded bit rates, but rather the number of slices. Therefore, it is modeled as

$$X_{SH} = \kappa_1(n-1) \tag{14}$$

where κ_1 is a positive constant and n is the number of slices of a picture. If $n = 1$, there is no increase of slice header bits. Figure 7 illustrates that equation (14) matches well with the experimental result.

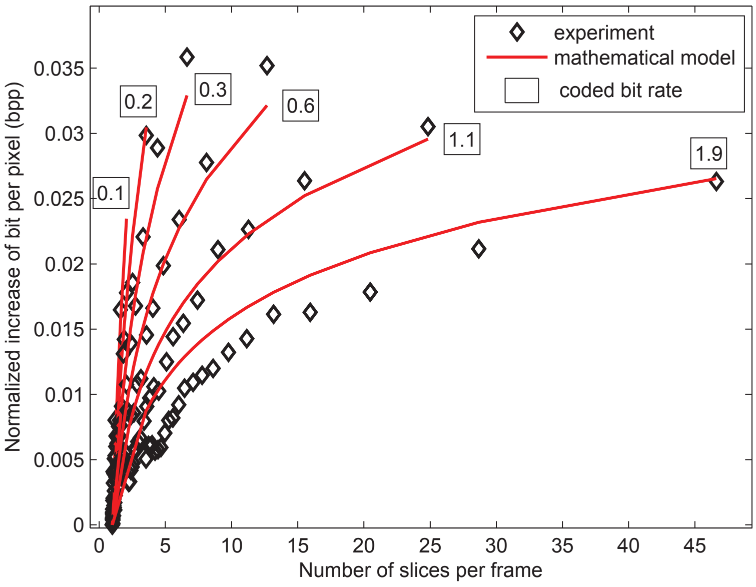


Fig. 8. Sum of other bits increase w.r.t. the number of slices at different bit per pixels from An & Nguyen (2008a) (©[2008] IEEE).

Second, MV prediction, intra prediction, MB mode prediction and context of entropy coding are restricted for independent decoding. If a current MB is at the boundary of different slices, neighboring MBs of the current MB are not available for MV prediction, intra prediction, MB mode prediction and context of entropy coding. Therefore, they increase coded bits for MVs, luminance Y, color residual C and MB mode. Figure 8 illustrates that overall normalized bits except slice header bits increase differently according to both the number of slices and coded bit rates. Furthermore, Figure 8 suggests out that effect of picture segmentation is larger at low bit rates. Consequently, we model bit increments from the restriction of prediction as follows:

$$X_{OT} = \kappa_2\sqrt{x}\log(n) \tag{15}$$

where κ_2 is a positive constant, n is the number of slices of a picture and x is a source-coded bit rate. For $n = 1$, there are no bit increments. The mathematical model and experimental results are shown in Figure 8.

Last, the restriction of MV prediction, intra prediction, MB mode prediction and context of entropy coding induce different coded bits and motion-compensated prediction errors. Thus, RD optimization of H.264 in (16) can choose different MB modes from MB modes of one sliced picture. It affects both coded bits $Z(\mathbf{m})$ and distortion $D(\mathbf{m})$ from the RD optimization:

$$\min_{\mathbf{m}} D(\mathbf{m}) + \lambda(QP) \cdot Z(\mathbf{m}), \quad \mathbf{m} = (MV, Mode) \tag{16}$$

$$\lambda(QP) = \xi 2^{\frac{(QP-12)}{3}} \tag{17}$$

where ξ and λ are a positive constant and the Lagrange multiplier, respectively. Vector $\mathbf{m}$ contains optimization variables which consist of MV and MB modes. The effects on bit rates are already reflected in (14) and (15). Here, an variation of distortion is discussed. Figure 9 shows variations of PSNR and coded bits according to the number of slices at each bit per pixel (bpp). Although PSNR does not change, bpp increases w.r.t. the number of slices as shown in Figure 9. It results from the fact that the RD optimization of H.264 helps to reduce distortion at the high bit rate region. From the RD optimization (16) and the relation (17) between λ and Quantization Parameter (QP) in Weigand et al. (2003), λ is smaller at high bit rates (small QP) which means that the RD optimization tries to minimize more distortion $D(\mathbf{m})$ than coded bits $Z(\mathbf{m})$. Therefore, the variation of distortion can be diminished. On the contrary, the RD optimization increases distortion at the low bit rate region, but distortion does not change significantly since the distortion of low bit rates is already large and the number of slices is small. From the above results, we assume that the number of slices does not affect source-induced distortion but rather source-coded rates. The bit increments X_{SL} are modeled as the sum of (14) and (15).

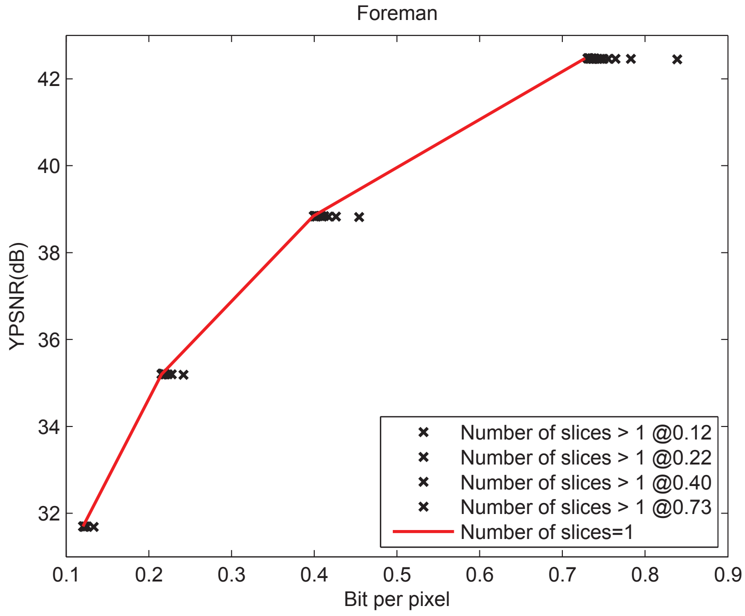


Fig. 9. PSNR vs. the number of slices at each bit per pixel from An & Nguyen (2008a) (©[2008] IEEE).

2.3 Objective function of network utility maximization problem

In this chapter, we use the sum of negative of distortion functions as an objective function for maximization. We may consider the sum of PSNR functions as an objective function. However, these optimization problems have quite different solutions. If we only consider a single utility function as an objective function, the solution of maximization of a PSNR function is equal to the solution of minimization of a distortion function:

$$\arg\max_{x} PSNR(x) = \arg\min_{x} D(x)$$

However, the solution of maximization of the sum of PSNR functions is different from the solution of minimization of the sum of distortion functions:

$$\begin{aligned}\arg\max_{\mathbf{x}} -\sum_s D_s(x_s) &= \arg\min_{\mathbf{x}} \sum_s D_s(x_s)\\ \neq \arg\max_{\mathbf{x}} \sum_s PSNR_s(x_s) &= \arg\max_{\mathbf{x}} \sum_s 10\log\frac{255^2}{D_s(x_s)}\\ = \arg\min_{\mathbf{x}} \sum_s \log D_s(x_s) &= \arg\min_{\mathbf{x}} \prod_s D_s(x_s) \qquad (18)\end{aligned}$$

Moreover, if we compare average PSNR between two methods, the optimal PSNR (average PSNR) value from solution of the sum of distortion is always smaller than the optimal PSNR value from solution of the sum of PSNR. The average PSNR can be calculated by two definitions.

First, if we find the optimal solution $\mathbf{x}_{\mathbf{d}}^{*}$ of the sum of distortion, we can calculate PSNR of each utility function and then average these values. However, this average PSNR ($\frac{1}{N}\sum_{s=1}^{N} PSNR_s$) is always smaller than average PSNR from the optimal solution $\mathbf{x}_{\mathbf{p}}^{*}$ of the sum of PSNR because we solve a convex optimization problem and the solution is global optimal solution, thus any other solutions such as $\mathbf{x}_{\mathbf{d}}^{*}$ can not achieve larger sum of PSNR (average PSNR) than $\mathbf{x}_{\mathbf{p}}^{*}$.

Second, if we define average distortion[4] and its PSNR[5] as notes, the average PSNR value $PSNR_d$ is less than or equal to average PSNR which is proved as follows:

$$\begin{aligned}PSNR_d &= 10\log_{10}\frac{255^2}{\frac{1}{N}\sum_s D_s(x_s)}\\ &= 10\log_{10}255^2 - 10\log_{10}\frac{1}{N}\sum_s D_s(x_s)\\ &\leq 10\log_{10}255^2 - \frac{10}{N}\sum_s \log_{10} D_s(x_s)\\ &= \frac{1}{N}\sum_s \Big(10\log_{10}255^2 - 10\log_{10} D_s(x_s)\Big)\\ &= \frac{1}{N}\sum_s 10\log_{10}\frac{255^2}{D_s(x_s)} = \frac{1}{N}\sum_s PSNR_s\end{aligned}$$

It means that even though we achieve the minimum sum of distortion (average distortion), PSNR of the average distortion is smaller.

Here, we show one example. From Table 1, there are two utility functions and three configurations, and configuration 2 is current distortion and PSNR of two utility functions. If we reallocate bits to maximize sum of PSNR functions or minimize sum of distortion

[4] Average distortion$D_{avg} \triangleq \frac{1}{N}\sum_{s=1}^{N} D_s$, where N is the number of utility functions.

[5] Average PSNR $PSNR_d \triangleq 10\log_{10}\frac{255^2}{D_{avg}}$.

functions, the distortion of U_0 varies slightly and the distortion of U_1 varies significantly because U_0 is currently operating at low distortion and U_1 operates at high distortion. The convex property of a distortion function induces different variation of distortion according to reallocation of bits. If we reduce bits of U_1 and reallocate the bits to U_0, the distortion and its PSNR of two utility functions change from configuration 2 to configuration 1. The other case changes from configuration 2 to configuration 3.

Table 1 shows that if we reallocate bits to maximize sum of PSNR, we should reallocate bits of two utility functions for configuration 1 and if we decide bits of two utility functions to minimize sum of distortion, we should choose the solution of configuration 3. This result matches with the result of (18), because the maximization of the sum of PSNR functions is equivalent to the minimization of multiplication of distortion. The multiplication of distortion are 300, 400 and 450 from configuration 1 to 3. Thus, configuration 1 has the minimum multiplication of distortion which corresponds to the maximum of sum of PSNR.

Which solution is better? Generally, configuration 3 is better than the others for fair resource allocation since variation of PSNR and distortion between utility functions decreases, even though average PSNR is smaller. Consequently, we use the sum of distortion functions as an objective function of NUM problem, even if average PSNR is smaller than average PSNR obtained from the solution of maximization of the sum of PSNR.

Utility function	Config 1	Config 2	Config 3
U_0	5 (41)[a]	10 (38)	15 (36)
U_1	60 (30)	40 (32)	30 (33)
$U_0 + U_1$	65 (71)	50 (70)	45 (69)

[a] Distortion (PSNR[dB])

Table 1. Example of max. PSNR vs. min. Distortion.

2.4 Physical layer model

After Viterbi decoding, Lin & Costello (2004) showed that bit error probability P_b of a binary-input and continuous-output AWGN channel is bounded by

$$\begin{aligned} P_b &< \sum_{d=d_{free}}^{\infty} B_d Q\left(\sqrt{\frac{2drE_b}{N_0}}\right) \\ &< \sum_{d=d_{free}}^{\infty} B_d e^{-\frac{drE_b}{N_0}},\ (Q(x) < e^{-\frac{x^2}{2}}) \qquad (19) \\ &\approx B_{d_{free}} e^{-\frac{d_{free} r E_b}{N_0}} \qquad (20) \end{aligned}$$

where B_d is the total number of nonzero information bits on all weight-d paths, d_{free} is the free distance of convolutional code, and r is the channel code rate. Equation (20) is derived with the assumption of dominance of first term in (19) at large $\frac{E_b}{N_0}$. However, it is difficult to use these equations as a mathematical model because of the two main reasons, that is, convexity and dependency of parameters B_d and d_{free} with respect to channel code rate r.

First, Figure 10 illustrates the experimental results of RCPC (Hagenauer (1988)) for memory $M = 6$ to show the variation of P_b according to the number of paths d from d_{free} to $d_{free} + 6$.

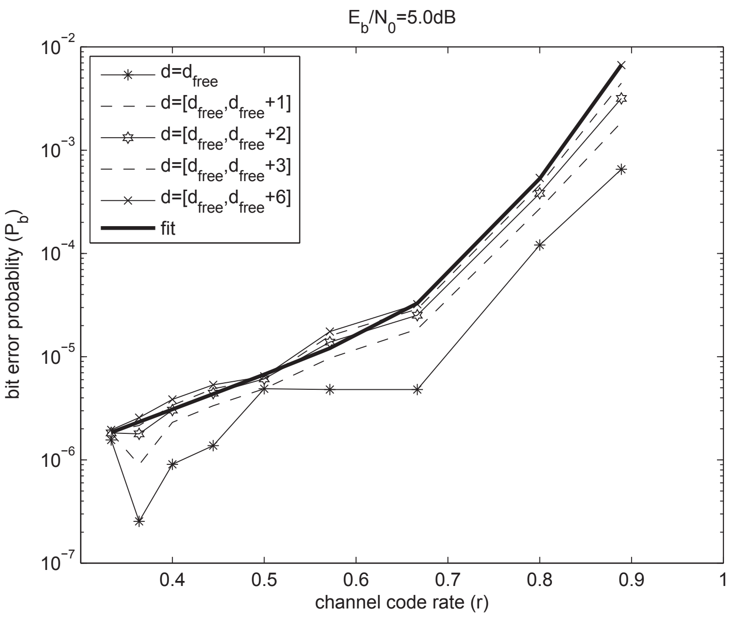


Fig. 10. Bit error probability P_b at $\frac{E_b}{N_0} = 5dB$ from An & Nguyen (2008b) (©[2008] IEEE)

Figure 10 shows that the first term of (19), that is, equation (20) is not dominant at low SNR. From Figure 10, we recognize that P_b is not a convex function of r, especially at high SNR, but it can be mildly assumed to be a convex function at low SNR. This convexity is necessary for the convex optimization which will be discussed in section 3. So we propose a convex model for the bit error probability P_b with respect to r given $\frac{E_b}{N_0}$ and M as follows:

$$P_b = p_1 e^{p_2 r} + p_3 e^{p_4 r} \tag{21}$$

where p_1, p_2, p_3 and p_4 are positive variables which depend on $\frac{E_b}{N_0}$. Equation (21) is a convex-hull mapping of equation (19) where B_d and d_{free} are obtained from Hagenauer (1988). Figure 10 illustrates that the proposed model is close to the experimental results of Hagenauer (1988). However, Figure 10 shows that some R values are not convex hull points but the difference between convex hull points (module curve) and experimental values are not large and mathematical P_b is also lower bound experimental values. Thus, this convexity model is only adequate at low SNR region. Therefore, we confine that SNR is lower than or equal to 7dB.

2.5 MAC layer model

For a MAC frame length of L Bytes, the error probability of a MAC frame P_{fr} is

$$\begin{aligned} P_{fr} &\leq 1 - (1 - P_b)^{8L} \approx 8LP_b, \quad (P_b \ll 1) \\ &= 8L\left(p_1 e^{p_2 r} + p_3 e^{p_4 r}\right) \end{aligned} \tag{22}$$

where (22) is derived with assumption of low bit error probability. If errors occur during the transmission of a MAC frame, the MAC frame is retransmitted up to the number of the maximum retransmission d_{max}. Therefore, the average time T_{avg} to transmit successfully one

MAC frame is

$$T_{avg} = \sum_{i=1}^{d_{max}} P(n=i)\Big((i-1)\tau_f + \tau_s\Big) \tag{23}$$

where $P(n = i)$ is the success probability at i-th times transmission. τ_f and τ_s are one transaction time (time duration from transmitting a data frame to receiving its ACK packet) of fail and success, respectively. In this chapter, we assume $\tau_f = \tau_s = \tau$ since they are almost same in TDMA media access method because given a time slot, the medium can be accessed without waiting. In contrast, CSMA/CA of 802.11 (1999) increases the contention window to two times, if there is an error. Therefore, τ_f usually becomes longer according to the number of retransmission (802.11 (1999)). There is some possibility of not receiving ACK packets which can induce longer waiting time, but we assume that ACK packets are always received without errors because the length of packets is relatively short. Consequently, the average transmission time of one MAC frame T_{avg} is

$$\begin{aligned} T_{avg} &= (1-P_{fr})\tau\Big(1 + 2P_{fr} + 3P_{fr}^2 + ... + d_{max}P_{fr}^{d_{max}-1}\Big) \\ &= (1-P_{fr})\tau\left(\frac{1-P_{fr}^{d_{max}}}{(1-P_{fr})^2} - \frac{d_{max}P_{fr}^{d_{max}}}{1-P_{fr}}\right) \\ &= \tau\left(\frac{1-P_{fr}^{d_{max}}}{1-P_{fr}} - d_{max}P_{fr}^{d_{max}}\right) \approx \frac{\tau}{1-P_{fr}}, \quad (P_{fr} \ll 1) \end{aligned} \tag{24}$$

One MAC frame, which has $8L$ bits length, can be transmitted successfully during T_{avg}. Consequently, the average goodput x_{gp} (application layer throughput excluding protocol overheads, retransmitted data packets and so on) is

$$x_{gp} = \frac{8(L-h_{ov})}{T_{avg}} = \frac{8(L-h_{ov})(1-P_{fr})}{\tau} \tag{25}$$

Here, h_{ov} is overhead of a MAC frame including the MAC header, Frame Check Sequence (FCS), service information and tail shown in Figure 11, as well as other protocols (e.g. Internet Protocol). If transmission of one MAC frame fails up to the number of the maximum retransmission d_{max}, the error probability of one MAC frame is $P_{fr}^{d_{max}}$, and one MAC frame encapsulates $8(L-h_{ov})$ of application layer bits. Therefore, the average bit error probability of the application layer after ARQ is $\frac{P_{fr}^{d_{max}}}{8(L-h_{ov})}$.

In this chapter, 802.11 (1999) MAC and 802.11a (1999) PHY are considered for our mathematical model. The transaction time τ shown in Figure 11 is

$$\begin{aligned} \tau &= T_{Preamble} + T_{Sig} + \frac{8LT_{symbol}}{N_{SD} r \log_2 M_o} + 2SIFS + \tau_{ack} \\ &= \frac{1}{N_{SD}}\left(A_0 + \frac{8LT_{symbol}}{r\log_2 M_o} + \frac{8L_{ack}T_{symbol}}{r\log_2 M_o}\right) \end{aligned}$$

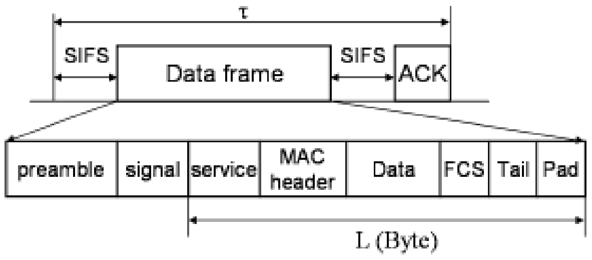


Fig. 11. One transaction and MAC frame structure from An & Nguyen (2008b) (©[2008] IEEE).

where

$$\tau_{ack} = T_{Preamble} + T_{Sig} + \frac{8L_{ack}T_{symbol}}{N_{SD} r \log_2 M_o},$$
$$A_0 = 2N_{SD}(T_{Preamble} + T_{Sig} + SIFS)$$

Here, N_{SD} is the number of Orthogonal Frequency Division Multiplexing (OFDM) data subcarriers. $T_{Preamble}$, T_{Sig} and Short InterFrame Space (SIFS) are the preamble time, the signal time and the short time space between frames, respectively. T_{symbol} is OFDM symbol time. L_{ack} and M_o represent the length of an ACK packet and the M_o-ary modulation, respectively. Consequently, the goodput x_{gp} is

$$x_{gp} = N_{SD} \frac{8(L - h_{ov})(1 - P_{fr})}{A_0 + \frac{8LT_{symbol}}{r \log_2 M_o} + \frac{8L_{ack}T_{symbol}}{r \log_2 M_o}} \tag{26}$$

For TDMA, contention-free period T_{pcf} is divided into some amount of time T_s for each source, i.e., $\sum_s T_s = T_{pcf} - B$ where B is beacon time. During T_s, each source executes transactions with its own source code rate x_s, channel code rate r_s and frame length L_s. Therefore, each source can achieve different error probability of a MAC frame P^s_{fr} and its goodput x^s_{gp}. $T_{pcf} = T_{RI} - T_{dcf}$, where T_{RI} is the repetition interval and T_{dcf} is the contention period. Finally, each source's goodput x^s_{gp} is formulated as follows:

$$x^s_{gp} = t_s \cdot N_{SD} \frac{8(L_s - h_{ov})(1 - P^s_{fr})}{A_0 + \frac{8L_s T_{symbol}}{r_s \log_2 M^s_o} + \frac{8L_{ack}T_{symbol}}{r_s \log_2 M^s_o}} \tag{27}$$

where $t_s = \frac{T_s}{T_{RI} - T_{dcf} - B}$, $\sum_s t_s = 1$ and $t_s \geq 0$.

3. Problem formulation

In this section, we only consider a Basic Service Set (BSS) which consists of a set of nodes controlled by a single coordination function (one node which is named as a coordinator in a BSS performs this function), and we assume that each node in a BSS can transmit directly to its destination node. Note that in 802.11 (1999), all transactions have to pass through Access Point (AP) to reach their destination nodes. However, 802.11e (2005) allows each node to exchange frames directly through the direct link. Therefore, the number of links to reach the destination is only one for each source. For simplicity, a link index for each source is omitted. In reference

An & Nguyen (2008b), we formulated a cross-layer optimization problem with one slice per picture as follows using mathematical models in section 2:

$$\max_{\mathbf{x},\mathbf{p},\mathbf{L},\mathbf{r},\mathbf{t}} \quad -\sum_s \left(a_1 x_s^{-a_2} + \frac{p_s}{I_r(1-p_s)} D_{ECP} \right) \tag{28}$$

$$s.t. \quad \frac{x_s}{V_r} \frac{\left[8L_s(p_1 e^{p_2 r_s} + p_3 e^{p_4 r_s}) \right]^{d_{max}}}{8(L_s - h_{ov})} \le p_s \tag{29}$$

$$x_s \le t_s \cdot N_{SD} \frac{8(L_s - h_{ov})(1 - P^s_{fr})}{A_0 + \frac{8L_s T_{symbol}}{r_s \log_2 M_o^s} + \frac{8L_{ack} T_{symbol}}{r_s \log_2 M_o^s}} \tag{30}$$

$$x_s^{min} \le x_s \le x_s^{max}, \ r_s^{min} \le r_s \le r_s^{max}$$

$$L_s^{min} \le L_s \le L_s^{max}, \ p_s^{min} \le p_s \le p_s^{max}$$

$$\sum_s t_s \le 1, \ t_s \ge 0, \ \forall s$$

where $P^s_{fr} = 8L_s(p_1 e^{p_2 r_s} + p_3 e^{p_4 r_s})$ and $s \in S$ and S is a set of utility functions which transmit their video streams. Utility functions $U_s(x_s, p_s)$ in (31) are the negative end-to-end distortion D_t in (28) which was discussed in section 2. The constraint (29) is relaxed from the equality constraint of slice error probability of one video frame comprising one slice. In case of one slice of a picture, slice error function $P_s(x_s, L_s, r_s)$ of (32) is derived in (29) as $\frac{x_s}{V_r} \frac{(8L_s P_b)^{d_{max}}}{8(L_s - h_{ov})}$ where $(8L_s P_b)^{d_{max}}$ is MAC frame error probability after d_{max} ARQ retransmission. Each video frame is coded as $\frac{x_s}{V_r}$ bits where V_r is a video frame rate, and one MAC frame carries $8(L_s - h_{ov})$ information bits. Thus, the number of MAC frames to transfer one video frame is $\frac{x_s}{V_r} \cdot \frac{1}{8(L_s - h_{ov})}$. If one of MAC frames to carry a picture fails, the whole picture (one slice) is lost since a video picture is coded as a single slice. Consequently, slice error probability is a product of the number of MAC frames for a picture and MAC frame error probability. Equation (30) shows that the source bit rate should be less than or equal to the goodput x_{gp} of MAC layer in (27). The main solutions of this problem are the source code rate x, the MAC frame size L and the channel code rate r among the optimization variables. The slice error probability p can be considered as an auxiliary variable. Each optimization variable has its own minimum and maximum constraints which are represented as y_s^{min} and y_s^{max} where $y_s \in \{x_s, L_s, r_s, p_s\}$.
We rewrite the problem (28) for simple notation:

$$\max_{\mathbf{x},\mathbf{p},\mathbf{L},\mathbf{r},\mathbf{t}} \quad \sum_s U(x_s, p_s) \tag{31}$$

$$s.t. \quad P(x_s, L_s, r_s) \le p_s \tag{32}$$

$$x_s \le x_{gp}(t_s, L_s, r_s) \tag{33}$$

$$x_s^{min} \le x_s \le x_s^{max}, \ r_s^{min} \le r_s \le r_s^{max}$$

$$L_s^{min} \le L_s \le L_s^{max}, \ p_s^{min} \le p_s \le p_s^{max}$$

$$\sum_s t_s \le 1, \ t_s \ge 0, \quad \forall s$$

The problem (31) can be solved by the primal-dual decomposition, as explained in Palomar & Chiang (2007). First, we consider a primal decomposition of problem (31) by fixing the scheduling of the channel time allocation **t**. Then the problem (31) becomes two optimization problems as follows:

$$\begin{aligned}
\max_{\mathbf{x},\mathbf{p},\mathbf{L},\mathbf{r}} \quad & \sum_s U_s(x_s, p_s) && (34)\\
s.t. \quad & P_s(x_s, L_s, r_s) \le p_s && (35)\\
& x_s \le x^s_{gp}(L_s, r_s), \quad \forall s && (36)\\
& x_s^{min} \le x_s \le x_s^{max}, \;\; r_s^{min} \le r_s \le r_s^{max} &&\\
& L_s^{min} \le L_s \le L_s^{max}, \;\; p_s^{min} \le p_s \le p_s^{max} &&
\end{aligned}$$

and

$$\begin{aligned}
\max_{\mathbf{t}} \quad & \sum_s U_s^*(\mathbf{t}) && (37)\\
& \sum_s t_s \le 1, \;\; t_s \ge 0, \quad \forall s &&
\end{aligned}$$

where $U_s^*(\mathbf{t})$ is the optimal objective value of each source in (34) for a given **t**. The coupled constraints of (34) are decomposed by taking log of the constraints (35) and (36) and transforming optimization variables as $\tilde{x}_s = \log x_s, \tilde{p}_s = \log p_s, \tilde{L}_s = \log L_s$ and $\tilde{r}_s = \log r_s$ as in Chiang et al. (2007); Lee et al. (2006). Consequently the problem in (34) becomes

$$\begin{aligned}
\max_{\tilde{\mathbf{x}},\tilde{\mathbf{p}},\tilde{\mathbf{L}},\tilde{\mathbf{r}}} \quad & \sum_s \tilde{U}_s(\tilde{x}_s, \tilde{p}_s) && (38)\\
s.t. \quad & \tilde{P}_s(\tilde{x}_s, \tilde{L}_s, \tilde{r}_s) \le \tilde{p}_s &&\\
& \tilde{x}_s \le \tilde{x}^s_{gp}(\tilde{L}_s, \tilde{r}_s), \quad \forall s &&\\
& \tilde{x}_s^{min} \le \tilde{x}_s \le \tilde{x}_s^{max}, \;\; \tilde{r}_s^{min} \le \tilde{r}_s \le \tilde{r}_s^{max} &&\\
& \tilde{L}_s^{min} \le \tilde{L}_s \le \tilde{L}_s^{max}, \;\; \tilde{p}_s^{min} \le \tilde{p}_s \le \tilde{p}_s^{max} &&
\end{aligned}$$

where $\log y_s^{min} = \tilde{y}_s^{min}$, $\log y_s^{max} = \tilde{y}_s^{max}$ and $\log y_s = \tilde{y}_s$ and $y_s \in \{x_s, L_s, r_s, p_s\}$, and the functions $\tilde{U}_s(\tilde{x}_s, \tilde{p}_s)$, $\tilde{P}_s(\tilde{x}_s, \tilde{L}_s, \tilde{r}_s)$ and $\tilde{x}^s_{gp}(\tilde{L}_s, \tilde{r}_s)$ can be derived from the complete formulation presented below.

$$\begin{aligned}
\max_{\tilde{\mathbf{x}},\tilde{\mathbf{p}},\tilde{\mathbf{L}},\tilde{\mathbf{r}}} \quad & -\sum_s \left(a_1 e^{-a_2 \tilde{x}_s} + \frac{e^{\tilde{p}_s}}{I_r(1 - e^{\tilde{p}_s})} D_{ECP} \right)\\
s.t. \;\; & \tilde{x}_s + d_{max}\left[\log 8 + \tilde{L}_s + \log(p_1 e^{p_2 e^{\tilde{r}_s}} + p_3 e^{p_4 e^{\tilde{r}_s}}) \right] - \log(e^{\tilde{L}_s} - h_{ov}) - \log(8V_r) \le \tilde{p}_s\\
& \tilde{x}_s \le \log(t_s) + \log(8N_{SD}) + \log(e^{\tilde{L}_s} - h_{ov}) + \tilde{r}_s\\
& + \log\left[1 - 8e^{\tilde{L}_s}(p_1 e^{p_2 e^{\tilde{r}_s}} + p_3 e^{p_4 e^{\tilde{r}_s}}) \right] - \log(A_0 e^{\tilde{r}_s} + A_1 e^{\tilde{L}_s} + A_2)
\end{aligned}$$

$$\tilde{x}_s^{min} \le \tilde{x}_s \le \tilde{x}_s^{max}, \;\; \tilde{r}_s^{min} \le \tilde{r}_s \le \tilde{r}_s^{max}$$
$$\tilde{L}_s^{min} \le \tilde{L}_s \le \tilde{L}_s^{max}, \;\; \tilde{p}_s^{min} \le \tilde{p}_s \le \tilde{p}_s^{max}, \;\; \forall s$$

where

$$A_0 = 2N_{SD}(T_{Preamble} + T_{Sig} + SIFS),$$
$$A_1 = \frac{8T_{symbol}}{\log_2 M_o^s} \text{ and } A_2 = \frac{8L_{ack}T_{symbol}}{\log_2 M_o^s}$$

The problem (38) is a convex optimization problem and satisfies the Slater's qualification condition. Therefore, the Lagrangian duality can be used to obtain the optimal solutions (Boyd & Vandenberghe (2004)) which is called as a dual decomposition in Palomar & Chiang (2007). The partial Lagrangian of the problem (38) is

$$\begin{aligned}\mathcal{L}(\tilde{\mathbf{x}}, \tilde{\mathbf{P}}, \tilde{\mathbf{L}}, \tilde{\mathbf{r}}, \breve{}, \mathbf{fl}) &= \sum_s \tilde{U}_s(\tilde{x}_s, \tilde{p}_s) + \sum_s \Big[\gamma_s \Big(\tilde{p}_s - \tilde{P}_s(\tilde{x}_s, \tilde{L}_s, \tilde{r}_s) \Big) + \lambda_s \Big(\tilde{x}_{gp}^s(\tilde{L}_s, \tilde{r}_s) - \tilde{x}_s \Big) \Big] \\ &= \sum_s \mathcal{L}_s(\tilde{x}_s, \tilde{p}_s, \tilde{L}_s, \tilde{r}_s, \lambda_s, \gamma_s)\end{aligned}$$

where λ and γ are Lagrange multipliers. Moreover, the Lagrangian dual function is given as follows:

$$Q(\lambda, \gamma) = \max_{\tilde{\mathbf{x}}, \tilde{\mathbf{p}}, \tilde{\mathbf{L}}, \tilde{\mathbf{r}}} \sum_s \mathcal{L}_s(\tilde{x}_s, \tilde{p}_s, \tilde{L}_s, \tilde{r}_s, \lambda_s, \gamma_s) \tag{39}$$
$$\tilde{x}_s^{min} \le \tilde{x}_s \le \tilde{x}_s^{max}, \;\; \tilde{r}_s^{min} \le \tilde{r}_s \le \tilde{r}_s^{max}$$
$$\tilde{L}_s^{min} \le \tilde{L}_s \le \tilde{L}_s^{max}, \;\; \tilde{p}_s^{min} \le \tilde{p}_s \le \tilde{p}_s^{max}$$

The problem (39) can be solved at each source since the Lagrangian is separable. Therefore, the dual problem is also solved separately as follows:

$$\min_{\lambda \ge 0, \gamma \ge 0} \sum_s Q_s(\lambda_s, \gamma_s) \tag{40}$$

where

$$Q_s(\lambda_s, \gamma_s) = \max_{\tilde{x}_s, \tilde{p}_s, \tilde{L}_s, \tilde{r}_s} \mathcal{L}_s(\tilde{x}_s, \tilde{p}_s, \tilde{L}_s, \tilde{r}_s, \lambda_s, \gamma_s)$$

The dual problem is solved by the gradient projection method if the dual function $Q_s(\lambda_s, \gamma_s)$ is differentiable as in D.P.Bertsekar (2003):

$$\lambda_s^{k+1} = \left[\lambda_s^k - \eta^k \frac{\partial Q_s}{\partial \lambda_s} \right]^+,$$

$$\gamma_s^{k+1} = \left[\gamma_s^k - \eta^k \frac{\partial Q_s}{\partial \gamma_s}\right]^+$$

where η^k is a positive step size at iteration k, and $[\cdot]^+$ denotes the projection onto the nonnegative orthant. The projection operation guarantees that the Lagrange multipliers λ and γ satisfy their nonnegative conditions. In the previous formulation, we solve the optimization problem (34) for given the channel time $\mathbf{t}$. Here, we solve the master primal problem (37) using the subgradient method in D.P.Bertsekar (2003); Johansson & Johansson (2005). The subgradient of $U_s^*(t_s)$ with respect to t_s is given by $\lambda_s^*(t_s)\frac{\partial \tilde{x}^s_{gp}(t_s)}{\partial t_s}$ (Johansson & Johansson (2005)) where $\lambda_s^*(t_s)$ is the optimal Lagrange multiplier associated with the constraint $\tilde{x}_s \leq \tilde{x}^s_{gp}(\tilde{L}_s, \tilde{r}_s)$ in (38) for a given t_s. Therefore, the master primal problem (37) updates the channel time allocation $\mathbf{t}$ as follows:

$$\tilde{\mathbf{t}}^{k+1} = \mathbf{t}^k + \eta^k \begin{bmatrix} \lambda_1^*(t_1)\tilde{x}'^{1}_{gp}(t_1) \\ \vdots \\ \lambda_s^*(t_s)\tilde{x}'^{s}_{gp}(t_s) \end{bmatrix}, \quad \mathbf{t}^{k+1} = \left[\tilde{\mathbf{t}}^{k+1}\right]_{\mathcal{P}} \tag{41}$$

where $\tilde{x}'^{s}_{gp}(t_s) = \frac{\partial \tilde{x}^s_{gp}(t_s)}{\partial t_s}$ and $[\cdot]_{\mathcal{P}}$ denotes the projection onto the feasible convex set $\mathcal{P} \triangleq \{\mathbf{t} : \mathbf{t} \succeq 0, \sum_s t_s \leq 1\}$. Due to the projection, this subgradient update cannot be performed independently by each source. A coordinator in a BSS can solve the primal problem. The projection onto the feasible convex set can be formulated as another optimization problem as follows:

$$\begin{aligned} &\min_{\mathbf{t}} \parallel \mathbf{t} - \tilde{\mathbf{t}} \parallel^2 && (42) \\ &s.t. \sum_s t_s \leq 1, \;\; t_s \geq 0, \;\; \forall s \end{aligned}$$

The problem (42) is formulated from the fact that the projected point $\mathbf{t}$ from $\tilde{\mathbf{t}}$ minimizes the distance between two points. This problem can be solved using the very efficient algorithm in Palomar (2005).

If a picture is segmented as multiple slices, the problem (31) is modified as follows:

$$\begin{aligned} \max_{\mathbf{x},\mathbf{p},\mathbf{L},\mathbf{r},\mathbf{t},\mathbf{n}} \quad & \sum_s U(x_s, p_s) && (43) \\ s.t. \quad & N(x_s, n_s) \leq 8(L_s - h_{ov}) && (44) \\ & P(L_s, r_s) \leq p_s && (45) \\ & x_s + X_{SL}(x_s, n_s) \leq x_{gp}(t_s, L_s, r_s) && (46) \\ & x_s^{min} \leq x_s \leq x_s^{max}, \;\; r_s^{min} \leq r_s \leq r_s^{max} \\ & L_s^{min} \leq L_s \leq L_s^{max}, \;\; p_s^{min} \leq p_s \leq p_s^{max} \\ & n_s^{min} \leq n_s \leq n_s^{max}, \;\; \sum_s t_s \leq 1, \;\; t_s \geq 0, \quad \forall s \end{aligned}$$

where $X_{SL}(x_s, n_s) = \kappa_1(n_s - 1) + \kappa_2\sqrt{x_s}\log(n_s)$ from (14) and (15), and the slice length $N(x_s, n_s)$ is the number of bits per slice, that is, $\frac{x_s+X_{SL}(x_s,n_s)}{V_r} \cdot \frac{1}{n_s}$. The closeness of a coded slice to the bound in (44) is limited by the optimal MAC frame size which satisfies Constraint 1. This constraint is an active constraint at an optimal solution, that is, the slice length $N(x_s, n_s)$ is equal to information bits of one MAC frame $8(L_s - h_{ov})$. Equation (46) and utility functions in (43) are derived from the experimental results of subsection 2.2 the number of slices does not affect source-induced distortion but rather source-coded bit rates. Therefore, a source-coded bit rate (one sliced source code rate) x_s is increased by bit increments $X_{SL}(x_s, n_s)$ according to the number of slices and a coding bit rate x_s. However, utility functions are not functions of the sum of x_s and $X_{SH}(x_s, n_s)$ in order to maintain distortion at x_s since they do not depend on $X_{SH}(x_s, n_s)$ which are bit increments of the number of slices. If equation (44) is satisfied, the error probability of slices p_s of (45) is just error probability of a MAC frame after ARQ because one MAC frame only carries one slice. Thus, $P(L_s, r_s)$ is $(8L_s P_b^s)^{d_{max}}$. Here, one more optimization variable n is added for the number of slices. The complete mathematical formulation of (43) is described as follows:

$$\max_{\mathbf{x},\mathbf{p},\mathbf{L},\mathbf{r},\mathbf{n},\mathbf{t}} \quad -\sum_s \left(a_1 x_s^{-a_2} + \frac{p_s}{I_r(1-p_s)} D_{ECP} \right) \tag{47}$$

$$s.t. \quad \frac{x_s + \kappa_1(n_s - 1) + \kappa_2\sqrt{x_s}\log(n_s)}{V_r \cdot n_s} \le 8(L_s - h_{ov}) \tag{48}$$

$$\left(8L_s(p_1 e^{p_2 r_s} + p_3 e^{p_4 r_s}) \right)^{d_{max}} \le p_s \tag{49}$$

$$x_s + \kappa_1(n_s - 1) + \kappa_2\sqrt{x_s}\log(n_s) \le t_s \cdot N_{SD} \frac{8(L_s - h_{ov})(1 - P_{fr}^s)}{A_0 + \frac{8L_s T_{symbol}}{r_s \log_2 M_o^s} + \frac{8L_{ack} T_{symbol}}{r_s \log_2 M_o^s}} \tag{50}$$

$$x_s^{min} \le x_s \le x_s^{max}, \ r_s^{min} \le r_s \le r_s^{max}, \ L_s^{min} \le L_s \le L_s^{max}$$

$$p_s^{min} \le p_s \le p_s^{max}, \ n_s^{min} \le n_s \le n_s^{max}, \ \sum_s t_s \le 1, \ t_s \ge 0, \quad \forall s$$

where $P_{fr}^s = 8L_s(p_1 e^{p_2 r_s} + p_3 e^{p_4 r_s})$ and $s \in S$, and S is a set of utility functions which transmit their video streams. Utility functions $U(x_s, p_s)$ in (43) are the negative sum of (11) and (9) which was discussed in section 2.2for maximization. The constraints (48) and (49) are relaxed from the equality constraints of a slice length and slice error probability. Equation (50) shows that the source bit rate should be less than or equal to the goodput of MAC layer. The main solutions of this problem are the source code rate x, the MAC frame size L, the channel code rate r, the number of slices n and channel time allocation t among the optimization variables. The slice error probability p can be considered as an auxiliary variable.

The problem (43) is also solved by the primal-dual decomposition method. First, we perform a primal decomposition of the problem (43) by fixing scheduling of the channel time allocation **t**. Then the problem (43) becomes two optimization problems as follows:

$$\max_{\mathbf{x},\mathbf{p},\mathbf{L},\mathbf{r},\mathbf{n}} \quad \sum_s U(x_s, p_s) \tag{51}$$

$$s.t. \quad N(x_s, n_s) \le 8(L_s - h_{ov})$$

$$P(L_s, r_s) \le p_s$$

$$
\begin{aligned}
& x_s + X_{SL}(x_s, n_s) \leq x_{gp}(L_s, r_s) \\
& x_s^{min} \leq x_s \leq x_s^{max}, \;\; r_s^{min} \leq r_s \leq r_s^{max} \\
& L_s^{min} \leq L_s \leq L_s^{max}, \;\; p_s^{min} \leq p_s \leq p_s^{max} \\
& n_s^{min} \leq n_s \leq n_s^{max}
\end{aligned} \tag{52}
$$

and

$$
\begin{aligned}
\max_{\mathbf{t}} \quad & \sum_s U_s^*(\mathbf{t}) \\
& \sum_s t_s \leq 1, \;\; t_s \geq 0, \quad \forall s
\end{aligned} \tag{53}
$$

where $U_s^*(\mathbf{t})$ is the optimal objective value of each source in (51) for given $\mathbf{t}$. The problem has strong duality and thus, the Lagrangian duality can be applied to obtain the optimal solutions. The procedure to solve the problem (43) is almost the same as that for the problem (31).

4. Coexistence among utility functions with or without cross-layer optimization

In the practical environment, we consider coexistence among cross-layered utility functions and conventional utility functions which do not support cross-layer optimization. First, a conventional coordinator which does not solve the primal optimization problem (53) cooperates with cross-layered utility functions. In this case, cross-layered utility functions decide optimal solutions by solving the problem (51) for given channel times. This is just one instance of iterative optimization between primal-dual optimization. Second, a coordinator, which solves the primal optimization problem (53), coexists with conventional utility functions. In this case, each utility function needs to feedback its own subgradient $\lambda_s^*(t_s)\frac{\partial \tilde{x}_{gp}^s(t_s)}{\partial t_s}$, which was explained in the previous section, to a coordinator and then the coordinator can update channel time allocation $\mathbf{t}$ as the equations (41).

The issue is how conventional utility functions estimate their subgradient. In this optimization problem, the subgradient is $\frac{\lambda_s^*(t_s^k)}{t_s^k}$ but allocated channel times for utility functions are already available at the coordinator. Therefore, each utility function only needs to feedback its own λ_s^* for a given channel time t_s^k. The remain problem is how to estimate λ_s^* in the conventional utility functions. We can easily estimate an approximate value $\hat{\lambda}_s$ of λ_s^* from RD optimization of H.264 video encoder. The RD optimization is not standard part of H.264 (2009) but the reference software model of H.264 JM (2007) supports the RD optimization for better performance. The RD optimization is formulated as follows:

$$
\min_{\mathbf{m}} \quad \sum_{n=1}^{N} d_n(\mathbf{m}_n) \quad s.t. \quad \sum_{n=1}^{N} x_n(\mathbf{m}_n) \leq X_F \tag{54}
$$

where $\mathbf{m}_n = (M_n, \mathbf{MV}_n, QP_n, \mathbf{Ref}_n)$ which is a vector of Macro Block (MB) mode, Motion Vectors (MVs), Quantization Parameter (QP) and reference frames for inter prediction. N is the number of MBs in a frame, and X_F is the bit constraint of a frame. d_n and x_n are distortion and coded bits of the nth MB, respectively. The optimization problem (54) can be solved by the Lagrangian duality as follows:

$$q(\lambda) = \min_{\mathbf{m}} \quad \sum_{n=1}^{N} \Big(d_n(\mathbf{m}_n) + \lambda x_n(\mathbf{m}_n) \Big) - \lambda X_F \tag{55}$$

and its dual problem is

$$\max_{\lambda \geq 0} q(\lambda) \tag{56}$$

If we know the optimal solution of the dual problem (56), we can obtain the solution of the primal problem (54) after solving (55). However, in order to simplify the above optimization problems, the relation between λ and QP was derived in Sullivan & Wiegand (1998); Takagi (2002); Weigand et al. (2003); Wiegand & Girod (2001), and estimation of X_F from QP was studied in Chiang & Zhang (1997); Kim (2003). Thus, the reference software model of H.264 JM (2007) has the following relation:

$$\lambda = \kappa 2^{\frac{(QP-12)}{3}}, \tag{57}$$

$$X_F = \gamma \frac{MAD}{Q_{step}} + \xi \frac{MAD^2}{Q_{step}^2}, \;\; Q_{step} = \nu 2^{\frac{QP-12}{6}} \tag{58}$$

where κ is a function of slice types (I, P, B), the number of referenced frames and QP. γ and ξ are estimated using linear regression based on Mean Absolute Difference (MAD) and target bits X_F. ν is a function of QP. Equations (57) and (58) provide an approximate solution for λ of the dual problem (56), that is, QP is estimated from (58) for the given constraint X_F and then λ is induced from (57). Lee et al. (2000); Li et al. (2003) proposed how to estimate target bits X_F from video frame rate, buffer fullness, picture type and some other information.

If average value of the bit constraint X_F is well estimated to match with goodput x_{gp} of (52), average λ of the RD optimization is close to λ^* of subgradient. However, the RD optimization λ is obtained from original variables, but λ^* of subgradient is decided from transformed variables which is denoted in (38). Therefore, we need to find out the relation between transformed domain λ_t and original domain λ. Sullivan & Wiegand (1998) presented $\lambda = -\frac{dD(x)}{dx}$ such that transformed domain $\lambda_t = -\frac{dD(\tilde{x})}{d\tilde{x}} = -\frac{dD(x)}{dx} \cdot \frac{dx}{d\tilde{x}} = \lambda x$, where $\tilde{x} = \log x$.

In summary, the rate control algorithm of the video encoder changes QP not to overflow nor underflow a buffer which means rate of video encoder follows goodput of the network layer, and then λ of original domain is derived from (57). The approximate of subgradient $\hat{\lambda}$ is obtained from multiplication of λ and current coded rate. Average value of the approximate subgradient $\hat{\lambda}$ is transferred to a coordinator to receive updated channel time allocation.

If channel-induced distortion is considered, Reichel et al. (2007) described that λ is changed into $\lambda_{err} = (1-p) \cdot \lambda$. Consequently, conventional utility functions feedback average value of $(1-p) \cdot \lambda \cdot x$ to a coordinator. In order to decide error probability p, there are two ways from the PHY layer to the application layer (bottom-up) and from the application layer to the PHY layer (top-down). In the bottom-up case, error probability of the application layer can be derived from error probability of the network layer after maximizing goodput.

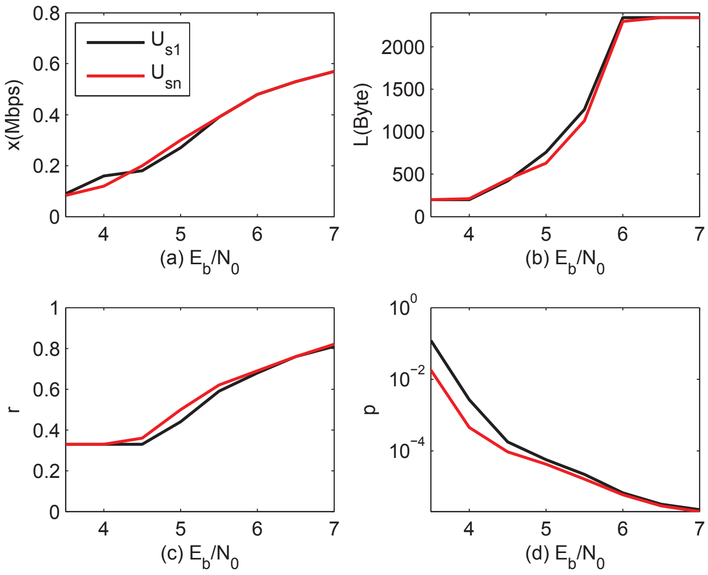


Fig. 12. Optimization variables (x, r, L and p) of a single-sliced utility function and a multiple-sliced utility function from An & Nguyen (2008a) (©[2008] IEEE).

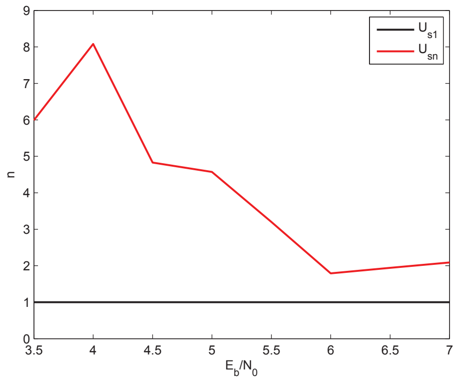


Fig. 13. The optimal number of slices of a multiple-sliced utility function from An & Nguyen (2008a) (©[2008] IEEE).

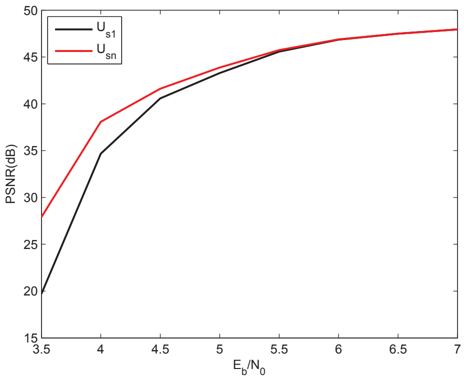


Fig. 14. PSNR of a single-sliced utility function vs. PSNR of a multiple-sliced utility function from An & Nguyen (2008a) (©[2008] IEEE).

5. Numerical Examples

In this chapter, 802.11a-like MAC and 802.11a (1999) PHY are considered for simulations. Main differences from standard 802.11 are TDMA access, fixed modulation, continuous channel code rate and adaptive MAC frame size without MAC fragmentation. The modulation is fixed as BPSK ($M_o^s = 2, \forall s$), and the channel code rate can be changed continuously from the mother code of RCPC to the maximum code rate. The other parameters of the PHY layer are the same as in 802.11a (1999). For simplicity, the maximization of negative end-to-end distortion is solved by the minimization of end-to-end distortion, but we still call the functions as utility functions instead of loss functions.

5.1 Single-sliced utility function vs. Multiple-sliced utility function

In this example, we compare performance of a multiple-sliced utility function U_{sn} with a single-sliced utility function U_{s1}. Here, we only solve the sub-dual problem (51) for U_{sn} and the sub-dual problem in problem (34) for U_{s1} for a given channel time. As a result, the optimal source code rate x, channel code rate r, MAC frame size L and the number of slices n of each utility function are obtained. Figures 12 and 13 show primal optimization variables of two utility functions. From Figure 12 (d), slice error probability p of U_{sn} is smaller than error probability of U_{s1} because the optimal number of slices of U_{sn} is larger as shown in Figure 13. Thus, the source and channel code rate of U_{sn} are higher since less error correction is needed which is shown at $E_b/N_0 > 4.5$ in Figures 12 (a) and (c). However, the source code rate x of U_{sn} is lower at $E_b/N_0 \leq 4.5$ because U_{sn} use channel bit rates for slice coding at the same channel code rate r. From Figure 13, the optimal number of slices increases as SNR decreases which is consistent with general intuition. However, the optimal number of slices is smaller at SNR 3.5 than SNR 4 as shown in Figure 13. The main reason is that the relative bit increments (penalty) w.r.t. the number of slices are larger at low bit rate which is shown in Figure 8 and the optimal source code rate x is decreased to satisfy the network capacity from SNR 3.5 to 4. Figure 14 indicates that gain from the picture segmentation is larger especially at low SNR.

5.2 Channel time allocation for multiple-sliced video coding

In this subsection, we consider channel time allocation for single-sliced and multiple-sliced utility functions. Furthermore, we will present that the optimal channel time allocation highly depends on video contents. For this work, we consider the following simulation environment: there are two BSSs which are completely separate. In each BSS, there are 16 utility functions (single node may have multiple utility functions) which send different video streams. All the utility functions denoted as U_{s1}^s in one BSS code each video picture as a single slice, and utility functions U_{sn}^s in the other BSS use multiple slices for error resilient video coding. All the utility functions operate at SNR 5dB. Multiple-sliced utility functions solve the problem (51), and single-sliced utility functions solve the corresponding subproblem in the problem (34) for given channel times $\mathbf{t}$ which was done in subsection 5.1 Here, the channel times $\mathbf{t}$ are iteratively allocated to utility functions after a coordinator solves the master problem (53). Reallocated channel times are transferred to each node by a beacon signal. Each utility function solves the optimization problem (51) for updated channel times and then feedbacks its subgradient to the coordinator during the contention period. The coordinator updates channel times based on subgradients from all the utility functions within a BSS after solving the problem (53). Thus, a coordinator and utility functions within a BSS keep iteratively solving the optimization problem (53) and (51), respectively.

In order to present dependency between channel times and video contents, utility functions stream different video sequences in two BSSs. However, we only explain multiple-sliced utility functions in one BSS since the situation is the same to single-sliced utility functions in the other BSS. Five utility functions including U_{sn}^{0} send the Football sequence which has high motion in video frames, and another four utility functions along with U_{sn}^{1} send the Crew sequence which has medium motion, and the others including U_{sn}^{2} send the Foreman sequence which has low motion characteristics. Figure 3 shows that different video sequences need different bit rates to achieve similar PSNR, for example the Foreman sequence with low motion needs a lower bit rate than the Football sequence with high motion to obtain similar PSNR. Although there are 16 utility functions within a BSS, the results of three representative utility functions U_{sn}^{0}, U_{sn}^{1} and U_{sn}^{2} are only explained, because the other utility functions operate the same as their representative utility functions.

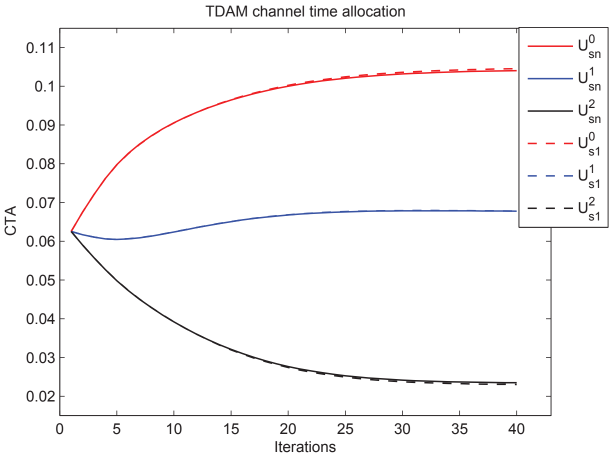


Fig. 15. Channel time allocation of three utility functions with one slice and multiple slices from An & Nguyen (2008a) (©[2008] IEEE).

From Figure 15, channel times are equally allocated to all the utility functions at the initial iteration in both BSSs. All the utility functions solve the optimization problem for given channel times, and their end-to-end distortion D_t in Figure 16 are calculated based on their optimal solutions as shown in Figure 12. From Figures 15 and 16, equal channel time allocation induces larger distortion to U_{s1}^{0} and U_{sn}^{0} which send the high-motion video streams. After several iterations, distortion of U_{s1}^{0} and U_{sn}^{0} is significantly reduced as channel times for U_{s1}^{0} and U_{sn}^{0} increase. On the contrary, distortion of U_{s1}^{2} and U_{sn}^{2} is a little bit increased due to smaller channel times. Thus, the sum of distortion of utility functions can be diminished. In Figure 15, utility functions with multiple slices induce less channel time variations from the initial iteration to the last iteration because utility functions with error resilient feature have less distortion as shown in Figure 16. However, the variations mainly depend on video contents. We can approximately allocate the same channel times without consideration of picture segmentation. In spite of similar channel time allocation, distortion of multiple-sliced utility functions U_{sn}^{s} is lower than single-sliced utility functions U_{s1}^{s} as shown in Figure 16. In addition, distortion gap among the utility functions is further reduced due to multiple slices. Figure 17 illustrates the optimal number of slices of U_{sn}^{0}, U_{sn}^{1} and U_{sn}^{2} according to the allocated channel times where the number of slice of U_{s1}^{s} is one. The more channel time is allocated to a utility function, the more number of slices is needed.

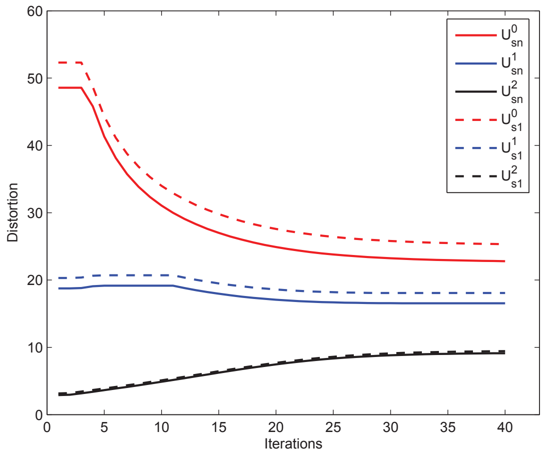


Fig. 16. End-to-end distortion D_t of three utility functions with one slice and multiple slices from An & Nguyen (2008a) (©[2008] IEEE).

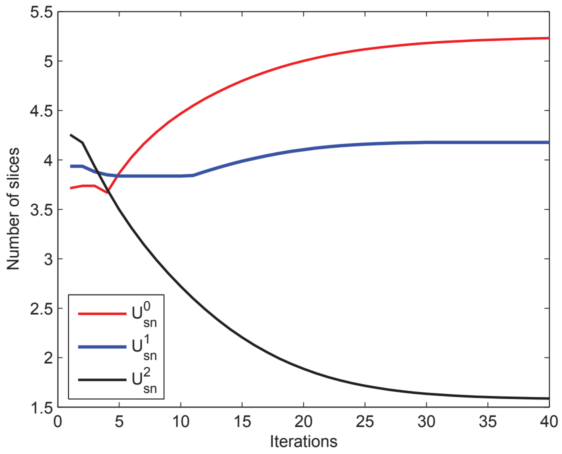


Fig. 17. Optimal number of slices of three utility functions with multiple slices from An & Nguyen (2008a) (©[2008] IEEE).

6. Conclusion

In this chapter, we show that elaborate mathematical models for resource allocation of the source code rate, channel code rate, MAC frame length and multiple slice coding with channel time allocation of TDMA can be formulated as a convex optimization problem. We also derive a mathematical model for multiple-sliced video coding to describe trade-offs between coding efficiency and error protection, and then we apply it for the joint optimization with the MAC and PHY layer constraints. The optimal sliced video coding gives larger gain at especially low SNR. Furthermore, error resilient video coding can achieve better performance along with the optimal channel time allocation. In this work, we use the distortion function as an objective method to evaluate video quality, and then we find optimal solutions to minimize the sum of end-to-end distortion.

7. References

802.11 (1999). Part 11:wireless lan medium access control (MAC) and physical layer (PHY) specifications.

802.11a (1999). Part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications: Higher speed physical layer in the 5 ghz band.

802.11e (2005). Wireless lan medium access control (MAC) and physical layer (PHY) specifications : Medium access control (mac) quality of service enhancements.

An, C. & Nguyen, T. Q. (2007). Analysis of utility functions for video, *Proc. IEEE ICIP*.

An, C. & Nguyen, T. Q. (2008a). Resource allocation for error resilient video coding over awgn using optimization approach, *IEEE Trans. Image Processing* 17: 2347–2355.

An, C. & Nguyen, T. Q. (2008b). Resource allocation for TDMA video communication over AWGN using cross-layer optimization approach, *IEEE Trans. on Multimedia* 10: 1406–1418.

Boyd, S. & Vandenberghe, L. (2004). *Convex Optimization*, Cambridge University Press.

Bystrom, M. & Modestino, J. W. (2000). Combined source-channel coding schemes for video transmission over an additive white gaussian noise channel, *IEEE J. Select. Areas Commun.* 18: 880–890.

Cheung, G. & Zakhor, A. (2000). Bit allocation for joint source/channel coding of scalable video, *IEEE Trans. Image Processing* 9: 340 – 356.

Chiang, M., Low, S. H., Calderbank, A. R. & Doyle, J. C. (2007). Layering as optimization decomposition: A mathematical theory of network architectures, *Proc. of IEEE*. to appear.
URL: *http://www.princeton.edu/ chiangm/publications.html*

Chiang, T. & Zhang, Y.-Q. (1997). A new rate control scheme using quadratic rate distortion model, *IEEE Trans. Circuits Syst. Video Technol.* 7: 246–250.

Chiew, T.-K., Hill, P., Ferre, P., Agrafiotis, D., Chung-how, J. T., Nix, A. & Bull, D. R. (2005). Error-resilient low-delay h.264/802.11 transmission via cross-layer coding with feedback channel, *Proc. Visual Communications and Image Processing*.

Chou, P. A. & Miao, Z. (2006). Rate-distortion optimized streaming of packetized media, *IEEE Trans. Multimedia* 8: 390–404.

Cote, G., Shirani, S. & Kossentini, F. (2000). Optimal mode selection and synchronization for robust videocommunications over error-prone networks, *IEEE J. Select. Areas Commun.* 18: 952–965.

D.P.Bertsekar (2003). *Nonlinear Programming*, second edn, Athena Scientific.

H.264, I.-T. R. (2009). Advanced video coding for generic audiovisual services.

Hagenauer, J. (1988). Rate-compatible punctured convolutional codes (RCPC Codes) and their applications, *IEEE Trans. Commun.* 34: 389–400.

Haratcherev, I., Taal, J., Langendoen, K., Lagendijk, R. & Sips, H. (2005). Fast 802.11 link adaptation for real-time video streaming by cross-layer signaling, *Proc. IEEE ISCAS*.

Haratcherev, L., Taal, J., Langendoen, K., Lagendijk, R. & Sips, H. (2006). Optimized video streaming over 802.11 by cross-layer signaling, *IEEE Communications Magazine* 8: 115–121.

Harmanci, O. & Tekalp, A. (2005). Rate distortion optimized slicing over bit error channels, *Proc. SPIE*.

Hochwald, B. & Zeger, K. (1997). Tradeoff between source and channel coding, *IEEE Trans. Inform. Theory* 43: 1412–1424.

Izzat, I., Mayer, M., Rhodes, R. & Stahl, T. (2005). Real time transmission of mpeg2 video over 802.11a wireless lans, *Proc. IEEE ICCE*.

JM (2007). *H.264/AVC reference software (JM11.0)*, HHI.
URL: *http://iphome.hhi.de/suehring/tml/download/*

Johansson, B. & Johansson, M. (2005). Primal and dual approaches to distributed cross-layer optimization, *Proc. 16th IFAC World Congress*.

k. Jain, A. (1989). *Fundamentals of Digital Image Processing*, Prentice-Hall International Editions.

K, S., Farber, N., Link, M. & Girod, B. (2000). Analysis of video transmission over lossy channels, *IEEE J. Select. Areas Commun.* 18: 1012 – 1032.

Kalman, M. & Girod, B. (2005). Optimal channel-time allocation for the transmission of multiple video streams over a shared channel, *Proc. IEEE MMSP*.

Kelly, F. P., Maulloo, A. & Tan), D. (1998). Rate control for communication networks: Shadow prices, proportional fairness and stability, *J. Oper. Res. Soc.* 49: 273–252.

Kim, H. (2003). Adaptive rate control using nonlinear regression, *IEEE Trans. Circuits Syst. Video Technol.* 13: 432–439.

Lee, H., Chiang, T. & Zhang, Y. (2000). Scalalble rate control for mpeg-4 video, *IEEE Trans. Circuits Syst. Video Technol.* 10: 878–894.

Lee, J. W., Chiang, M. & Calderbank, R. A. (2006). Price-based distributed algorithm for optimal rate-reliability tradeoff in network utility maximization, *IEEE J. Select. Areas Commun."* 24: 962–976.

Li, Z., Zhu, C., Ling, N., Yang, X., Feng, G., Wu, S. & Pan, F. (2003). A unified architecture for real-time video-coding systems, *IEEE Trans. Circuits Syst. Video Technol.* 13: 472– 487.

Lin, S. & Costello, D. J. (2004). *Error Control Coding*, second edn, Prentice Hall.

Masala, E., Yang, H., Rose, K. & Marthin, J. D. (2004). Rate-distortion optimized slicing, packetization and coding for error resilient video transmission, *Proc. IEEE DCC*.

Palomar, D. (2005). Convex primal decomposition for multicarrier linear mimo transceivers, *IEEE Trans. Signal Processing* 53(12): 4661–4674.

Palomar, D. & Chiang, M. (2007). Alternative distributed algorithms for network utility maximization: Framework and applications. to be published.
URL: *http://www.princeton.edu/ chiangm/publications.html*

Qiao, D. & Choi, S. (2001). Goodput enhancement of ieee 802.11a wireless lan via link adaptation, *Proc. IEEE ICC*.

Reichel, J., Schwarz, H. & e. Wien, M. (2007). Joint scalable video model jsvm-9.

Richardson, I. E. G. (2003). *H.264 and MPEG-4 video compression: video coding for next-generation Multimedia*, John Wiley & Sons Press.

Sullivan, G. J. & Wiegand, T. (1998). Rate-Distortion Optimization for Video Compression, *IEEE Signal Processing Mag.* 15: 74–99.

Takagi, K. (2002). Lagrange Multiplier and RD-characteristics.

Taubman, D. S. & Marcellin, M. W. (2002). *JPEG2000 : Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers.

Wang, Y., Wu, Z. & Boyce, J. M. (2006). Modeling of transmission-loss-induced distortion in decoded video, *IEEE Trans. Circuits Syst. Video Technol.* 16: 716–732.

Weigand, T., Schwarz, H., Joch, A., Kossentini, F. & Sullivan, G. J. (2003). Rate-Constrained Coder Control and Comparison of Video coding Standards, *IEEE Trans. Circuits Syst. Video Technol.* 13: 688–703.

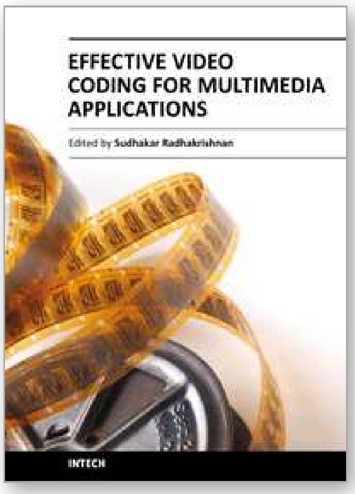
Wenger, S. (2003). H.264/avc over ip, *IEEE Trans. Circuits Syst. Video Technol.* 13: 645–656.

Wiegand, T. & Girod, B. (2001). Lagrange multiplier selection in hybrid video coder control, *Proc. IEEE ICIP*.

Wu, Q., Chan, S.-C. & Shum, H.-Y. (2006). A convex optimization-based frame-level rate control algorithm for motion compensated hybrid dct/dpcm video coding, *Proc. IEEE ICIP*.

Wu, Z. & Boyce, J. M. (2007). Adaptive error resilient video coding based on redundant slices of h.264/avc, *Proc. IEEE ICME*.

Z.He, j. Cai & Chen, C. W. (2002). Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding, *IEEE Trans. Circuits Syst. Video Technol.* 12: 511–523.


EFFECTIVE VIDEO CODING FOR MULTIMEDIA APPLICATIONS
Edited by Sudhakar Radhakrishnan
INTECH

Effective Video Coding for Multimedia Applications
Edited by Dr Sudhakar Radhakrishnan

ISBN 978-953-307-177-0
Hard cover, 292 pages
Publisher InTech
Published online 26, April, 2011
Published in print edition April, 2011

Information has become one of the most valuable assets in the modern era. Within the last 5-10 years, the demand for multimedia applications has increased enormously. Like many other recent developments, the materialization of image and video encoding is due to the contribution from major areas like good network access, good amount of fast processors e.t.c. Many standardization procedures were carrried out for the development of image and video coding. The advancement of computer storage technology continues at a rapid pace as a means of reducing storage requirements of an image and video as most situation warrants. Thus, the science of digital video compression/coding has emerged. This storage capacity seems to be more impressive when it is realized that the intent is to deliver very high quality video to the end user with as few visible artifacts as possible. Current methods of video compression such as Moving Pictures Experts Group (MPEG) standard provide good performance in terms of retaining video quality while reducing the storage requirements. Many books are available for video coding fundamentals.This book is the research outcome of various Researchers and Professors who have contributed a might in this field. This book suits researchers doing their research in the area of video coding.The understanding of fundamentals of video coding is essential for the reader before reading this book. The book revolves around three different challenges namely (i) Coding strategies (coding efficiency and computational complexity), (ii) Video compression and (iii) Error resilience. The complete efficient video system depends upon source coding, proper inter and intra frame coding, emerging newer transform, quantization techniques and proper error concealment.The book gives the solution of all the challenges and is available in different sections.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Cheolhong An and Truong Q. Nguyen (2011). Error Resilient Video Coding using Cross-Layer Optimization Approach, Effective Video Coding for Multimedia Applications, Dr Sudhakar Radhakrishnan (Ed.), ISBN: 978-953-307-177-0, InTech, Available from: http://www.intechopen.com/books/effective-video-coding-for-multimedia-applications/error-resilient-video-coding-using-cross-layer-optimization-approach

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China

www.intechopen.com

51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821