Spring 4-19-2011

OPTIMIZED DELAY-SENSITIVE MULTIMEDIA COMMUNICATIONS OVER WIRELESS NETWORKS

Haiyan Luo
University of Nebraska-Lincoln, petrel.luo@gmail.com

OPTIMIZED DELAY-SENSITIVE MULTIMEDIA COMMUNICATIONS OVER WIRELESS NETWORKS

by

Haiyan Luo

A Dissertation

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Doctor of Philosophy

Major: Engineering

Under the Supervision of Professor Song Ci

Lincoln, Nebraska

April, 2011

OPTIMIZED DELAY-SENSITIVE MULTIMEDIA COMMUNICATIONS OVER WIRELESS NETWORKS

Haiyan Luo, Ph.D.

University of Nebraska, 2011

Advisor: Song Ci

Multimedia communication is expected to be the major application of the next-generation wireless networks. However, providing delay-sensitive, loss-tolerant and resource-guzzling multimedia services over the current resource-limited wireless networks is still a very challenging task. In addition to their heterogeneous nature and resource limitation, wireless networks also suffer from channel variations and environmental changes. In this dissertation, we study the optimization techniques for wireless multimedia communications and address the issues from different theoretical and practical perspectives.

To solve the issues resulting from the increasingly heterogeneous wireless networks, we propose a framework for error resilient source coding and distributed application-layer error control, where the encoder calculates source encoding options, enabling globally optimal rate allocation between source and channel bits for all the hops considered in the end-to-end path.

Further, we propose content-aware video communications to differentiate the region of interest (ROI) from the less important background area so that resources can be classified, prioritized and more efficiently utilized for encoding and transmission. We also apply the theories by proposing a wireless e-healthcare system.

To take into consideration of channel variations and to provide a framework that can dynamically adapt to the instant channel changes, we propose a cross-layer optimized framework which jointly optimizes different functions residing at different network layers, including video encoding, network congestion, retransmission and link adaptation. We also demonstrate how this design can be widely used in wireless environments by presenting our research on the scheduling algorithm of wireless peer-to-peer (P2P) multimedia networks.

At last, we move one step further on dynamic adaptation to many other "context" information such as user preference, location, weather, etc. By providing the design of an adaptive multimedia communications system based on context-aware services, we study

how to retrieve and utilize all the context data to better serve the users with significantly-improved video quality.

The proposed optimization techniques are usually discussed within certain application scenarios to demonstrate the practicality, efficacy and efficiency. However, these research findings can be widely used in other wireless networks and applications.

Acknowledgments

First of all, I would like to express my sincere gratitude to my advisor, Prof. Song Ci, for his excellent supervision and inspiring support during the past few years. His extensive expertise and rich experience combining with his passion and enthusiasm for research have made my Ph.D. studies here at the Intelligent Ubiquitous Computing Laboratory (iUbiComp Lab) of the University of Nebraska-Lincoln very pleasant. He is a wonderful professor and I also appreciate his guidance and support on my professional and personal development.

I am very grateful to my committee members: Prof. Hamid Sharif, Prof. Hong Jiang, and Prof. Haifeng Guo. I would like to thank them for serving on my committee and for their time and effort on my research and the dissertation. Especially, I would like to express my gratitude to Prof. Hamid Sharif for his kind encouragement and invaluable advice during my doctoral studies.

I am grateful to Dr. Haohong Wang, Prof. Aggelos Katsaggelos, Dr. Antonios Argyriou and Dr. Hui Tang. They have provided very constructive comments and discussions in the development of this research.

I would like to thank all members in the iUbiComp Lab, especially Dr. Dalei Wu and Dr. Jiucai Zhang for all the useful discussions and all the gracious help on my research during the past few years. I also owe my thanks to all colleagues at the Peter Kiewit Institute, past and present, for creating a friendly, pleasant, and stimulating atmosphere over these years. My special thanks go to Dr. Shichuan Ma, Dr. Honggang Wang, Dr. Hua Fang, Dr. Wei Wang and Dr. Kun Hua for their kind help.

I also like to thank my previous managers, co-workers and friends at Lucent Technologies for their encouragement on my Ph.D. studies and gracious help on my life in the USA: Dr. Yuming Ma, Dr. Xin Wang, Ms. Xiaoling Wang (Dale), Mr. Wenwei Tian (Mark), Mr. Hongsheng Qi (Hollis), Mr. Thong Luu, Ms. Carolyn Spencer Brown, Mr. Austin Qi, Ms. Shilu Ni (Lucy), Dr. Min Zhao, Ms. Chen Yang (Grace), Ms. Jing Sun (Jane), Mr.

Guanghua Huang (Jacob), Dr. Samuel Tan, Ms. Jingyi Wei, Dr. Taiming Feng etc.

My gratitude also goes to my friends Ben LeDuc and Carolyn Spencer Brown for their help on my English.

Finally, this dissertation is dedicated to my wife and parents for their love, sacrifice, and support and to my daughter who has been the the fun and inspirer of my life.

This work was partially supported by NSF under Grant No. CCF-0830493 and CNS-0707944. It was also partially supported by the Layman Foundation.

To my wife Rita Zhang and my parents

for their unconditional love and support!

To my lovely daughter Claire!

Contents

Figures

Tables

Chapter 1

Introduction

1.1 Research Background

Delay-sensitive multimedia communications services have brought profound changes to human society. More and more people have found their lives being enriched and facilitated by video applications such as video telephony, online video streaming, video conferencing, video gaming, and mobile TV broadcasting. For example, YouTube, PPStream, video phones have increasingly gained in popularity. Furthermore, according to statistics, multimedia applications have already become the major constantly-increasing traffic in the Internet environment, dwarfing the traditional text-based network traffic resulting from HTTP web services. It is worth mentioning that advanced video-aided applications such as telepresence, video surveillance, video-based e-healthcare systems have also gained more and more commercial usage.

On the other hand, as one of the fastest growing industries, wireless products have become increasingly popular. Laptops, PDAs, cell phones, iPADs and GPSs have not only enriched people's daily lives, but also require higher bandwidth for richer applications. With the advances of the next-generation networks, such as 3rd/4th generation (3G/4G) mobile telecommunications networks, WiFi, the IEEE 806.16-based Wireless Metropolitan Area networks (WMAN) and the IEEE 802.11-based Wireless Local Area Networks (WLAN), ubiquitous computing has become a life reality. In the industry, Intel has invested a huge amount of money in the research and development of mobile computing. PayPal is moving fast towards mobile payment. The increasing mobility, energy-efficient networks, increasing bandwidth efficiency and state-of-the-art wireless technologies have led to various new applications and have brought profound changes to every aspect of our lives. Increasing mobility

with higher bandwidth has become the trend of today's and future wireless development. Another technological trend in this regard is that wireless networks have become increasingly heterogeneous due to the coexistence of different natures of wireless and mobility technologies.

Moreover, recent advances in video compression techniques have made video compression algorithms more efficient, more flexible, and more robust. These significant advances have promoted the wireless delivery of high-quality video by using relatively low bit rates while maintaining high fidelity. As one of the most state-of-the-art video codec, H.264/AVC has achieved significant improvement in rate-distortion efficiency [2]. According to the standard, it can provide approximately 50% bit rate saving for equivalent perceptual quality relative to the performance of prior standards.

Due to the rapid development of next-generation wireless technologies, increasing mobility and state-of-the-art video compression technologies, multimedia applications are increasingly become wireless and mobile. A wide variety of wireless multimedia applications have become feasible and increasingly popular, such as watching football through video phones, playing video games on iPAD, and using wireless multimedia sensors for video surveillance. Furthermore, there is a strong and constantly increasing demand to bring more and more multimedia applications to the pervasive and mobile computing devices.

1.2 Research Motivation

Despite the increasing demand, wireless multimedia communications, especially real-time video applications, still suffer from a lot of problems. The wireless environment is so different from the Internet that it usually leads to performance degradation by directly applying video transmission techniques that are used in the current Internet environment.

With the advancement of Gigabyte networks and fiber-based high speed transmission, bandwidth is usually not an issue in today's Internet environment. On the contrary, most of the problems for media streaming applications over wireless networks can still be identified

as the lack of bandwidth guarantees, random packet losses, and delay jitter [3]. Even though bandwidth is increasingly available for end users, high-quality multimedia streaming applications are still bandwidth-demanding. Furthermore, delay jitter and packet losses make the problem of video delivery more challenging, especially in multi-hop wireless mesh networks, where an end-to-end path that consists of several interconnected physical networks is usually heterogeneous with asymmetric properties.

The current video communications technologies used in the Internet environment usually treat all the pixels in the video clip with equal importance. Thus, their transmissions and propagations generally consume an equal amount of resources. This works well in the Internet environment since resources such as bandwidth and available energy can be assumed to be large enough. However, in delay-sensitive wireless environments, especially mobile networks, resources are much more limited. Treating everything the same will lead to a waste of resources and thus the degradation of overall system performance. In contrast, classifying the video frames by identifying the region of interest (ROI) can lead to more efficient resource allocation and thus enhancement of the overall system performance.

Further, the loss-tolerant Internet environment is operated on the basis of the Open Systems Interconnection (OSI) model, which works well in the wireline environment. The Internet-based multimedia applications assume perfect channel conditions. However, wireless channels experience constant variations resulting from channel fading. In an extreme case, the congestion control algorithm of TCP does not work at all in the wireless environment since it treats packet loss as network congestion by assuming a loss-free channel condition [4]. Thus, the traditionally-layered architecture won't bring desirable quality-of-service (QoS) for the wireless environment, especially for delay-sensitive applications such as real-time multimedia streaming services. For example, in multimedia applications, the behavior of video encoding at the application layer and the modulation and coding at the physical layer can both affect the packet loss rate (PLR). Without knowledge of each other, the system cannot utilize their interactions, which will affect the overall performance.

The constantly changing channel quality in wireless environments can actually be treated as a special kind of "context" information [5]. Our research indicates that dynamic adaptation on channel variations can lead to performance improvement in wireless multimedia networks [6]. Inspired by this, we have found out that a lot of other "context" data can also be used in the same way, which include location, available energy, user preference, weather information, geospatial data, etc. All these context data can thus be integrated into an adaptive wireless multimedia communications system based on context-aware services [7], which not only improves the user-perceived quality for the end users, but also provides better services.

In all, multimedia communications over wireless networks call for new design, development and methodologies to improve video quality against heterogeneous properties, limited resources and changing context to provide a higher standard of user satisfaction. Aiming at this, we have conducted a wide variety of research that involves different innovative techniques, covering joint source coding and network-supported distributed error control, video content-aware communications, cross-layer optimization and context-aware multimedia communications to optimize the system performance of delay-sensitive multimedia communications over wireless networks. Specifically, how to efficiently decrease video distortion and thus improve the video performance perceived by end users under the constraints of limited resources, such as bandwidth, delay deadline, and different application scenarios, is the issue that is addressed in this dissertation.

1.3 Outline of this Dissertation

The remainder of this dissertation is organized as follows:

In Chapter 2, we describe some of the research foundations. In our research, we use the state-of-the-art H.264/AVC as the video codec. So we first introduce some background knowledge of H.264 in Chapter 2. We also discuss the concept of video distortion and how to accurately calculate the estimated video distortion using Recursive Optimal Per-pixel

Estimate (ROPE) [8]. Furthermore, link adaption and Adaptive Modulation and Coding (AMC) techniques [9, 10] are studied.

In Chapter 3, we focus our attention on the problem of source coding and link adaptation for packetized video streaming in wireless multi-hop networks, where the end-to-end path is composed of heterogeneous physical links. We consider a system where source coding is employed at the video encoder by selecting the encoding mode of each individual macro-block, while error control is exercised through application-layer retransmissions at each media-aware network node. For this system model, the contribution of each communication link on the end-to-end video distortion is considered separately in order to achieve globally optimal source coding and ARQ error control. To reach the globally optimal solution, we formulate the problem of Joint Source and Distributed Error Control (JSDEC) and devise a low-complexity algorithmic solution based on dynamic programming.

In Chapter 4, we conduct research on video content-aware analysis and its application in a wireless e-healthcare system. Specifically, we propose a new accurate and cost-effective e-healthcare system of real-time human motion tracking over wireless networks, where the temporal inter-frame relation, spectral and spatial inter-pixel dependent contexts of the video frames are jointly utilized to accurately collect and track the human motion regions by using a low-cost camera in an ordinary markerless environment. On the basis of this video content-aware analysis, the extracted human motion regions are coded, transmitted and protected in video encoding with a higher priority against the insignificant background areas. Furthermore, the encoder behavior and the adaptive modulation and coding scheme are jointly optimized in a holistic way to achieve highly-improved overall video quality in wireless networks.

In Chapter 5, we present our research on cross-layer optimization to improve video performance for wireless multimedia networks. Specifically, the application scenario here is a P2P network. Considering the tightly-coupled relationship between P2P overlay networks and the underlying networks, we propose a distributed utility-based scheduling algorithm

on the basis of a quality-driven cross-layer design framework to jointly optimize the parameters of different network layers to achieve highly-improved video quality for P2P video streaming services over wireless networks. In this chapter, the distributed utility-based P2P scheduling algorithm is first presented and its essential part is formulated into a cross-layer based distortion-delay optimization problem distributed to each neighboring node, where the expected video distortion is minimized under the constraint of a given packet playback deadline to select the optimal combination of system parameters residing at different network layers. Specifically, encoding behaviors, network congestion, ARQ and modulation and coding are jointly considered during the cross-layer optimization. The distributed optimization running at each peer node adopted in the proposed scheduling algorithm can greatly reduce the computational complexity at each node.

In Chapter 6, we expand our dynamic adaptation to other "context" information. Here, we build an adaptive wireless multimedia system with context-awareness using ontology-based models. The proposed system can dynamically adapt its behaviors according to changes in various context data. First, it chooses the appropriate video content based on the retrieved static context data such as user's profile, location, time, and weather forecast. Then, media adaptation is performed to greatly improve the video quality perceived by the end users by adapting to various context data such as varying wireless channel quality, available energy of the end equipment, network congestion and application Quality of Services (QoS).

Finally, Chapter 7 summarizes the contributions and presents the future research topics, thus concluding this dissertation.

Chapter 2

Research Foundations

In this chapter, we discuss the research foundations, including H.264/AVC video codec, the calculation of video distortion, link adaption and adaptive modulation and coding (AMC).

2.1 Video Codec and Expected Video Distortion

H.264/AVC is one of the most advanced video formats for recording, compression, and distribution of high definition video. It is an industrial standard for video compression, the process of converting digital video into a format that takes up less capacity for storing or transmitting. Video compression (or video coding) is an essential technology for applications such as digital television, DVD video, mobile TV, video-conferencing and Internet video streaming.

A video encoder converts video into a compressed format, while a video decoder converts compressed video back into an uncompressed format. Figure 2.1 shows the encoding and decoding processes that are covered by the H.264/AVC standard. Specifically, an H.264 video encoder carries out prediction, transform and encoding processes to produce a compressed H.264 bitstream. An H.264 video decoder carries out the complementary processes of decoding, inverse transform and reconstruction to produce a decoded video sequence [1].

2.1.1 Video Encoder

The video encoder processes a frame of video in units of Macroblocks (MBs), where a Macroblock is defined as 16×16 displayed pixels. It forms a prediction of the macroblock based on previously-coded data, either from the current frame (intra prediction) or from

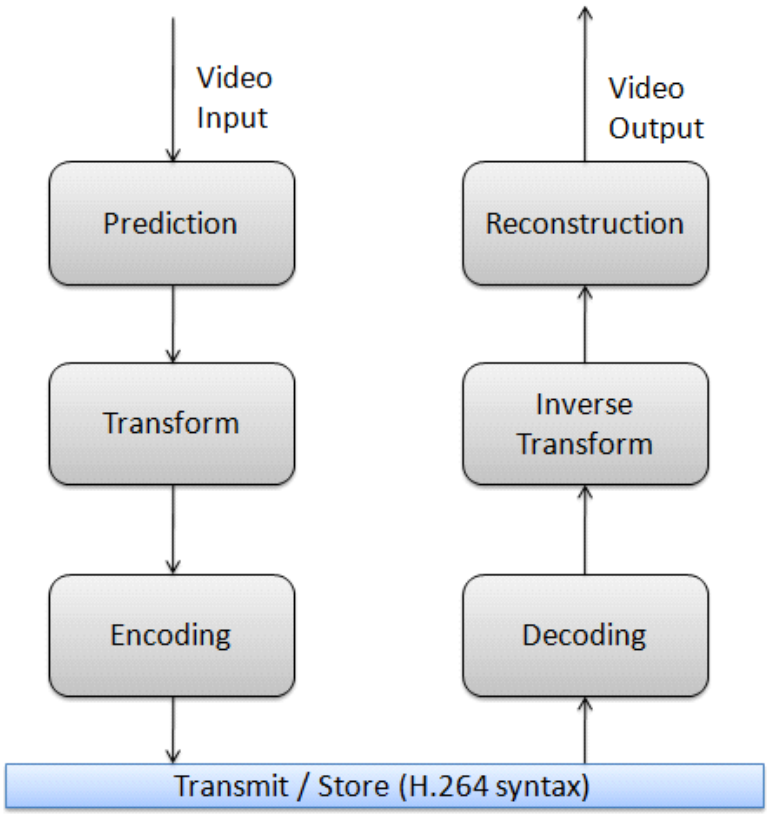


Figure 2.1 : The overall H.264 video encoding and decoding processes.

other frames that have already been coded and transmitted (inter prediction). These two prediction modes are the two mostly commonly used prediction methods in H.264/AVC standards. The encoder subtracts the prediction from the current macroblock to form a residual [11].

Intra prediction uses 16×16 and 4×4 block sizes to predict the macroblock from the surrounding, previously-coded pixels within the same frame as shown in Figure 2.2. On the other hand, inter prediction uses a range of block sizes (from 16×16 down to 4×4) to predict pixels in the current frame from similar regions in previously-coded frames, which is shown in Figure 2.3.

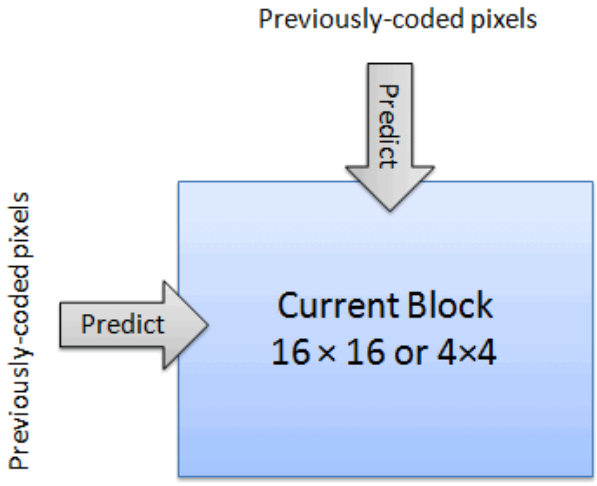


Figure 2.2 : Illustration of H.264 Intra Prediction.

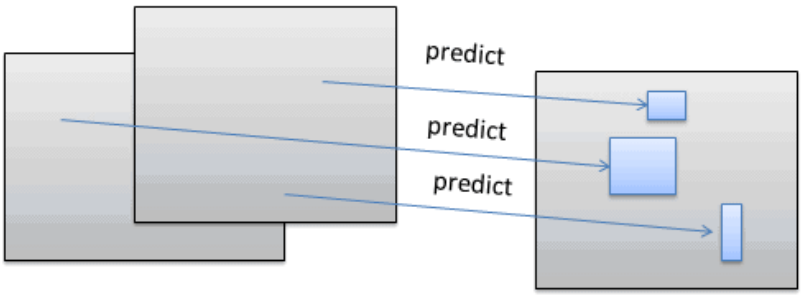


Figure 2.3 : Illustration of H.264 Inter Prediction.

A block of residual samples is transformed by using a 4×4 or 8×8 integer transform, an approximate form of the Discrete Cosine Transform (DCT) [1, 11]. The transform outputs a set of coefficients, each of which is a weight value for a standard basis pattern. When combined, the weighted basis patterns re-create the block of residual samples. Figure 2.4 shows how the inverse DCT creates an image block by weighting each basis pattern according

to a coefficient value and combining the weighted basis patterns [1].

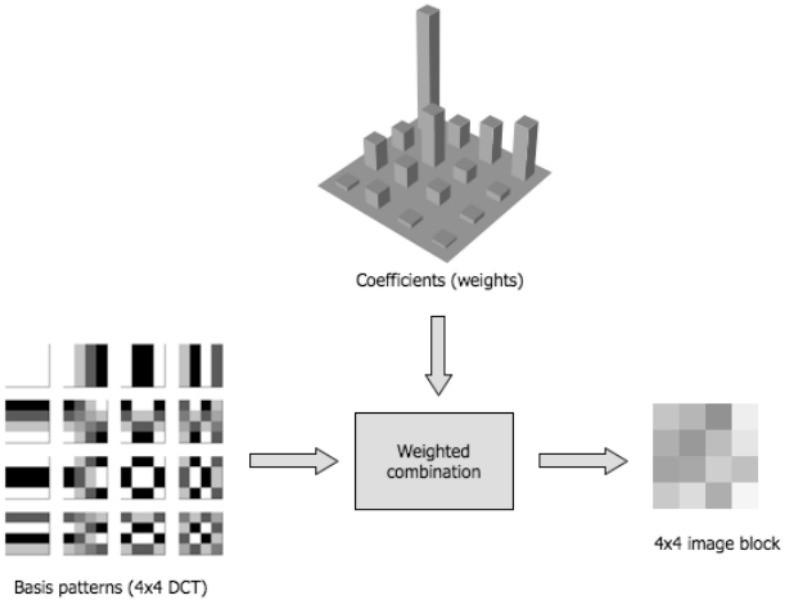


Figure 2.4 : Illustration of H.264 inverse transform: combining weighted basis patterns to create a 4x4 image block. [1]

The output of the transform, a block of transform coefficients, is quantized, i.e., each coefficient is divided by an integer value. Quantization reduces the precision of the transform coefficients according to a quantization parameter (QP). Typically, the result is a block in which most or all of the coefficients are zero, with a few non-zero coefficients. Setting QP to a high value means that more coefficients are set to zero, resulting in high compression at the expense of poor decoded image quality. Setting QP to a low value means that more non-zero coefficients remain after quantization, leading to better decoded image quality but lower compression [1, 2].

2.1.2 Video Decoder

The H.264 decoder is composed of several steps including rescaling, inverse transform and reconstruction.

A video decoder receives the compressed H.264 bitstream, decodes each of the syntax elements and extracts the information described above (quantized transform coefficients, prediction information, etc). This information is then used to reverse the coding process and recreate a sequence of video images.

Then, the quantized transform coefficients are re-scaled. Each coefficient is multiplied by an integer value to restore its original scale. An inverse transform combines the standard basis patterns, weighted by the re-scaled coefficients, to re-create each block of residual data. These blocks are combined together to form a residual macroblock.

Then, for each macroblock, the decoder forms an identical prediction to the one created by the encoder. The decoder adds the prediction to the decoded residual to reconstruct a decoded macroblock which can be displayed as part of a video frame.

2.1.3 Expected Video Distortion

In this research, the first challenging task is to derive an accurate model for the expected video distortion that can be used as the objective function in wireless multimedia communications networks.

Many research efforts have been developed to deal with distortion estimation for hybrid motion-compensated video coding and transmission over lossy channels [8, 12–14]. In this type of video coders, each video frame is represented in block-shaped units of the associated luminance and chrominance samples (16×16 pixel region) called macroblocks (MBs). In the H.264 codec, macroblocks can be both intra-coded or inter-coded from samples of previous frames [2]. Intra-coding is performed in the spatial domain, by referring to neighboring samples of previously coded blocks which are to the left and/or above the block to be predicted. Inter-coding is performed with temporal prediction from samples of previous frames.

It is evident that many coding options exist for a single macroblock, and each of them provides different rate-distortion characteristics. In this work, only pre-defined macroblock

encoding modes are considered, since we want to apply error resilient source coding by selecting the encoding mode of each particular macroblock. It is crucial to allow the encoder to trade off bit rate with error resiliency at the macroblock level. The Recursive Optimal Per-pixel Estimate (ROPE) algorithm has been adopted to calculate distortion recursively across frames [15, 16], meaning that the estimation of the expected distortion for a frame currently being encoded is derived by considering the total distortion introduced in previous frames.

Here, we consider a N-frame video clip $\{f_1, ..., f_n, ..., f_N\}$. During encoding, each video frame is divided into 16 $\times$ 16 macroblocks (MB), which are numbered in scan order. In our implementation, the packets are constructed such that each packet consists of a row of MBs and is independently decodable. The terms of row and packet sometimes are interchangeably used. When a packet is lost during transmission in the network, we use the temporal-replacement error concealment strategy. Therefore, the motion vector of a missing MB is estimated as the median of motion vectors of the nearest three MBs in the preceding row. If the previous row is lost too, the estimated motion vector is set to zero. The pixels in the previous frame, which are pointed by the estimated motion vector, are used to replace the missing pixels in the current frame.

Let I be the total number of packets in one video frame, and J the total number of pixels in one packet. Let us denote by $f_{n,i}^j$ the original value of pixel j of the ith packet in frame n, $\hat{f}_{n,i}^j$ the corresponding encoder reconstructed pixel value, $\tilde{f}_{n,i}^j$ the reconstructed pixel value at the decoder, $E[d_{n,i}^j]$ the expected distortion at the receiver for pixel j of the ith packet in frame n. We use the expected Mean-Squared Error (MSE) as the distortion metric, which is commonly used in the literature [8]. Then the total expected distortion $E[D]$ for the entire video sequence can be calculated by summing the expected distortion of all the pixels

$$E[D] := \sum_{n=1}^{N} \sum_{i=1}^{I} \sum_{j=1}^{J} E[d_{n,i}^j] \tag{2.1}$$

where

$$\begin{aligned} E[d_{n,i}^{j}] &= E[(f_{n,i}^{j} - \tilde{f}_{n,i}^{j})^2] \\ &= (f_{n,i}^{j})^2 - 2f_{n,i}^{j}E[\tilde{f}_{n,i}^{j}] + E[(\tilde{f}_{n,i}^{j})^2] \end{aligned} \tag{2.2}$$

Since $\tilde{f}_{n,i}^{j}$ is unknown to the encoder, it can be considered as a random variable. To compute $E[d_{n,i}^{j}]$, the first and second moments of $\tilde{f}_{n,i}^{j}$ are needed. From the works [17–19], the equations to calculate the first and second moments based on intra-coded MBs (2.3) and inter-coded MBs (2.4) are given below:

$$\begin{aligned} E[(\tilde{f}_{n,i}^{j})^2](I) &= (1-\rho_{n,i})(\hat{f}_{n,i}^{j})^2 \\ &+ \rho_{n,i}(1-\rho_{n,i-1})E[(\tilde{f}_{n-1,u}^{k})^2] \\ &+ \rho_{n,i}\rho_{n,i-1}E[(\tilde{f}_{n-1,i}^{j})^2] \end{aligned} \tag{2.3}$$

where if the current packet is lost and the previous packet is received, the concealment motion vector associates the pixel j of packet i in the current frame with the pixel k of packet u in the previous frame. For inter-coded MBs, let us assume that the pixel j of the row i in the current frame n is predicted from the pixel m of the row l in the previous frame by the true motion vector during encoding. Therefore,

$$\begin{aligned} \mathcal{E}[(\tilde{f}_{n,i}^{j})^2](P) &= (1-\rho_{n,i})((\hat{\mathfrak{r}}_{n,i}^{j})^2 + 2\hat{\mathfrak{r}}_{n,i}^{j}E[\tilde{f}_{n-1,l}^{m}] \\ &+ E[(\tilde{f}_{n-1,l}^{m})^2]) + \rho_{n,i}\rho_{n,i-1}E[(\tilde{f}_{n-1,i}^{j})^2] \\ &+ \rho_{n,i}(1-\rho_{n,i-1})E[(\tilde{f}_{n-1,u}^{k})^2] \end{aligned} \tag{2.4}$$

where $\hat{\mathfrak{r}}_{n,i}^{j}$ is the quantized prediction residue. Thus, the expected distortion is accurately calculated by the ROPE algorithm [17–19] under instantaneous network conditions and can be used as the objective function for optimization.

It is worth mentioning that the calculation of $E[d_{n,i}^{j}]$ and $E[(\tilde{f}_{n,i}^{j})^2]$ depends on the used error concealment strategy as well as the packetization scheme. The only parameter that

these formulas require is the residual packet error rate ρ. This parameter is updated after each packet is encoded, since the individual contribution of each path is also continuously updated.

2.1.4 Computational Complexity

Adopting the ROPE algorithm, the expected video distortion can be precisely computed for every pixel. However, this advantage of precise distortion approximation comes at the cost of a modicum increase of computational complexity.

First, most of the computational overhead comes from the fact that the distortion and two moments of $\tilde{f}_n$ for both intra-mode and inter-mode of every pixel need to be calculated. As we discussed earlier, computational complexity is reduced due to the identical error concealment regardless of the macroblock encoding mode. For each pixel in an inter-coded MB, we need 16 addion/multiplication operations for calculating the moments of $\tilde{f}_n$, while for intra-coded MB, 11 addition/multiplication operations are necessary for the same operation. This computational complexity is comparable to the number of DCT operations [2]. Furthermore, the error concealment algorithm has to be implemented at the encoder for every block, which might bring additional computational overhead, depending on how sophisticated the concealment algorithm is. Besides, we also need to store two moments as two floating-point numbers for every pixel. However, this additional storage complexity is negligible considering modern high-performance computers.

2.2 Link Adaptation

Link adaptation, or adaptive coding and modulation, is a method and strategy used in wireless communications to denote the matching of the modulation, coding and other signal and protocol parameters to the conditions on the radio link such as the path loss, the interference due to signals coming from other transmitters, the sensitivity of the receiver and the available transmitter power margin.

Adaptive modulation systems invariably require some channel state information at the transmitter. This could be acquired in time division duplex systems by assuming the channel from the transmitter to the receiver is approximately the same as the channel from the receiver to the transmitter. Alternatively, the channel knowledge can also be directly measured at the receiver, and fed back to the transmitter. Adaptive modulation systems improve rate of transmission, and/or bit error rates, by exploiting the channel state information that is present at the transmitter. Over fading channels which model wireless propagation environments, adaptive modulation systems exhibit great performance enhancements compared to systems that do not exploit channel knowledge at the transmitter.

Adaptive modulation and coding (AMC) has been advocated to enhance the throughput of future wireless communication systems at the physical layer [9, 10]. With AMC, the combination of different constellations of modulation and different rates of error-control codes are chosen based on the time-varying channel quality. For example, in good channel conditions, AMC schemes with larger constellation sizes and higher channel coding rates can be adopted to guarantee the required packet error rate, which means that AMC can effectively decrease the transmission delay, while satisfying the constraint of packet loss rate. Each AMC mode consists of a pair of modulation scheme a and FEC code c as in 3GPP, HIPERLAN/2, IEEE 802.11a, and IEEE 802.16 standards [10, 20, 21]. Furthermore, we adopt the following approximated Bit Error Rate expression:

$$p_m^e(\gamma) = \frac{a_m}{e^{\gamma \times b_m}} \tag{2.5}$$

where m is the mode index and γ is the received SNR. Coefficients a_m and b_m are obtained by fitting (5.4) to the exact BER shown in Table 2.1.

Table 2.1 : Available AMC Modes at the Physical Layer

AMC Mode (m)	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$	$m = 6$
Modulation	BPSK	QPSK	QPSK	16-QAM	16-QAM	64-QAM
Coding Rate (c_m)	1/2	1/2	3/4	9/16	3/4	3/4
R_m (bits/sym.)	0.50	1.00	1.50	2.25	3.00	4.50
a_m	1.1369	0.3351	0.2197	0.2081	0.1936	0.1887
b_m	7.5556	3.2543	1.5244	0.6250	0.3484	0.0871

Chapter 3

Joint Source Coding and Distributed Error Control

In this chapter, we describe our research on the optimization of video quality for real-time video communication services by using joint source coding and network-supported distributed error control in heterogeneous wireless networks.

3.1 Introduction

Real-time video communication services such as video telephony, video conferencing, video gaming, and mobile TV broadcasting are considered very important applications for wireless multi-hop networks. However, the dissemination of pre-compressed or real-time video over wireless networks is characterized by several problems [3]. Most of the problems of media streaming applications can be identified as the lack of bandwidth guarantees, random packet losses, and delay jitter. Even though bandwidth is increasingly available for end users, high-quality multimedia streaming applications are still bandwidth-demanding. Furthermore, delay jitter and packet losses make the problem of video delivery more challenging, especially in multi-hop wireless mesh networks. Most of the real-time video encoding and streaming applications (e.g., video conferencing) usually employ a wide range of intelligent techniques at the endpoints in order to enhance the user-perceived quality [22].

Nonetheless, a particular characteristic of existing and emerging networks, usually overlooked by video streaming applications, is that an end-to-end path consists of several interconnected physical networks which are heterogeneous and have asymmetric properties in terms of throughput, delay, and packet loss. Congested last-mile wireline links (e.g., DSL links) or wireless access networks (e.g., WiFi) are some of the real-life examples. In such networks, employing measurements or congestion control in the end-to-end fashion might

not provide end-systems with a correct status of the network characteristics [23, 24]. For example, some wireless hops in the end-to-end path as shown in Figure 3.1 may face bad channel conditions. For a video streaming application, a pure end-system implementation might unnecessarily limit the choices of the video encoder or the streaming algorithms with respect to the optimal streaming rate and the error control strategy along the end-to-end path. Therefore, proxy-based solutions are usually proposed to deal with this situation. However, for real-time video encoding systems, the majority of the current proxy-based streaming approaches have mainly focused on wireless networks only at the last hop [22, 25]. Moreover, even for wireline networks, most of the current research only makes use of a single proxy [26, 27].

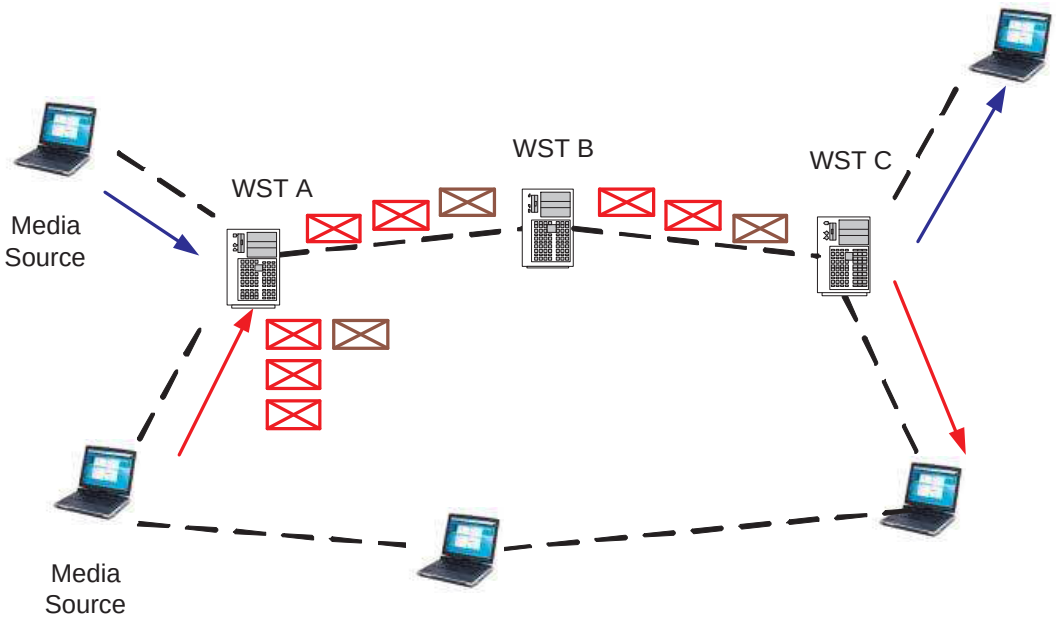


Figure 3.1 : The general scenario addressed by this chapter consists of multiple wireless stations (WSTs) that generate real-time encoded unicast bitstreams, which are subsequently forwarded by the WSTs in the end-to-end wireless path.

To address the aforementioned problems in a systematic fashion, in this chapter we propose a novel mechanism to integrate into a joint optimization framework the parameters

that affect the encoding and transmission of a video stream. Toward this end, we focus on generalizing the approach on the basis of Joint Source and Channel Coding (JSCC), which can integrate both network/transport parameters and the source characteristics for improving the system performance [28–30]. The objective of JSCC is to distribute the available channel rate between source and channel bits so that the decoder distortion is minimized. For JSCC to be optimal, it is imperative that the sender has an accurate estimation of the channel characteristics, such as available bandwidth, round-trip-time (RTT), packet loss, burst/random errors, congestion, and so on. The more accurate this information is, the more efficient the resource allocation with JSCC will be. However, the situation that we described will essentially translate into sub-optimal JSCC allocation over several physical channels [24, 31].

We have recognized the importance of this situation in [23], where we showed that for streaming fine-granularity scalable (FGS) video, the problem of distributed source/channel coding actually corresponds to a flow control problem. In this chapter, we generalize this problem since we focus on the more fundamental issues of real-time video encoding and transmission over multiple tandem-connected network elements. We use the term *proxy* to refer to such network nodes that are able to perform media-aware tasks, like error control. We examine the online adaptation of the encoding mode of individual macroblocks (source coding) and the level of error control. To emphasize the difference with JSCC, our proposed algorithm is called Joint Source and Distributed Error Control (JSDEC). In this chapter, we regard channel error control in terms of retransmission as one form of channel coding. This is reasonable because, in general, retransmission is also a mechanism for error correction. To the best of our knowledge, this is the first work that considers the problem of exercising error resilient source coding and distributed error control in a joint optimization framework. Our target application scenario is video-conferencing, as Figure 3.1 indicates. Note that we are not concerned with trans-rating or trans-coding techniques, which have been heavily studied in the literature.

3.2 Joint Source and Distributed Error Control for Video Streaming Applications

To illustrate more clearly the main concept behind this work, we present in Figure 3.2 the R-D performance of a JSCC video streaming system under three different multihop wireless channel realizations. The curves represent the convex-hull [32] of all feasible (rate, distortion) points for the same video sequence. More specifically, the two lower curves correspond to video transmission over two different channels that have available channel rates R_{T1} and R_{T2}, respectively. When each of these channels is considered separately, the tuples of the optimal source-channel rates that are derived from the JSCC algorithm are $\{R_{S1}, (R_{T1} - R_{S1})\}$ and $\{R_{S2}, (R_{T2} - R_{S2})\}$ respectively. However, when the channels are connected in tandem, the form of the operational joint rate-distortion curve is different. The most important implication is that the globally optimal source-channel bit allocation is not the same anymore. If JSCC is still employed at the encoder, and we allow the sender to probe the end-to-end path, the bottleneck link in this case would determine which is the "optimal" source-channel rate [24, 31]. In case there is spare bandwidth at the second hop, error protection can be increased, and this can be applied at the discretion of existing proxies that reside between the two links. Although simple, this solution is still sub-optimal, as is demonstrated in the figure, because the server can only select R_{S1} as the optimal streaming rate. Therefore, an optimal streaming resource allocation algorithm should be proposed to be able to select the globally optimal R-D operating point. Thus, JSDEC is developed in this chapter to find this point by achieving the real "optimal" source-to-channel ratio for any given available channel rate. To calculate the solution to this problem, the encoder must have knowledge of both the available channel rate on the next hop (i.e. R_{T2}) and Packet Erasure Rate(PER) (i.e. ϵ_2) of that hop. Another way to rephrase the problem at hand is the following: *The encoder should calculate source encoding options, enabling globally optimal rate allocation between source and channel bits for all the hops considered*

in the end-to-end path.

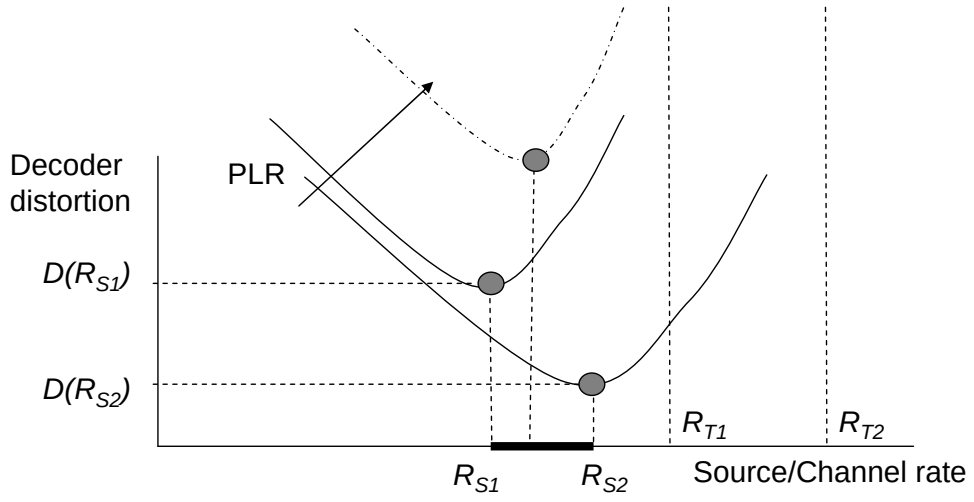


Figure 3.2 : Convex hull of the feasible RD points for encoded video transmission for each hop individually, and connected in tandem.

3.3 Video Encoder

Both the distortion estimation and resource allocation algorithms are implemented at the streaming server and encoder. The encoder executes the algorithm for joint source and channel allocation by striving to minimize the global distortion. To achieve this, the encoder calculates for each macroblock the estimated source distortion and channel distortion. The average channel distortion is calculated for each packet erasure channel h, based on the reported feedback from the corresponding proxy (packet erasure rate ϵ_h). By using these estimates, the encoder calculates the optimal encoding mode μ for each individual macroblock (source coding), while the optimal number of retransmissions σ for the corresponding transport packet (channel coding) is calculated locally by each proxy. Therefore, source-channel coding is applied jointly, i.e. the encoder decides on the encoding mode while it also indirectly determines the optimal rate dedicated to channel coding with ARQ. An important

advantage of the Joint Source and Distributed Error Control algorithm is that the proxy does not require the notification of the optimal rate for channel coding. Furthermore, each proxy decides individually the maximum number of allowed retransmissions for a particular video packet, since it can deduce the optimal channel coding rate. The relationships among encoder, proxy and ARQ in our proposed JSDEC system are illustrated in Figure 3.3.

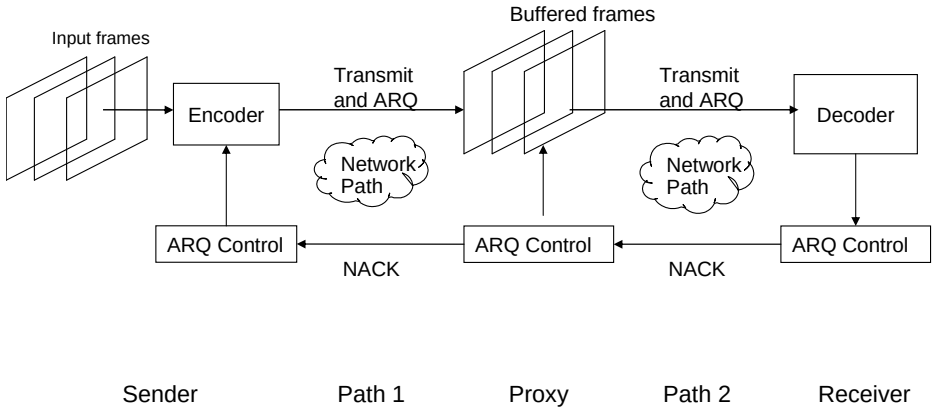


Figure 3.3 : Process of frame encoding and transmission at the sender/encoder and an intermediate ARQ proxy.

3.4 The Error Control Proxy

Despite the importance of the distortion estimation loop, the basic component of our architecture is the error control proxy. The functionality realized at the proxy is an application-layer delay-constrained ARQ error control algorithm for the video packets. ARQ is exercised on a local scope only between two successive nodes (i.e., proxy, sender, or receiver). A negative acknowledgment scheme is used for this purpose. In practice, the proxy is a network node that terminates RTP sessions [33]. This is necessary since it has to monitor the sequence number space of each RTP flow. It is worth mentioning that RTP/RTCP was originally designed for Internet applications, assuming a much lesser varying channel envi-

ronment. Therefore, the RTCP reports are usually transmitted at relatively long intervals, leading to performance degradation when used in wireless networks due to the much more unstable channel quality.

Part of the available bandwidth at the proxy is allocated to forward incoming video packets. The remaining of the available bandwidth is used for retransmitting packets locally stored at the proxy. The simplicity of the delay-constrained ARQ algorithm means that the computational overhead is very low, which is a critical requirement in practice, since we envision that the ARQ proxy can be implemented as part of existing media-aware network nodes [33]. Note that other more advanced coding schemes could be adopted at the intermediate nodes (e.g., Raptor codes [34]).

3.4.1 Packet Loss Rate

Based on the above design, we can calculate the impact of the ARQ algorithm at each proxy on the aggregate packet loss rate and the latency experienced by the transmitted video packets.

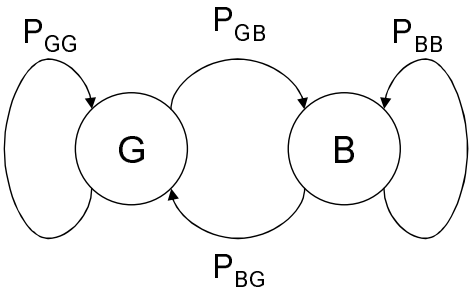


Figure 3.4 : Two-state Markov chain channel model for a single network hop.

The communications network considered in this chapter is packet-switched, and the adopted packet loss model is an "erasure channel". We consider a Markov chain for characterizing transitions from good to bad states of each individual channel as shown in Figure 3.4.

For this channel model, if the packet erasure rate of hop h at the network layer is ϵ_h and the average number of retransmissions for a particular video transport packet i is $m_{h,i}$, then the resulting residual packet loss rate is $\epsilon_h^{m_{h,i}}$. Sometimes, packet loss is not caused by packet erasure ($\epsilon_h^{m_{h,i}}$), but rather the excessive delay τ_h for the hop h. If the transmission delay for hop h is L_h, the above probability can be expressed as $P_r\{L_h > \tau_h\}$. Then the overall packet loss rate due to packet erasure and packet delay is:

$$\rho_{h,i} = \epsilon_h^{m_{h,i}} + (1 - \epsilon_h^{m_{h,i}}) P_r\{L_h > \tau_{h,i}\} \tag{3.1}$$

An important parameter of this formula is the actual distribution of retransmissions for a given permissible range. This value can be calculated using the adopted channel model. The probability of k retransmissions for a successful delivery of packet i is given by:

$$\pi(k, \sigma_{h,i}) = \frac{(1 - \epsilon_h)\epsilon_h^k}{1 - \epsilon_h^{\sigma_{h,i}+1}}. \tag{3.2}$$

where $\sigma_{h,i}$ is the maximum transmissions for source packet i on link h. Given $\sigma_{h,i}$ and ϵ_h, we can also calculate the average number of retransmissions per packet i for hop h as:

$$m_{h,i} = \frac{1 - \epsilon_h^{\sigma_{h,i}+1}}{1 - \epsilon_h} \tag{3.3}$$

Equation (3.2) accounts for the two possible reasons of packet loss in the network, namely excessive delay and channel erasure. In literature, this method of calculating the residual PER was first introduced in the seminal work of [35]. Regarding Equation (3.3), it captures the channel erasure effect as a Bernoulli distribution as well as the effect of truncated ARQ by $\sigma_{h,i}$, which is also widely used in the literature [36, 37]. Therefore, the *average* packet transmission delay for each hop h is given by

$$L_h(i) = \sum_{k=0}^{\sigma_{h,i}} \pi(k, \sigma_{h,i})[k * RTT_h + FTT_h], \tag{3.4}$$

where FTT_h and RTT_h are the forward and the round trip delay of hop h, respectively. Within the proposed framework, error control (i.e., channel coding) is exercised locally at

each proxy by enforcing the maximum number of retransmissions for each specific packet and hop $\sigma_{h,i}$.

Therefore, we can express the overall end-to-end packet loss rate for H tandem proxies as

$$\rho_i = 1 - \prod_{H}(1 - \rho_{h,i}). \tag{3.5}$$

3.4.2 Network Delay Model

The only parameter which needs to be estimated now is the forward delay of each of the tandem-connected links. In this chapter, we model the one-way network delay as a Gamma distribution, since this distribution captures the main reason of network packet delays, which is due to buffering in the wireless nodes [38]. The probability density function of the Gamma distribution with rightward shift γ, parameters ν and α is given by

$$f_{L_N}(t|rcvd) = \frac{\alpha}{\Gamma(\nu)}(a(t-\gamma))^{(\nu-1)}e^{(-\alpha(t-\gamma))} \text{ for } t \geq \gamma \ , \tag{3.6}$$

where ν is the number of end-to-end hops, γ is the total end-to-end processing time, and α is the waiting time at a router following the exponential distribution and can be modeled as an M/M/1 queue. From the networking aspect, this model can be perceived as if the packet traverses a network of ν M/M/1 queues, where each of them has a mean processing time plus waiting time $\frac{\gamma}{\nu} + \frac{1}{\alpha}$ and variance $\frac{1}{\alpha^2}$. Therefore, the forward trip delay has a mean $\mu_F = \gamma + \frac{\nu}{\alpha}$ and variance $\sigma_F = \frac{\nu}{\alpha^2}$. Both α and ν are calculated by periodically estimating the mean and variance of the forward and backward trip delays at the receiver and transmitting their values to the sender [35].

3.5 Optimization Framework for Source Coding and Distributed Channel Coding

In this section, we jointly optimize the source coding and distributed error control parameters within our proposed framework. The goal is to minimize the perceived video distortion

for given available link capacity and packet error rate. First, we formulate the problem as a constrained minimum distortion problem. Then, we give the optimal solution using dynamic programming.

3.5.1 Problem Formulation

The implemented ROPE algorithm can accurately estimate the expected distortion. Meanwhile, our rate-distortion optimization framework takes into account the expected distortion due to compression at the source, error propagation across frames, and error concealment induced by packet losses at the erasure channel that each proxy is attached to. We incorporate the proposed distortion model into a rate-distortion optimization framework, which will be used at the encoder for determining the encoding mode of each individual MB and the corresponding optimal protection with ARQ. In this chapter, we consider the bandwidth constraint and the packet loss rate of each link. Therefore, the distortion minimization problem has to satisfy multiple channel rate constraints. Also, a group of MBs can be coded and packed into the same transport packet, which means that this group of MBs will be subject to the same error protection decisions throughout the existing proxies in the end-to-end path.

In the following, we will define the rate-distortion optimization problem. Let $\vec{\mu}(n)$ and $\vec{\sigma}(n)$ denote the vectors of source coding and channel error control parameters for the n-th frame, respectively. Then, the end-to-end distortion minimization problem is

$$\begin{aligned} \min_{\mu\in\mathbf{U},\sigma\in\mathbf{\Sigma}} & \sum_{n=1}^{N}\sum_{i=1}^{I} E[D_{n,i}](\vec{\mu}(n),\vec{\sigma}(n)) \\ \text{s.t. } & R_S + R_{C_h} \le R_{T_h} \quad \forall h \in H \end{aligned} \tag{3.7}$$

where $\mathbf{U}$ and $\mathbf{\Sigma}$ are the entire sets of available vectors of $\vec{\mu}(n)$ and $\vec{\sigma}(n)$ respectively, and H is the number of tandem-connected nodes. R_S, R_C denote the rates allocated to source coding and error control, respectively. R_T is the total available channel rate. Furthermore,

the available channel rate R_{T_h} can be denoted in terms of channel bandwidth W_{T_h}:

$$R_{T_h} = W_{T_h} log_2(1 + \xi) \tag{3.8}$$

where ξ is the channel SNR. The rate constraint for retransmitted packets must be satisfied by every proxy in the end-to-end path. Recall that the average number of retransmissions at each proxy is $m_{h,i}$ and the residual network packet loss rate is $\rho_{h,i}$. Therefore, the bit rate constraint that needs to be satisfied for frame n at each proxy h is

$$\sum_{i=1}^{I} R_s(\vec{\mu}(n)) \cdot m_{h,i} + \sum_{i=1}^{I} S_i \cdot m_{h,i} \leq R_{T_h} \quad \forall h \in H \tag{3.9}$$

where S_i is the size of the parity bits in packet i. The first term in Equation (3.9) denotes the source coding rate for frame n, and the second term is the channel rate estimation for the retransmitted packets of frame n. What we have achieved in (3.9) is to decouple the transmission rate constraints for the paths connected in tandem. To facilitate further algebraic manipulations, we denote the left-side part in (3.9) as R_h that represents the total bit rate of source and channel coding (i.e, $R_h = R_S + R_{C_h}$). Therefore, the objective function and the corresponding constraint of (3.7) can be denoted as

$$\begin{aligned} \min_{\mu \in \mathbf{U}, \sigma \in \mathbf{\Sigma}} & \sum_{n=1}^{N} \sum_{i=1}^{I} E[D_{n,i}](\vec{\mu}(n), \vec{\sigma}(n)) \\ & \text{s.t. } R_h \leq R_{T_h} \quad \forall h \in H \end{aligned} \tag{3.10}$$

3.5.2 The Optimal Solution for the Minimum Distortion Problem

For simplicity, let us denote the parameter vector of packet i in video frame n as $\mathcal{V}_w := \{\vec{\mu}(n), \vec{\sigma}(n)\}$, where $w(1 \leq w \leq N \times I)$ is the index of the packet i of the whole video clip. Clearly, in (3.10), any selected parameter vector $\mathcal{V}_w$ resulting in the total bit rate of source and channel coding to be greater than R_{T_h} is not in the optimal parameter vector $\mathcal{V}_w^* := \{\vec{\mu}^*(n), \vec{\sigma}^*(n)\}$. Therefore, we can make use of this fact by redefining the distortion

as follows:

$$E[D_{n,i}] := \begin{cases} \infty, & R_h > R_{T_h} \\ E[D_{n,i}], & R_h \leq R_{T_h}. \end{cases} \tag{3.11}$$

In other words, the average distortion for a packet with bit rate larger than the maximum allowable bit rate is set to infinity, meaning that a feasible solution, as defined in (3.10), will not result in any bit rate greater than R_{T_h}. Therefore, the minimum distortion problem with rate constraint can be transformed into an unconstrained optimization problem.

Furthermore, most decoder concealment strategies introduce dependencies between packets. For example, if the concealment algorithm uses the motion vector of the MB received earlier to conceal the lost MB, then it would cause the calculation of the expected distortion of the current packet to depend on its previous packet. Without losing generality, we assume that the current packet will depend on the latest a packets ($a \geq 0$), due to the concealment strategy. To solve the optimization problem, we define a cost function $\mathcal{C}_i(\mathcal{V}_{i-a}, \ldots, \mathcal{V}_i)$ which represents the minimum average distortion up to and including the packet i, given that $\mathcal{V}_{i-a}, \ldots, \mathcal{V}_i$ are the decision vectors for the packets $(i-a), \ldots, i$. Let $\mathcal{P}$ be the total packet number of the video clip, and we have $\mathcal{P} := N \times I$. Therefore, $\mathcal{C}_{\mathcal{P}}(\mathcal{V}_{\mathcal{P}-\mathbf{a}}, \ldots, \mathcal{V}_{\mathcal{P}})$ represents the minimum total distortion of the whole video clip. Thus, solving (3.10) is equivalent to solve

$$\underset{\mathcal{V}_{\mathcal{P}-a}, \ldots, \mathcal{V}_{\mathcal{P}}}{\text{minimize}} \ \mathcal{C}_{\mathcal{P}}(\mathcal{V}_{\mathcal{P}-a}, \ldots, \mathcal{V}_{\mathcal{P}}). \tag{3.12}$$

The key observation for deriving an efficient algorithm is the fact that given $a+1$ control vectors $\mathcal{V}_{i-a-1}, \ldots, \mathcal{V}_{i-1}$ for the packets $(i-a-1), \ldots, (k-1)$, and the cost function $\mathcal{C}_{i-1}(\mathcal{V}_{i-a-1}, \ldots, \mathcal{V}_{i-1})$, the selection of the next control vector $\mathcal{V}_i$ is independent of the selection of the previous control vectors $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_{i-a-2}$. This means that the cost function can be expressed recursively as

$$\begin{aligned} \mathcal{C}_i\left(\mathcal{V}_{i-a}, \ldots, \mathcal{V}_i\right) = & \underset{\mathcal{V}_{i-a-1}, \ldots, \mathcal{V}_{i-1}}{\text{minimize}} \left\{\mathcal{C}_{i-1}\left(\mathcal{V}_{i-a-1}, \ldots, \mathcal{V}_{i-1}\right)\right. \\ & \left. + E[D_{n,i}]\right\}. \end{aligned} \tag{3.13}$$

This recursive representation of the cost function makes the future step of the optimization process independent from its past step, which consists of the foundation for dynamic programming.

The convexity of $E[D_{n,i}]$ is shown in Section 2.1.4 as well as in the literature [32]. Therefore, the problem at hand can now be converted into a graph theory problem of finding the shortest path in a directed acyclic graph (DAG) [39]. The computational complexity of the algorithm is $O(\mathcal{P} \times |\mathcal{V}|^{a+1})$ (where $|\mathcal{V}|$is the cardinality of $\mathcal{V}$), which depends directly on the value of a. For most cases, a is a small number, so the algorithm is much more efficient than an exhaustive search algorithm with exponential computational complexity. The exact value of a is decided by the encoder behavior. If resources allow, the proposed algorithm can always achieve the optimal solution. However, with the increase of a, the computational complexity increases exponentially. In this case, it is advisable to decrease the value to lower the complexity, which leads to the achievement of a sub-optimal solution.

3.6 Experimental Results

In this section, we design experiments to validate the performance of the proposed optimization algorithm. We evaluate the performance of our proposed system under time-varying network conditions in terms of both available channel rate and packet loss rate.

A topology similar to the one in Figure 3.1 is used for this experiment. The QCIF (176 $\times$ 144) sequence "Foreman" is adopted for real-time encoding with the H.264/AVC JM12.2 software [40]. The network topology is simulated using NS-2. All frames except the first are encoded as P frames. The target frame rate is set to 30 frames/second. Each link in the path set has an average SNR $\bar{\xi}$, and the instantaneous link quality is captured by the received SNR value ξ randomly produced from (3.14) with $\bar{\xi}$. We assume the channel is frequency flat, remaining time invariant during a packet while varying from packet to packet. We adopt the Rayleigh channel model to describe ξ statistically. The received SNR

ξ per packet is a random variable with a Probability Density Function (PDF):

$$p_{\xi}(\xi) = \frac{1}{\bar{\xi}} e^{-\frac{\xi}{\bar{\xi}}} \tag{3.14}$$

where $\bar{\xi} := E\{\xi\}$ is the average received SNR.

The propagation delay on each link is fixedly set to $10\mu s$. In the simulation, for the source coding parameter, we consider the quantization step size (QP) q. For comparison, we use the PSNR metric between the original sequence and the reconstructed sequence at the receiver. Since the duration of the sequence is short (300 frames), the initial startup delay at the decoder playback buffer is set to be five frames. In the case of a buffer underflow, re-buffering is performed until two more frames are received. We compare the proposed JSDEC system with a JSCC system where each proxy employs error control individually, without coordination with the encoder at the sender. In all experiments of this section, we set the number of frames stored at each proxy equal to four.

First, we present the experimental results for the proposed JSDEC framework, comparing with the existing JSCC when the packet loss rate is constant. The results for two tandem connected hops (i.e., one ARQ proxy) when $W_{T2} = 0.3MHz$ and $R_{T2} = 0.6MHz$ can be seen in Figure 3.5 and Figure 3.6, respectively. We compare the performance difference of the JSCC and JSDEC algorithms while the feedback sent from the proxy back to the encoder is subject to a delay equal to the transmission delay of one frame. The two links are also attached to the same proxy which are configured with the same packet erasure rate of $\epsilon_1 = \epsilon_2 = 2\%$. The y-axis corresponds to PSNR (dB) while the x-axis corresponds to the available channel bandwidth on the first channel (i.e., W_{T1}).

In Figure 3.5, we fix the available channel bandwidth of the second hop to $0.3MHz$ (i.e., $W_{T2} = 0.3MHz$), while the available channel bandwidth of the first hop W_{T1} changes from $0.2MHz$ to $0.4MHz$ at a step size of $50KHz$. Our simulation results indicate that the proposed JSDEC system achieves the largest performance gain (about 6dB) at around $W_{T1} = W_{T2} = 0.3MHz$. When W_{T1} is either increased or reduced a little from $0.3MHz$, the

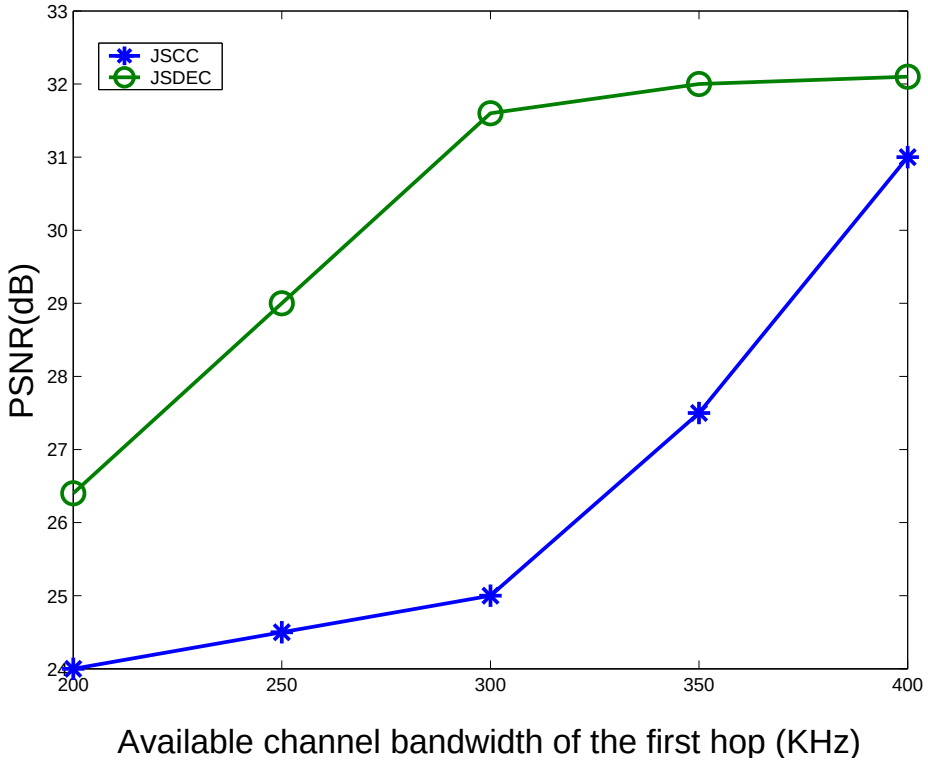


Figure 3.5 : Frame PSNR comparison between the proposed JSDEC framework and the systems using JSCC with constant packet loss rates. ($\epsilon_1 = \epsilon_2 = 2\%, W_{T2} = 0.3MHz$)

performance gain is reduced, but it is still significant. However, if the two available channel bandwidths differ much more, performance gain decreases significantly. For instance, when W_{T1} is at either $0.2MHz$ or $0.4MHz$, we only achieve 1-2dB PSNR performance increase. To explain this, when W_{T2} is fixed at $0.3MHz$, the much lower value of W_{T1} constrains the upper bound of the source encoding rate. This indirectly allows most of the available rate on the second channel R_{T2} to be used for error control and thus leads to minimization of the channel distortion. Similarly, if the value of W_{T1} is much higher than $0.3MHz$, the first channel introduces minimal channel distortion because of the significant spare bandwidth available for error control. Nevertheless, when the network does not operate in such highly asymmetric as far as channel rates are concerned, the considerable performance

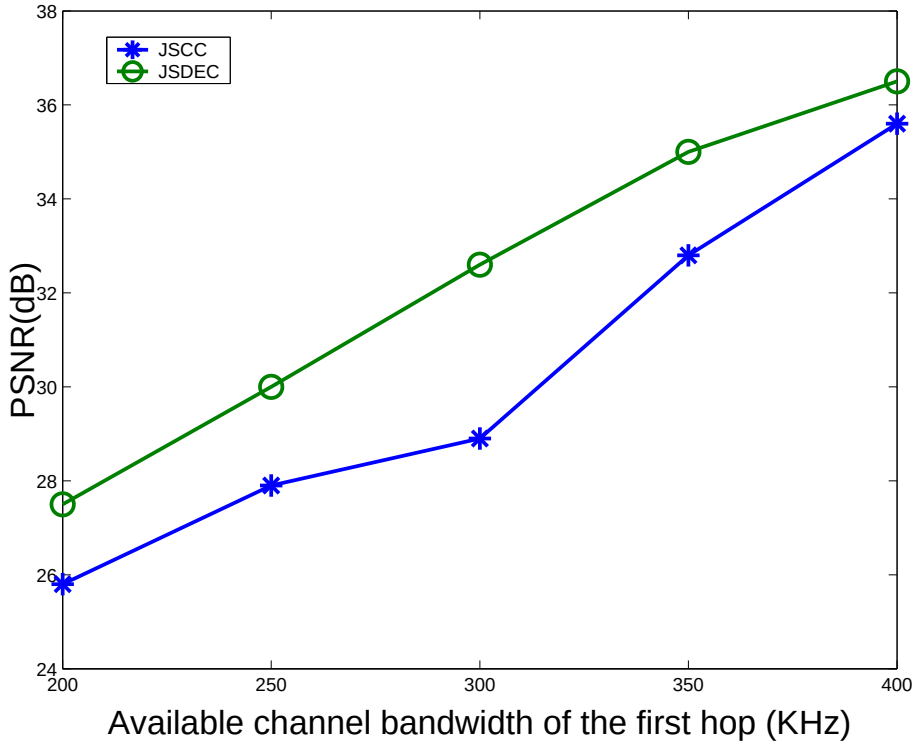


Figure 3.6 : Frame PSNR comparison between the proposed JSDEC framework and the systems using JSCC with constant packet loss rates. ($\epsilon_1 = \epsilon_2 = 2\%, W_{T2} = 0.6MHz$)

improvement of decoded video quality of JSDEC over JSCC is because the sender can calculate an optimal source rate from a wider range, which is not limited by any of the two channels. That is to say, our proposed system is more aggressive in allocating part of the available channel rate for error control since it can lead to improvement in the overall video quality. Therefore, the resources used by source coding and error control are globally optimized in a joint and distributed way to achieve the best user-perceived quality. This conclusion is also supported by the simulation results we provide in the following paragraphs.

Similarly, in Figure 3.6, W_{T2} is set to a fixed value at $0.6MHz$, while W_{T1} varies between $0.2MHz$ to $0.4MHz$ at a step size of $50KHz$. In this figure, the performance gain of JSDEC over JSCC is much lower compared with the results in Figure 3.5. This is because in Figure 3.6, the available channel bandwidth of the second channel is always much higher

than that of the first one. Thus, there is always enough rate on the second channel to be allocated for error control, which makes the channel distortion modicum.

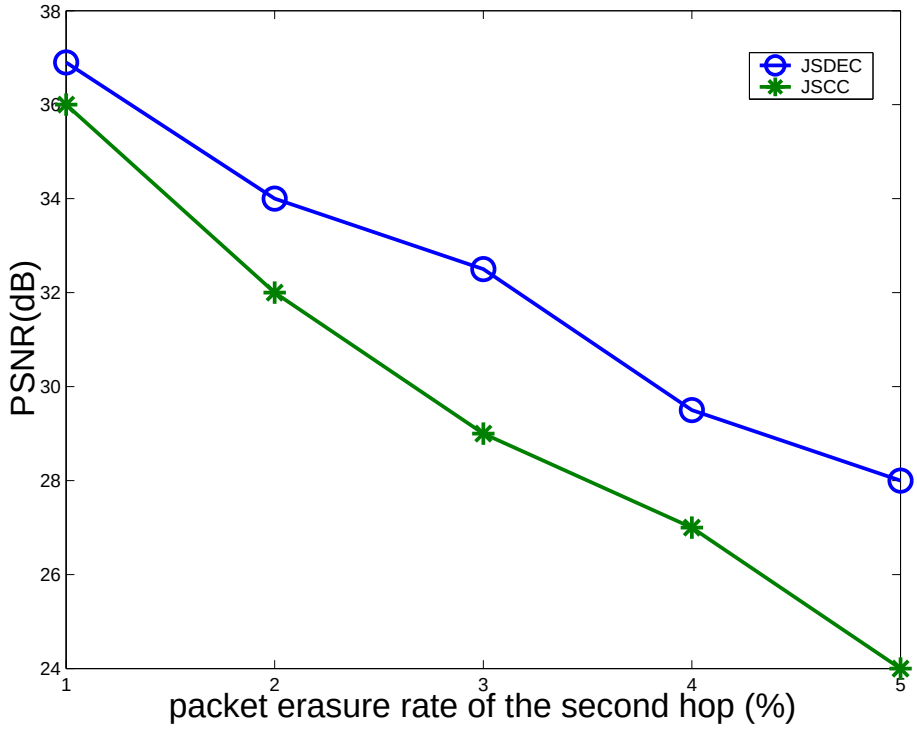


Figure 3.7 : Frame PSNR comparison between the proposed JSDEC framework and the systems using JSCC with constant bandwidth. ($W_{T1} = W_{T2} = 0.3MHz, \epsilon_1 = 2\%$)

To test system performance under different parameters, we then present in Figure 3.7 and Figure 3.8 experimental results for a configuration where the packet loss rates vary within the range of $1 - 5\%$, while the channel bandwidths are kept constant at $0.3MHz$. Similarly, we also assume that the two links in tandem are attached through a single proxy. In both figures, we set the average PER of the first hop ϵ_1 to 2% and 5%, respectively, while ϵ_2 changes from 1% to 5%.

When comparing the results in these two figures, we can conclude that if either ϵ_1 or ϵ_2 is low, the performance does not differ too much. This is easy to explain. Low PER corresponds to few packet losses and thus minimal channel distortion. However, as either

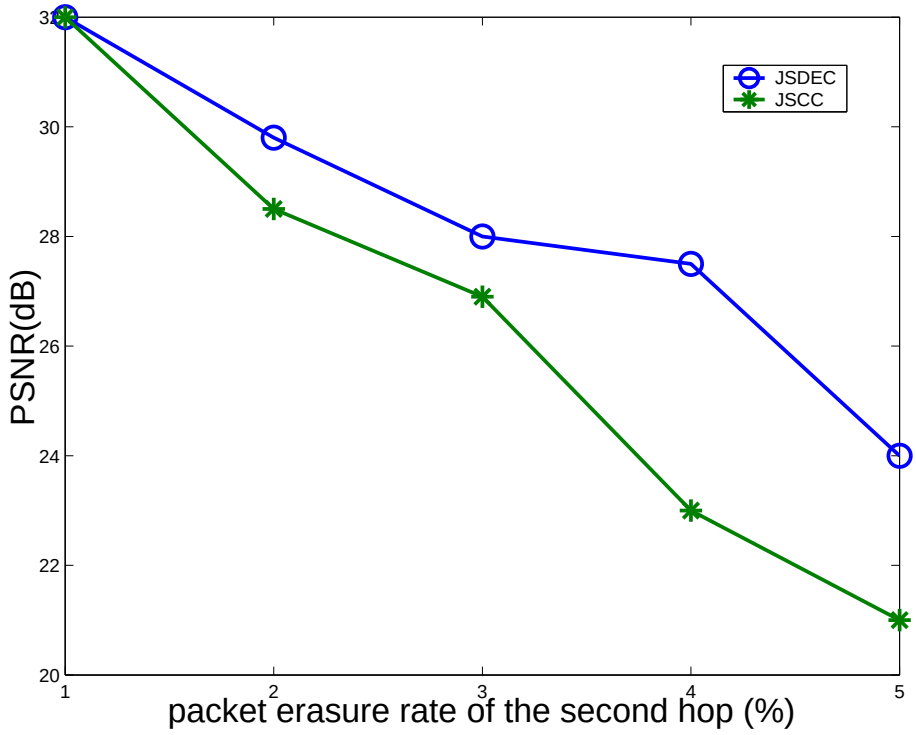


Figure 3.8 : Frame PSNR comparison between the proposed JSDEC framework and the systems using JSCC with constant bandwidth. ($W_{T1} = W_{T2} = 0.3MHz, \epsilon_1 = 5\%$)

of the packet erasure rates increases, channel distortion also increases. More interestingly, when the discrepancies in the packet erasure rates are considerable, and especially for high PER, the performance gain of the proposed system is more significant when compared with JSCC. This result essentially means that it is more crucial to use JSDEC when the two channels experience significant variations in link quality, which also makes our proposed system particularly useful for wireless multi-hop scenarios.

Now we demonstrate another set of simulation results in Figure 3.9. In this case, W_{T1} and W_{T2} are both set to $0.3MHz$. We measure the bit rate allocated for error control for JSDEC and JSCC when averagely ϵ_1=2% and ϵ_1=5%, respectively, while ϵ_2 changes from 1% to 5% at a step size of 1% for both cases. It is obvious from the figure that with the increase of packet erasure rate, the resources allocated for error control always increase.

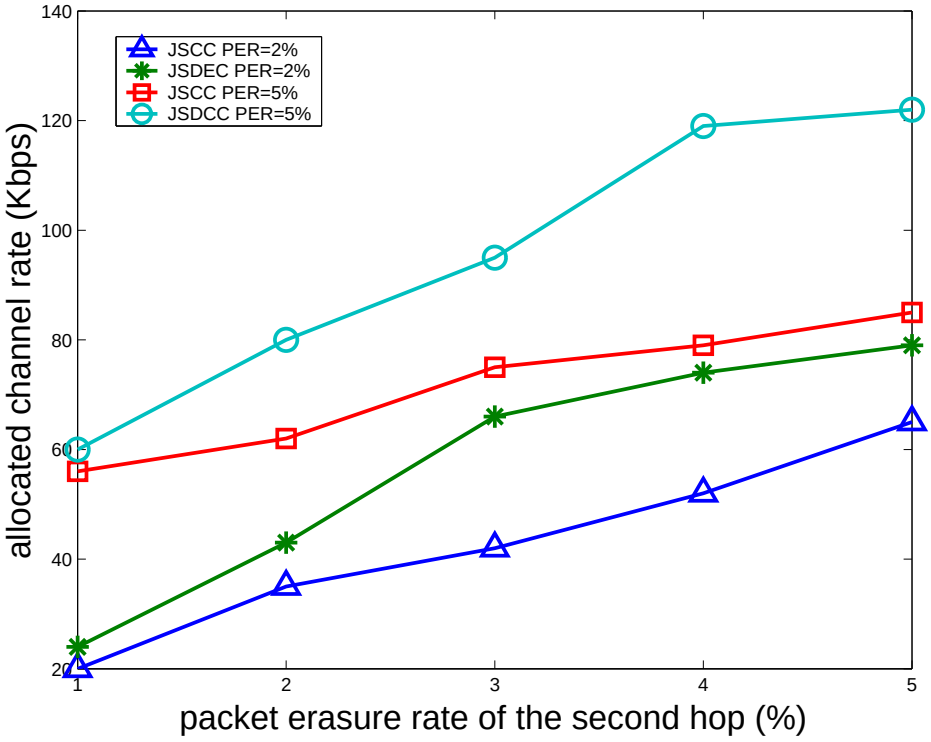


Figure 3.9 : Frame allocated channel coding rate comparison between the proposed JSDEC framework and the systems using JSCC with constant channel rates. ($W_{T1} = W_{T2} = 0.3MHz$)

More importantly, even under the same PER value, our proposed JSDEC always allocates more resources for error control. This explains the performance advantages observed in Figures 3.5 through 3.8. Also, this more aggressive resource allocation strategy is used in the case where the PER differs significantly across the tandem links. This is necessary given the need for more aggressive error control when increasing packet losses lead to increasing channel distortion. However, the JSCC scheme is impossible to account for these situations on the second channel, and as a result it cannot select a source rate that enables optimal error control on the second hop.

Further, we consider a more complicated scenario where multiple proxies are connected in tandem. We simulate as many as six proxies connected in this chain. The available

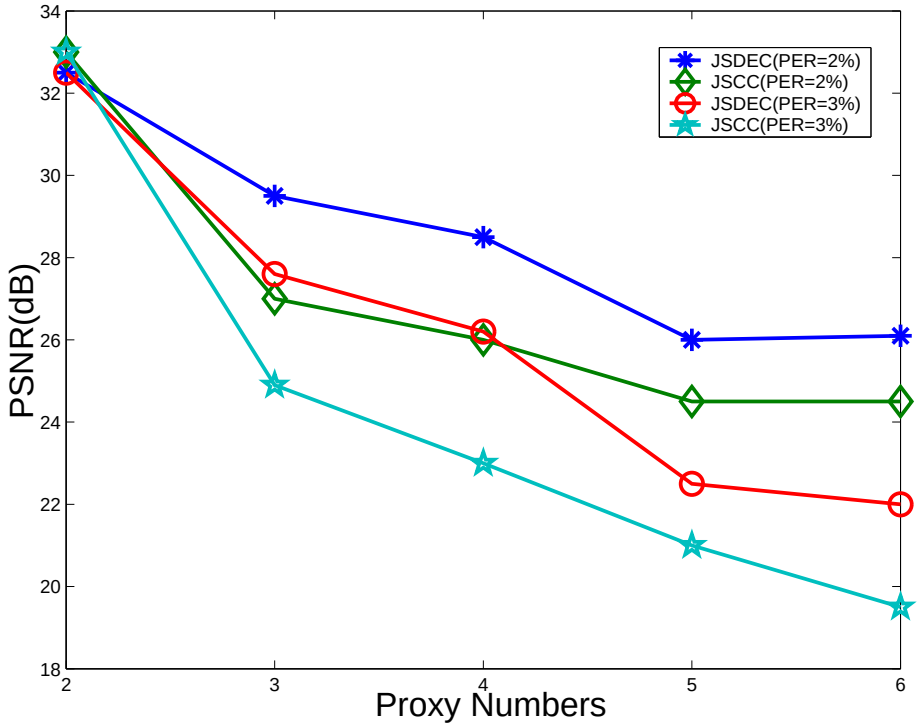


Figure 3.10 : Frame PSNR comparison between the proposed JSDEC framework and the systems using JSCC with multiple proxies connected in tandem.

bandwidths in the network topology are set in this way – each link, which is connected in tandem, is set to $0.3MHz$ or $0.4MHz$ in a continuous and alternate way. That is, link 1, link 2, link 3, link 4, link 5 and link 6 are connected by proxy 1, proxy 2, proxy 3, proxy4, and proxy 5 to consist of a serial network topology. The available bandwidth of link 1, link 2, link 3, link 4, link 5, and link 6 are set to $0.3MHz$, $0.4MHz$, $0.3MHz$, $0.4MHz$, $0.3MHz$, $0.4MHz$, respectively. The average PER is set to 2% in the first round of simulation and then 3% in the second round. The PSNR performance enhancement is demonstrated in Figure 3.10. Evidently, the increase of proxies increases the end-to-end path and packet loss rate as well, which in turn decreases the measured PSNR. But this is due to the fact that we simulate an increasing number of erasure channels. Nevertheless, our proposed JSDEC algorithm demonstrates the ability to minimize the video distortion by

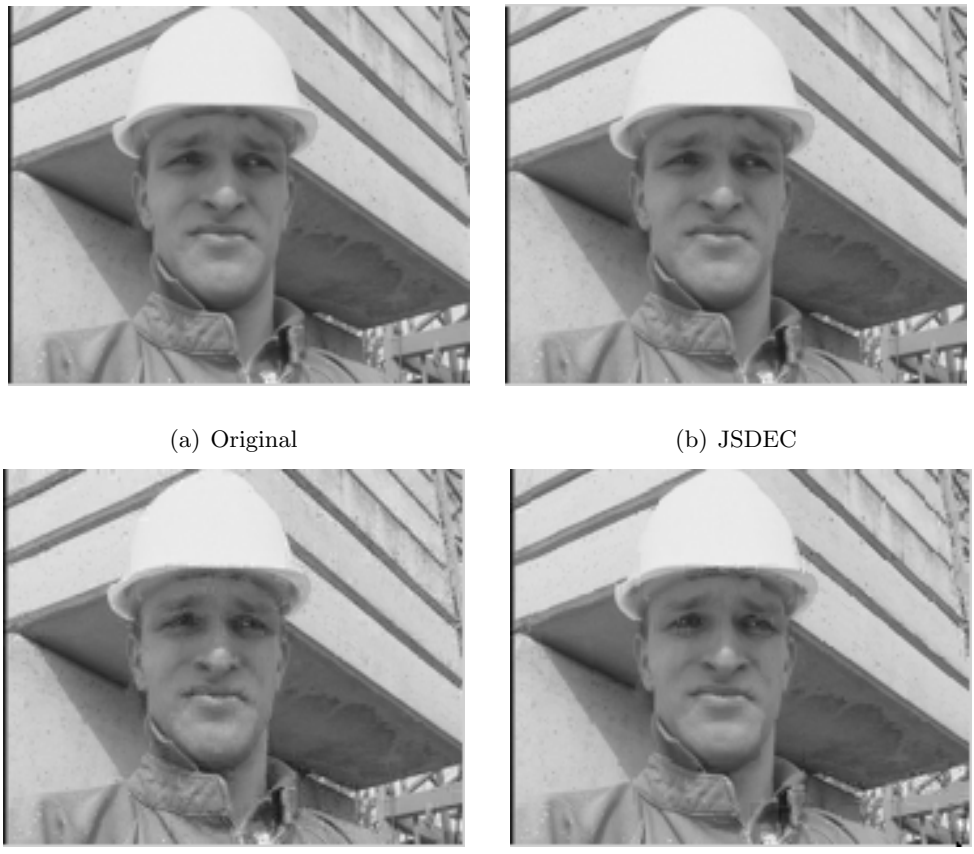
jointly optimizing the resources allocated for source coding and error control. Moreover, as demonstrated in this figure that even when the PER is low, the proposed algorithm is still very effective in improving user-perceived video quality. Another important observation is that when the JSDEC algorithm is used, the performance can be even better than a system using JSCC with a slightly lower PER. This result is significant evidence of the effectiveness of the proposed system. As an explanation, again from Figure 3.10, when the number of proxies is equal and less than 4, the simulation using JSDEC with PER=3% gains a better performance than a system using JSCC with PER=2%. This is only possible because JSDEC is more aggressive to assign resources to channel coding.

All these experimental results are geared towards demonstrating the possible performance enhancement of our proposed JSDEC algorithm compared with the existing end-to-end JSCC. Overall, our system is especially suited to an environment of highly asymmetrical link qualities with available channel rate differences falling into a relatively small range. This is almost always true in a wireless multi-hop environment, where the wireless bandwidth is usually the same for any given wireless application, while the channel quality such as packet erasure rates are highly varying due to the unstable radio communication.

Finally, to illustrate the visual quality superiority produced by our system, we plot the 27th frame of the reconstructed sequence of "Foreman" used in our system as demonstrated in Figure 3.11. The figure shows the user-perceived video quality differences achieved under the different circumstances such as JSDEC, JSCC or no joint coding at all. The simulation environment is the same with that of the results presented in Figure 3.5. Obviously, the proposed JSDEC architecture provides a higher user-perceived video quality at the receiver end.

3.7 Related Work

Our work brings together techniques of error resilient source coding and error control but in a distributed framework. Therefore, we will review mechanisms that deal with enhancing



(a) Original (b) JSDEC

(c) JSCC (d) Standard H.264

Figure 3.11 : Frame quality comparison between the proposed JSDEC framework and the systems using JSCC and standard H.264 codec.

robustness of video streaming with joint source/channel coding and network-supported error control.

One of the most important characteristic of the proposed scheme is that the expected distortion is accurately calculated by ROPE. This is different from the use of a distortion metric in the literature. For example, in [41], the fixed distortion deduction obtained by successfully decoding a video unit is adopted as the objective function. Also, in [42], the resulting distortion from losing one or multiple video descriptions as the optimization metric is employed but the amount of the distortion is preset and fixed, which means that the codec cannot adapt to the instantaneous network conditions. This is extremely important for the

class of applications in which we are interested.

Regarding the error control functionality, the most modern techniques employ both application layer ARQ and FEC for achieving optimal error control [43, 44]. Joint source and channel coding at the application layer has also been studied in an integrated framework when a real-time video encoder is employed [30, 45]. In these works that make use of hybrid ARQ and FEC error control mechanisms, it has been shown that retransmissions are suitable for networks with low delay, low packet loss probability, and low transmission rate, while FEC is more suitable otherwise. Therefore, the aforementioned methods attempt to identify the best combination of the two. However, all these mechanisms consider end-to-end retransmissions, which practically introduces unacceptable delay for real-time interactive video communications.

Several works moved to the next level, and proposed the use of a proxy node for performing error control across asymmetric networks. For example, in [27] the authors propose a method for pre-encoded video streaming through a proxy in a rate-distortion optimized way, with the assumption that all the interdependencies between the video data units are available. Clearly, this approach is computationally expensive when it is employed at several nodes in the end-to-end path. Another interesting work that considers multiple proxies can be found in [46]. In that chapter, the authors propose a scheme that employs forward error correction (FEC) at intermediate nodes in the network. The disadvantages of that approach is the constant overhead required by FEC, and the uncoordinated, and therefore sub-optimal, selection of the FEC parameters. In [47] the authors provide an analysis that supports their claim that it is viable to use intermediate proxies for ARQ in real-time Internet video streaming applications. We should also mention another recent trend in the area of channel coding for video streaming applications, which is the use of rateless codes where the streaming server already encodes proactively the bitstream [34, 48]. However, even in these systems the implications of real-time encoding is not considered.

3.8 Summary

In this chapter, we presented a framework for error resilient source coding and distributed application-layer error control (JSDEC) suitable for video streaming in wireless multi-hop networks. Our systems employed distributed error control by a very lightweight application-layer ARQ algorithm that is introduced at intermediate nodes. We demonstrated that in order to achieve globally optimal joint source and channel coding decisions for packet video transmission over wireless multi-hop networks, the end-to-end video distortion estimate must consider the contribution of each communications link. We derived such a model for a path that consists of multiple packet erasure channels connected in tandem. Subsequently, we formulated the optimization problem that considered source coding, which is employed by selecting the appropriate encoding mode for each individual macro-block, and error control, which is applied through retransmissions at each proxy. Our solution algorithm is based on dynamic programming and introduces moderate complexity at the streaming server.

Our experiments with NS-2 and H.264/AVC validated the efficacy of the proposed scheme in terms of video quality improvement. We evaluated our framework for time-varying channels by using packet-level feedback from the nodes. Our algorithm was shown to be highly successful, especially in scenarios where there are highly asymmetric packet loss rates.

Chapter 4

Content-aware Multimedia Communications

In this chapter, we discuss our research on video content-aware analysis and how to use it to optimize wireless video delivery. The application studied here is a wireless e-healthcare platform.

4.1 Introduction

In this chapter, we focus on gait analysis and its wireless delivery, which can be used in surveillance or biomedical assistance. Gait recognition is the process of identifying an individual by the manner in which they walk. This is a markerless unobtrusive biometric, which offers the possibility to identify people at a distance. Using gait as a biometric is a relatively new area of study within the realms of computer vision. It has been receiving growing interest within the computer vision community and a number of gait metrics have been developed.

The increasing prevalence of inexpensive hardware such as video phones and CMOS cameras has fostered the development of wireless multimedia technologies, such as Wireless Multimedia Sensor Networks (WMSN), the interconnected devices through wireless networks that is able to ubiquitously retrieve multimedia content such as still images, audio and video streams, as well as scalar sensor data from the environment. WMSNs not only enhance existing sensor network applications such as tracking, surveillance, home automation, and environmental monitoring, but also facilitate new medical applications such as telemedicine, an advanced health care delivery. In truth, telemedicine is a rapidly developing application of clinical medicine, where medical information is transferred via telephone, the Internet or wireless networks for the purpose of consulting, and sometimes remote med-

ical procedures or examinations. Telemedicine sensor networks can be integrated with 3G broadband multimedia networks to provide ubiquitous health care services. Furthermore, remote monitoring is a new technology emerging to improve disease treatment and lower medical costs. The essence of remote monitoring is to enable assessment of an individual's medical status in real time regardless of his or her location, and to allow a doctor or a computer to view the information anywhere to aid diagnosis, observe how a treatment is working, or determine if a condition has become acute. Such a capability can combine more accurate, more up-to-date data gathering with better data analysis, while in the meantime allowing the individual to remain in comfortable surroundings. By eliminating many trips to a physician's office or care facility, medical cost is significantly reduced and convenience and care quality are improved.

Figure 4.1 : The current marker-based system for human motion capturing.

Therefore, an e-healthcare platform based on wireless multimedia technologies for rapid or real-time markerless human motion tracking and gait analysis will significantly improve the current clinical medicine practices, especially for resource-limited environments such as clinics in rural communities. It can also be widely employed in medical examinations in

sports fields, disaster zones and battle fields. For example, track and field athletics on the competition site in some case may need preliminary but speedy examinations to decide if they are in a good condition to participate in a subsequent competition, or sometimes they need to know what precautious measures they should take to finish the remaining games given their up-to-date body conditions.

However, in existing systems, the huge amount of collected human motion data are usually transmitted to the medical center for analysis over wired connections or by off-line transmissions. When applied to the wireless environment for rapid or real-time medical examinations, the current systems are no longer applicable due to the bandwidth limitation and varying wireless channel qualities, leading to excessive delay for medical prognosis and diagnosis. Furthermore, in a traditional marker-based motion capturing environment as shown in Figure 4.1, markers are attached to the subject's key points, namely the key joints of the body framework, to represent the constructed 3D points using geometrical methods from multi-camera feeds. Despite the willingness or likelihood of the monitored person to put up such markers in real situations, the intrusive marker-based methods are known for the implementation difficulties. Footage taking in a specifically designed environment requires elaborate computation as well as expensive equipment. Background interference and occlusions usually prevent certain markers from being detected accurately by any camera and pose a serious challenge for human motion tracking and gait analysis.

Upon these observations, in this chapter we present an e-healthcare system for real-time markerless remote human motion tracking based on content-aware wireless streaming to solve the aforementioned issues, as illustrated in Figure 4.2. Simply put, it is a cyber-enabled, low-cost, highly accurate, portable e-healthcare platform for real-time retrieval of gait data from an end point located in an wireless network and to complete speedy prognosis and diagnosis of pathological human gait rhythms. The system first accurately extracts the human motion regions, by jointly utilizing the properties of the inter-frame relations, the spectral and spatial inter-pixel dependent contexts of the video frames. Thus, it avoids the

reliance on the expensive equipment of the traditional pre-designed intrusive marker-based environment. Then, based on the content-aware analysis results, the collected video gait is transmitted to the medical center for rapid or real-time medical prognosis and diagnosis through a wireless network, where a proposed quality-driven distortion-delay framework of optimization is able to achieve highly improved video quality. Specifically, video coding parameters at the application layer and the Adaptive Modulation and Coding (AMC) scheme at the physical layer are jointly optimized through a minimum-distortion problem to satisfy a given playback delay deadline. In other words, the optimal combination of encoder parameters and AMC scheme are chosen to achieve the best fidelity for the collected human motion data. Also, the extracted human motion regions are coded, transmitted in video encoding with a heavier protection compared with the insignificant background areas under the given QoS requests, to improve the overall video quality. The experimental results by using H.264/AVC prove the validity and effectiveness of the overall performance enhancement on the received human motion video data through our proposed system.

4.2 Human Motion Tracking

Given a gait video, the human motion regions contain valuable information for gait analysis and medical examinations. In contrast, the background areas usually provide much less useful knowledge for medical prognosis. However, according to our study, for a video frame of any collected human gait data, the background areas usually comprise more than 50% of the whole video frame. Transmitting the background areas and the human motion regions without differentiation is not only unnecessary but also wasteful of precious wireless network resources. Therefore, the identification of human motion regions through our proposed content-aware analysis techniques is a meaningful task. The result provides a foundation for wireless streaming of real-time remote human motion tracking by prioritizing the transmission of the human motion regions against the unimportant background areas under the resource-limited environment. Furthermore, the proposed markerless human mo-

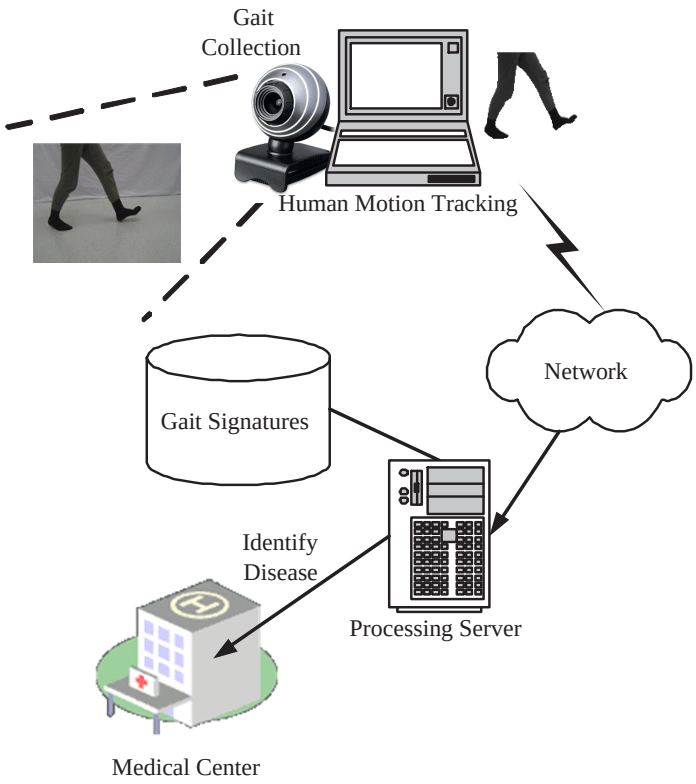


Figure 4.2 : The scenario of proposed quality-driven content-aware wireless streaming for real-time markerless remote human motion tracking.

tion tracking system makes the expensive infrastructure and specially designed facilities no long necessary.

Figure 4.3 shows the procedure for markerless human motion tracking and their corresponding results achieved by using the proposed system, where background subtraction and contextual classification play the major role. Through these two steps, the temporal inter-frame correlation of the video clip, the spectral and spatial inter-pixel dependent contexts of the vide frames are cooperatively utilized, making it possible to accurately track the human motion regions in a markerless environment.

1) Background Subtraction.

The rationale of background subtraction is to detect the moving objects from the difference between the current frame and a reference frame, often called the "background image."

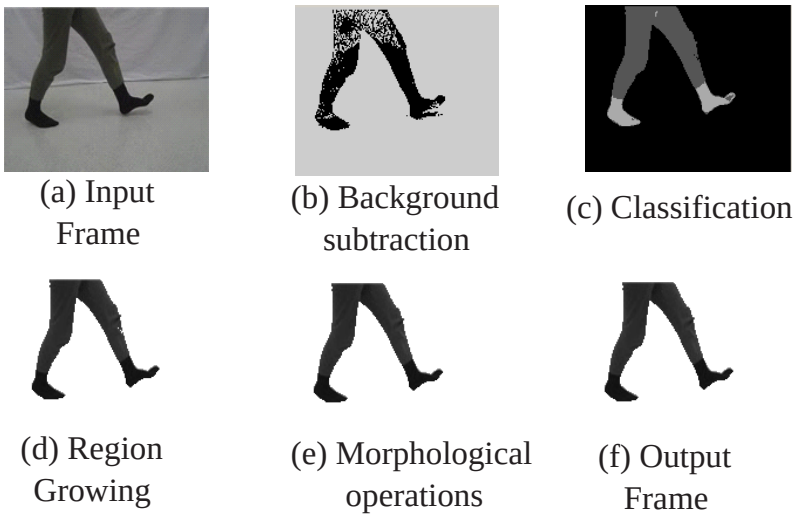


Figure 4.3 : Illustration of different stages of the proposed markerless system of human motion tracking.

Therefore, the background image is ideally a representation of the scene with no moving objects and is kept regularly updated so as to adapt to the varying luminance conditions and geometry settings [49]. Given a video frame, the objective of background subtraction is to detect all the foreground objects, that is, the human motion regions in this chapter. The naive description of the approach is to depict the human motion regions as the difference between the current frame Fr_i and the background image Bg_i:

$$|Fr_i - Bg_i| > T_h \tag{4.1}$$

where T_h denotes a threshold.

However, the background image is not fixed. Therefore, before this approach can actually work, certain factors need to be adapted to, which includes illumination changes, motion changes as well as sometimes the changes in background geometry. Over time, different background objects are likely to appear at the same pixel location. Sometimes the changes in the background object are not permanent and appear at a rate faster than that of the background update. To model this scenario, a multi-valued background model can be adopted to cope with multiple background objects. Therefore, algorithms such as the

proposed Gaussian Mixture Model (GMM) can define an image model more properly as it provides a description of both foreground and background values [50]. The result of this step is illustrated in (b) of Figure 4.3. In the following, we explain in detail the GMM model that is used to perform background substraction.

A mixture of K Gaussian distributions is adopted to model the pixel intensity ($K = 3$ in this work), where each Gaussian is weighted according to the frequency with the corresponding observed background. The probability that a certain pixel has intensity X_t at time t is estimated as:

$$P(X_t) = \sum_{i=1}^{K} w_{i,t} \frac{1}{\sqrt{2\pi\sigma_i}} e^{\frac{1}{2}(X_t-\mu_{i,t})^T \sum^{-1}(X_t-\mu_{i,t})} \tag{4.2}$$

where $w_{i,t}$ is the normalized weight, μ_i and σ_i are the mean and standard deviation of the i^{th} distribution. As the parameters of the mixture model of each pixel change, the most likely Gaussian of the mixture produced by the background processes is determined as follows. The K distributions are ordered based on the value of w/σ, where the most likely background distributions remain on top and the less probable transient background distributions gravitate toward the bottom. The most likely background distribution model B within b distributions is found by

$$B = arg\ min_b \left(\sum_{j=1}^{b} w_j > T \right) \tag{4.3}$$

where the threshold T is the fraction of the total weight given to the background. The new pixel is checked against the existing K Gaussian distributions until a match is found. A match is defined as the distance between the mean of the distribution and the new pixel value is within 2.5 standard deviations the distributions. If none of the K distributions match the current pixel value, the least probable distribution, which has the smallest value of w/σ, is replaced by a new distribution with the current new pixel value as the mean, an initially high variance and low prior weight. In general, a new pixel value can always be represented by one of the major components of the mixture model of K Gaussian distributions. If this

matched distribution is one of the B background distributions, the new pixel is marked as background, otherwise foreground. To keep the model adaptive, the model parameters are continuously updated by using the pixel values. For the matched Gaussian distributions, all the parameters at time t are updated with this new pixel value X_t. In addition, the prior weight is updated as

$$w_t = (1 - \alpha)w_{t-1} + \alpha \tag{4.4}$$

The mean and variance are updated as

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \tag{4.5}$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^2 \tag{4.6}$$

where α is the learning rate controlling adoption speed, $1/\alpha$ defines the time constant which determines change, and ρ is the probability associated with the current pixel, scaled by the learning rate α. So ρ can be represented by

$$\rho = \alpha \frac{1}{\sqrt{2\pi}\sigma_t} e^{-\frac{(X_t - \mu_t)^2}{\sigma_t^2}} \tag{4.7}$$

For unmatched distributions, the mean μ_t and variance σ_t remain unchanged, while the prior weight is updated by

$$w_t = (1 - \alpha)w_{t-1} \tag{4.8}$$

One advantage of this updating method is that, when it allows an object to become part of the background, it doesn't destroy the original background model. In other words, the original background distribution remains in the mixture until it becomes the least probable distribution and a new color is observed. So if this static object happens to move again, the previous background distribution will be rapidly reincorporated into the model.

2) Contextual Classification.

The objective of classification is to classify an video frame by the object categories that it contains. Supervised classification is a type of automatic multi-spectral image interpretation in which the user supervises feature classification by setting up prototypes (collections of sample points) for each feature, class to be mapped. A supervised contextual classification that utilizes both spectral and spatial contextual information can better discriminate between the pixels with similar spectral attributes but located in different regions. First, in many images, especially remotely-sensed images, object sizes are much greater than the pixel element size. Therefore, the neighboring pixels are more likely to belong to the same class, forming a homogeneous region. Furthermore, some classes have a higher possibility of being placed adjacently than others, so the information available from the relative assignments of the classes of neighboring pixels is also very important. By using both spectral and spatial contextual information, the speckle error can be effectively reduced and the classification performance can be significantly improved. Nonetheless, this type of classification also suffers from the problem of the small training sample size, where the class conditional probability has to be estimated in the analysis of hyper-spectral data.

Therefore, algorithms such as the adaptive Bayesian contextual classification that utilizes both spectral and spatial inter-pixel dependent contexts to estimate the statistics and classification can be adopted for accurate classification [51]. This model is essentially the combination of a Bayesian contextual classification and an adaptive classification procedure. In this classification model, only inter-pixel class dependency context is considered, while the joint prior probabilities of the classes of each pixel and its spatial neighbors are modeled by using the Markov Random Fields (MRF) [52]. Furthermore, as an adaptive classification procedure, the estimation of statistics and classification is performed recursively. Consequently, this contextual classification achieves higher accuracy and mitigates the small training sample problem in the analysis of hyper-spectral data as shown in (c) of Figure 4.3. In the following, we describe in detail the MRF model that is used to perform object segmentation.

In the adaptive Bayesian Contextual classification model, the label of each semi-labeled sample is updated after each classification, including Maximum Likelihood (ML), Maximum A Posterior Probability (MAP), and post-processing classification at each cycle, and the weight of each semi-labeled sample is updated after each cycle. Correspondingly, the class conditional statistics are updated at each cycle as well. One complete cycle of the Adaptive Bayesian Contextual Classifier is illustrated in Figure 4.4.

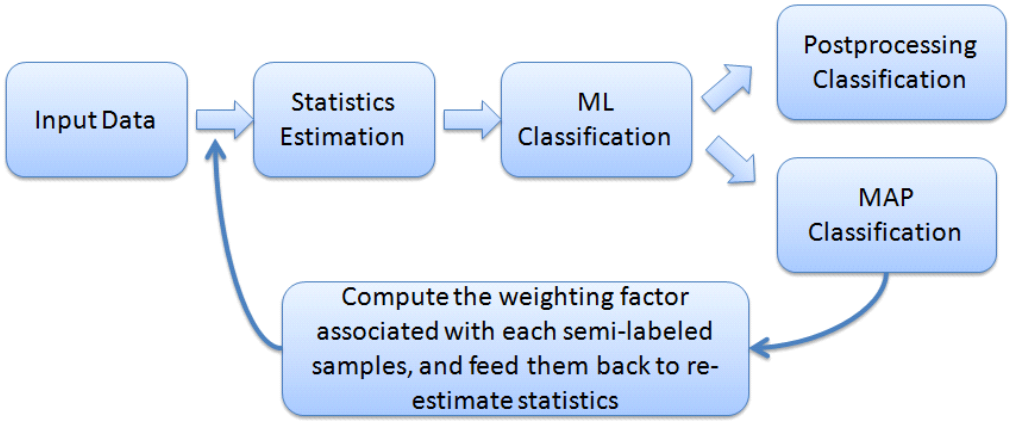


Figure 4.4 : One complete cycle of the Adaptive Bayesian Contextual Classifier.

Assume the initial class conditional statistics and classification has been obtained by using the training samples, and all L classes can be represented by Gaussian distributions. Denote $y = (y_1, ..., y_{im_i})$ as the training samples from the i^{th} class, whose pdf is $f_i(y|\phi_i)$, and $x = (x_{i1}, ..., x_{in_i})$ are the semi-labeled samples that have been classified to the i^{th} class. Hence m_i is the number of training samples for i^{th} class, and n_i is the number of semi-labeled samples classified to the i^{th} class, and ϕ_i represents the set of parameters for the i^{th} class.

The following is the procedure for this algorithm: [51]

Cycle 1 (Initial Cycle)

1) Use only training samples to estimate statistics, and then perform classification using a

ML classifier.

2) Perform classification using a MAP classifier based on the classification map from the ML:

$$X(s) \in u \leftrightarrow arg\ min_{1 \leq u \leq L} \left[ln| \sum_u | + (X(s) - \mu_u)^T \sum_u^{-1} (X(s) - \mu_u) + 2m\beta \right] \tag{4.9}$$

where β is empirically determined.

3) Perform classification using a post-processing classifier based on the classification map from the ML

$$X(s) \in u \leftrightarrow u(s) = arg\ max_{u(s)} \left[p\{u(s)|\partial u(s)\} \right] \tag{4.10}$$

The purpose of using the post-processing classifier is to compare the results from the MAP classifier.

Cycle 2:

1) Compute weighting factors using contextual information together with the likelihood based on the classification results from the MAP classifier in step (2) from the previous cycle:

$$w_{uj}^c = \frac{p(x_{uj}|\phi_u^c)}{p}(u(s)|u(\partial s)) \sum_{k=1}^{L} p(x_{uj}|\phi_k^c) p(k(s)|k(\partial s)) \tag{4.11}$$

Note that unit weight is assigned to each training sample.

2) Obtain the class conditional statistics by maximizing the mixed log likelihood of training samples and of semi-labeled samples, which are obtained from the MAP classifier in step (2) from the previous cycle.

$$\mu_i^+ = \frac{\sum_{j=1}^{m_i} y_{ij} + \sum_{j=1}^{n_i} w_{ij}^c x_{ij}}{m_i + \sum_{j=1}^{n_i} w_{ij}^c} \tag{4.12}$$

$$\sum_i^+ = \frac{\sum_{j=1}^{m_i} (y_{ij} - \mu_i^+)(y_{ij} - \mu_i^+)^T + \sum_{j=1}^{n_i} w_{ij}^c (x_{ij} - \mu_i^+)(x_{ij} - \mu_i^+)^T}{m_i + \sum_{j=1}^{n_i} w_{ij}^c} \tag{4.13}$$

Note that the estimated statistics are affected by training samples and semi-labeled samples.

3) Perform classification based on the maximum likelihood (ML) classification rule:

$$X(s) \in u \leftrightarrow u(s) = arg\ min_{1 \le u \le L} \left[ln| \sum_{u}^{+} | + (X(s) - \mu_u^+)^T (\sum_{u}^{+})^{-1} (X(s) - \mu_u^+) \right] \quad (4.14)$$

4) Perform classification using the MAP classifier based on the classification map from the MLC

$$X(s) \in u \leftrightarrow u(s) = arg\ min_{1 \le u \le L} \left[ln| \sum_{u}^{+} | + (X(s) - \mu_u^+)^T (\sum_{u}^{+})^{-1} (X(s) - \mu_u^+) + 2m\beta \right] \quad (4.15)$$

5)Perform classification using the post-processing classifier based on the classification map from the MLC (step 3).

$$X(s) \in u \leftrightarrow u(s) = arg\ max_{u(s)} \left[p\{u(s)|u(\partial s)\} \right] \quad (4.16)$$

The steps of cycle 2 are repeated until convergence is reached, where the classification results have small changes.

3) Region Growing.

At this stage shown in (d) of Figure 4.3, density check is adopted to combine the results from the previous two steps to form the continuous human motion regions [53]. As long as a homogeneous region achieved from stage 2 contains more than a threshold percentage of human motion pixels obtained from stage 1, this region is regarded as the human motion region. Otherwise, it falls into the background areas. The algorithm is shown in Algorithm 1.

where T is the density threshold for foreground and background. In this research T is set to 0.05.

4) Morphological Operations and Geometric Corrections.

Results from the previous stage contain undesired noises and holes. As shown in (e) of Figure 4.3, morphological operations use dilation and erosion to populate the holes in

Input: A still image after object segmentation.

Output: A more homogeneous image after region growing.

Denote E the foreground pixels after background subtraction.

Denote $I_1, I_2, I_3, \cdots I_n$ the different regions from image classification.

repeat

C_1 = all the elements in I_i;

C_2 = all the elements in I_i and all the elements in E;

if $C_2/C_1 > T$ **then**

Mark all the elements in I_i as foreground region.

end

until $\forall e \in E,\ e \notin I_i$;

Algorithm 1: Region growing algorithm used for content-aware analysis.

the human motion regions and remove the small objects in the background areas. Then, geometric correction can be performed horizontally and vertically to further remove noises for the accurate achievement of human gait. [53]. Finally, we achieve the desired results of the extracted human motion regions as displayed in (f) of Figure 4.3.

4.3 Real-time Streaming of Human Motion Video over Wireless Environment

4.3.1 The Proposed System Model

Based on the proposed markerless human motion tracking results, we also develop a unified quality-driven optimization system of wireless streaming for rapid or real-time delay-bounded human motion tracking. Figure 4.5 illustrates the proposed system model, which consists of an optimization controller, the maker-less human motion tracking module, a video encoding module, as well as the modulation and coding module. To increase the overall video quality, we first adopt the proposed methods mentioned in Section 4.2 to i-

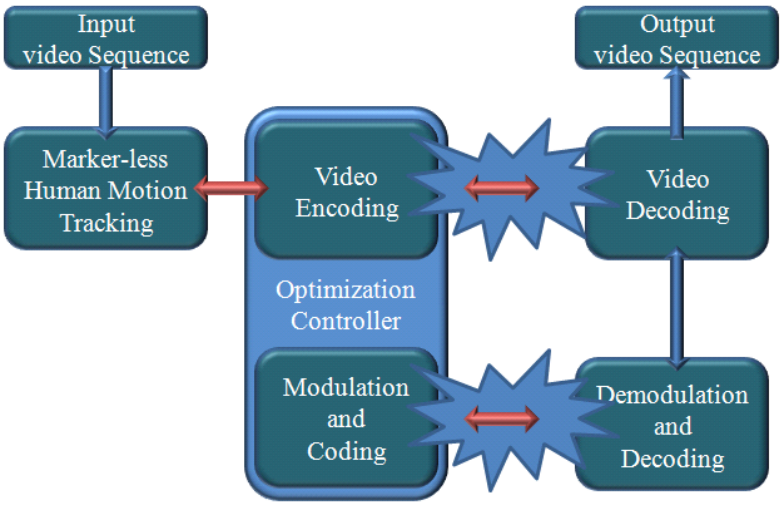


Figure 4.5 : The system model of the proposed quality-driven content-aware wireless streaming system for real-time human motion tracking.

dentify the human motion regions in a maker-less environment. With the consideration of the different contributions of the human motion regions and the background areas for gait analysis, at the video encoder of the application layer, the human motion regions can be coded by a finer quality, and at the physical layer the packets of human motion regions can use larger constellation sizes and higher channel coding rate to guarantee the required packet error rate. By redistributing the limited resources needed for encoding and transmission according to the video content, the overall quality of real-time human motion video delivery over wireless networks will be significantly improved.

The optimization controller is the core of the proposed system, which is equipped with the key system parameters of video codec in the application layer and the modulation and coding schemes in the physical layer. Therefore, through these parameters, the controller can control the behaviors of the video encoder and the modulation and coding module. More importantly, adjusting with coordination the system parameters of the video encoder residing in the application layer and the modulation and coding schemes residing in the physical layer can greatly enhance the overall network performance. As shown in Figure 4.6, the

system performance in terms of video distortion is jointly decided by the encoder behavior (that is, quantization step size, or QP) and packet loss rate. Furthermore, packet loss rate is determined by bit error rate (BER), which is then collectively affected by the channel quality and the AMC scheme. Therefore, all related system parameters can be holistically optimized toward the best possible video quality under a given delay constraint. For example, when a wireless channel is experiencing bad quality, the time-varying channel information can be used to dynamically adapt the system parameters of AMC scheme to minimize the packet loss rate,thus enhancing the received video quality over wireless networks. Therefore, the proposed cross-layer based joint optimization is able to choose the optimal set of parameter values to achieve the best received video performance, providing a natural solution to improve the overall system performance for wireless streaming of remotely real-time human motion tracking.

4.3.2 The Optimized Content-Aware Real-time Wireless Streaming

At the video encoder, for hybrid motion-compensated video coding and transmission over lossy channels, each video frame is generally represented in block-shaped units of the associated luminance and chrominance samples (16×16 pixel region) called macroblocks (MBs). In the H.264 codec, macroblocks can be both intra-coded or inter-coded from samples of previous frames [2]. Intra coding is performed in the spatial domain, by referring to neighboring samples of previously coded blocks which are to the left and/or above the block to be predicted. Inter coding is performed with temporal prediction from samples of previous frames. It is evident that many coding options exist for a single macroblock, and each of them provides different rate-distortion characteristics. In this work, only predefined macroblock encoding modes are considered, since we want to apply error resilient source coding by selecting the encoding mode of each particular macroblock. This is crucial to allow the encoder to trade off bit rate with error resiliency at the macroblock level. For real-time source coding, the estimated distortion caused by quantization, packet loss,

and error concealment at the encoder can be calculated by using the "Recursive Optimal Per-pixel Estimate" (ROPE) method [8], which provides an accurate video-quality based optimization metric to the cross-layer optimization controller.

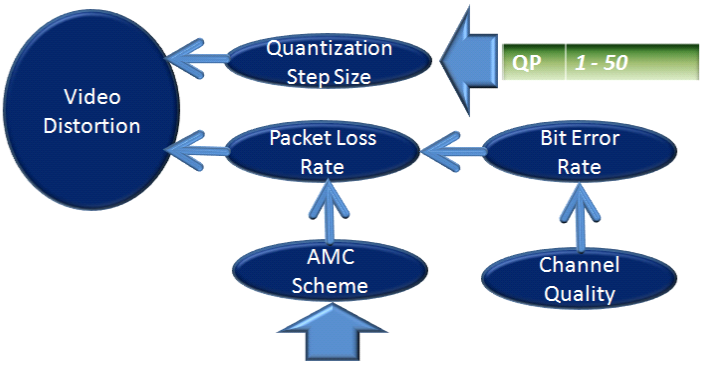
At the physical layer, the bit error rate p_m^e is decided by the dynamically-chosen mode of AMC, which has been advocated to enhance the throughput of future wireless communication systems at the physical layer [9]. With AMC, the combination of different constellations of modulation and different rates of error-control codes are chosen based on the time-varying channel quality. For example, in good channel conditions, an AMC scheme with larger constellation sizes and higher channel coding rate can guarantee the required packet error rate, which means that AMC can effectively decrease the transmission delay, while satisfying the constraint of packet loss rate. Each AMC mode consists of a pair of modulation scheme a and FEC code c as in 3GPP, HIPERLAN/2, IEEE 802.11a, and IEEE 802.16 standards. Furthermore, we adopt the following approximated bit error rate expression:

$$p_m^e(\gamma) = \frac{a_m}{e^{\gamma \times b_m}} \tag{4.17}$$

where m is the AMC mode index and γ is the received SNR. Coefficients a_m and b_m are obtained by fitting (5.4) to the exact BER as shown in Figure 4.6.

Therefore, we can sum that the expected mean-squared error (MSE) between the received pixels and original pixels of the video frames as the distortion metric [8]. Thus, the expected distortion accurately calculated by ROPE under instantaneous network conditions, which is represented by packet loss rate ρ, becomes the objective function in our proposed optimization framework. As shown in (5.4), packet loss rate ρ can be further calculated from bit error rate p_m^e as long as the packet size is known. Meanwhile, the transmission delay,which is constrained by the given frame delay bound, can be represented by bandwidth and data bit rate. Finally, the problem can be formulated as a minimum-distortion problem constrained by a given frame delay bound.

By eliminating from the potential solution set the parameters that make the trans-



AMC Scheme (m)	m=1	m=2	m=3	m=4	m=5	m=6
Modulation	BPSK	QPSK	QPSK	16-QAM	16-QAM	64-QAM
Coding Rate (c_m)	1/2	1/2	3/4	9/16	3/4	3/4
R_m (bits/sym.)	0.5	1.0	1.5	2.25	3.0	4.5
a_m	1.1369	0.3351	0.2197	0.2081	0.1936	0.1887
b_m	7.5556	3.2543	1.5244	0.625	0.3484	0.0871

Figure 4.6 : The different system parameters and relations among them that are considered in the joint optimization of wireless streaming for real-time human motion tracking.

mission delay exceed the delay constraint, the constrained problem can be relaxed to an unconstrained optimization problem. Furthermore, most decoder concealment strategies introduce dependencies among slices. For example, if the concealment algorithm uses the motion vector of the previous MB to conceal the lost MB, it would cause the calculation of the expected distortion of the current slice to depend on its previous slices. Without losing the generality, we assume that the current slice depends on its previous z slices ($z \leq 0$). Then it is evident that given the current decision vectors, the selection of the next decision vector is independent of the selection of the previous decision vectors, which makes the future step of the optimization process independent of its past steps, forming the foundation of dynamic programming. Therefore, the problem can be converted into and solved as a well-known problem of finding the shortest path in a weighted directed acyclic graph (DAG) [39]. In this way, the optimization problem is efficiently solved [54].

4.4 System Experiments

In the experiments, video coding is performed by using the H.264/AVC JM 12.2 codec, where the gait video is recorded through an ordinary video camera under the indoor environment, as shown in Figure 4.3. The frames of the recorded QCIF sequence are coded at the frame rate (R_{frame}) of 30 frames/second, where each P frame is followed by nine I frames. We set one slice to be one row of macroblocks, assuming the whole slice is lost if one of the packets of that slice is lost. This assumption is reasonable since the intra prediction is usually derived from the decoded samples of the same decoded slice. When a slice is lost during transmission, we use the temporal-replacement error concealment strategy. The motion vector of a missing MB can be estimated as the median of motion vectors of the nearest three MBs in the preceding row. If that row is also lost, the estimated motion vector is set to zero. The pixels in the previous frame, pointed by the estimated motion vector, are used to replace the missing pixels in the current frame. Furthermore, we adopt the Rayleigh channel model to describe SNR γ statistically. For the joint optimization of QP and AMC, we allow QP to range from 1 to 50 and AMC to be chosen from the six available schemes illustrated in Figure 4.6. The expected video quality at the receiver is measured by the average of the peak signal-to-noise ratio (PSNR) of the whole video clip. We compare the PSNRs, under the same network conditions, of the reconstructed video sequences at receiver side achieved by using the proposed system to those achieved by using the existing system of non-content-aware analysis, where the video clip is transmitted with fixed quantization step size and AMC scheme.

The relation between playback deadline $D_{playback}$ and frame rate R_{frame} meets Equation (4.18).

$$D_{playback} = \frac{1}{R_{frame}} \tag{4.18}$$

On this basis, we consider three different playback deadline values, $20ms$, $30ms$ and $40ms$, respectively. The received video quality achieved through our proposed system in the ex-

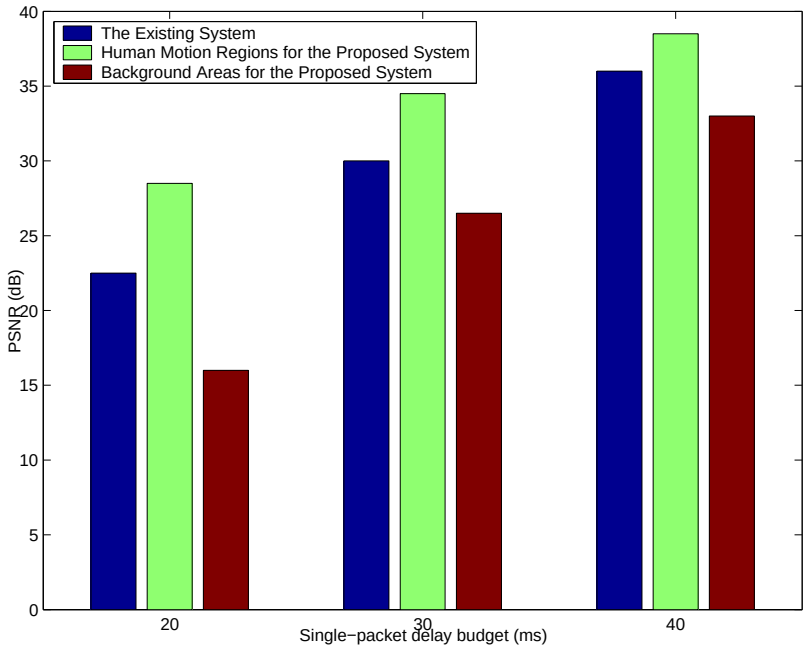


Figure 4.7 : PSNR comparison of different single-packet delay deadlines

periments compared with that through the existing system is demonstrated in Figure 4.7. From the figure, we can observe that in the proposed system, the human motion regions based on content-aware analysis have 3-5dB PSNR improvement compared with the existing system. Meanwhile, the performance gain of the human motion regions over the existing system is even larger in the case of $20ms$ than the other two cases, indicating that the more stringent the single-packet delay deadline is, the more PSNR improvement of the human motion regions the proposed framework can achieve. In other words, the proposed framework is extremely suitable for the delay-stringent wireless networks.

4.5 Summary

In this chapter, an e-healthcare system of quality-driven wireless streaming for remotely real-time human motion tracking has been proposed over wireless networks based on content-aware analysis. The proposed system is able to track human motion regions ac-

curately, avoiding the reliance on the traditionally cumbersome marker-based gait data collection facilities. The temporal relations of intra-frames within the video clip has been fully utilized to subtract the background. The spectral and spatial inter-pixel dependent contexts within a video frame have been further employed for contextual classification. Moreover, a distortion-delay framework has been proposed to optimize the wireless streaming for real-time retrieval of the collected video gait data for gait analysis, based on the extracted human motion regions. All related key system parameters residing in different network layers are jointly optimized in a holistic way to achieve the best received video quality in a wireless environment, including the quantization step size and the AMC scheme. The experimental results have demonstrated the significant performance improvement of the proposed system, which can be employed to provide great convenience and lower cost for real-time prognosis and diagnosis of pathological locomotion bio-rhythm over resource-constrained wireless environment.

Chapter 5

Cross-layer Optimization for Wireless P2P

In this chapter, we study cross-layer optimization and its usage in wireless P2P multimedia networks to dynamically adapt to the wireless channel variations and thus the significant improvement of overall system performance.

5.1 Introduction

Over the last decade, distributed interactive multimedia applications such as peer-to-peer networks have enjoyed tremendous growth. Statistics indicate that at the end of 2004, P2P protocols represented over 60% of the total Internet traffic, dwarfing Web browsing [55]. From the initial file-sharing systems such as Napster, BitTorrent and eMule [56], to the audio-based VoIP systems [57], and recently to the popular video-based IPTV systems such as CoolStreaming, Gridmedia, PPStream and LiveStation [58–61], various P2P systems have been proposed and commercialized. More and more people are now watching online TV/movies through P2P video applications. For instance, in the 2008 Olympic Games, millions of people used PPStream [60] to watch the live broadcasting.

Scheduling is a critical issue in P2P video streaming networks. Before playback, when a video segment at any node is detected missing, the node will either try to fetch the segment from the neighboring nodes or wait for it to arrive from the source code. To ensure good system performance against limited network resources, an effective scheduling algorithm is required to choose the best node from which to fetch the missing segment. As an illustrative example shown in Figure 5.1, every network node (A, B, C, D, E) in the wireless heterogeneous network is assumed to be equipped with a buffer which contains up to nine segments. Before playing back a certain video frame, node A finds out that three

segments p_1, p_5 and p_9 are missing. Then it starts searching for the missing segments from its partners. Here, there are four nodes in A's partner list, known as nodes B, C, D and E. Nodes B, C and E have the missing segment p_1, nodes A, D and E have segment p_5, while nodes C, D and E have p_9. How to effectively fetch the missing segments p_1, p_5, p_9 to be played back at node A with the best video quality over wireless networks is an important issue that will be addressed in this chapter.

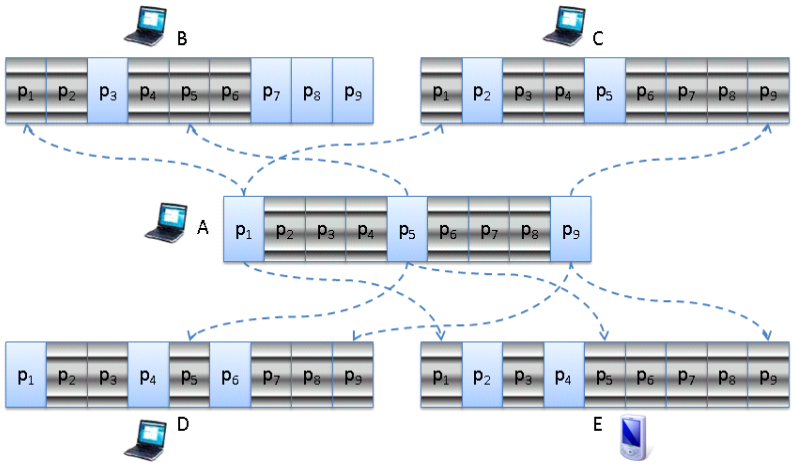


Figure 5.1 : Illustration of scheduling in P2P video streaming over wireless networks, where blue blocks denote missing segments while grey blocks denote available segments.

Despite the increasing popularity of P2P video streaming services in the Internet environment, huge challenges still exist before the wide deployment in wireless networks. The majority of the current research results and commercial products of P2P video streaming, which are based on the overlay network architecture, cannot be directly applied to wireless networks. They explicitly or implicitly assume that the network layers below the P2P overlay networks are in perfect condition, by either ignoring the lower layers or assuming an error-free Internet environment. However, in wireless networks, due to the time-varying channel characteristics and high heterogeneity, this assumption greatly affects the user-perceived video quality at the receiver end. Ignoring the underlying network proximity poses a challenge for wireless P2P video streaming services – the nodes that are adjacent

to each other at the overlay layer may actually be far from each other at the underlying network topology, especially in highly heterogeneous networks. The exacerbation of this problem in wireless networks leads to the degradation of video performance. For instance, in Figure 5.2, the same underlying network topologies ((a) and (c)) have totally different P2P overlay layer topologies ((b) and (d)).

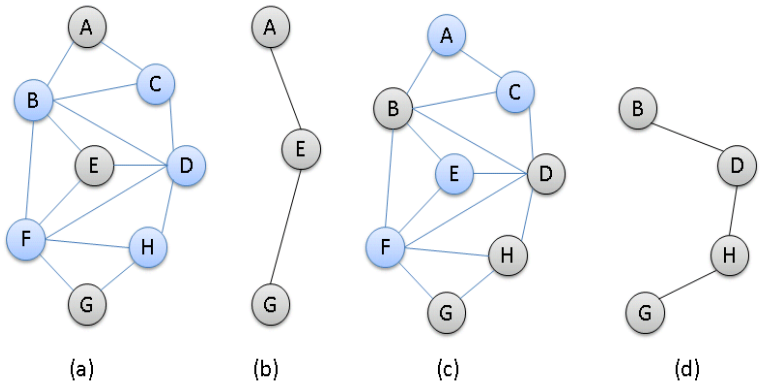


Figure 5.2 : The same underlying network topology vs. two different P2P overlay network topologies.

To further illustrate this, we take CoolStreaming [58] as an example, which is a data-driven overlay network framework for live media streaming service in the Internet. Its scheduling algorithm considers two overlay level constraints - the playback deadline for each segment and the streaming bandwidth from the partners. However, in wireless environments, even if the theoretical bandwidth is large enough and the playback deadline for each segment meets the requirement, the time-varying underlying link quality can still greatly affect the transmission, influencing the overall effectiveness of the scheduling algorithm.

Generally, the time-varying wireless channel conditions, the higher network heterogeneity, the interactions among different network layers, as well as the tendency of more peers joining and leaving activities in wireless interactive multimedia systems, lead to the fact

that the existing P2P scheduling algorithms for video streaming are not suitable for wireless environments [62, 63]. To meet these challenges, we propose a novel scheduling algorithm based on the cross-layer concept that can achieve significantly-improved user-perceived video quality while the computational overhead is evenly distributed to each peer node. To achieve this, a cross-layer based distortion-delay framework is implemented as the integral and essential part of the proposed distributed utility-based scheduling algorithm at each peer node, where functions provided by different network layers can be optimized for the P2P video streaming services in wireless networks. One of the major contributions of this work is that for the first time, it formulates and quantifies the performance impact of various network parameters residing at different layers of the P2P wireless environment. Another important aspect of the proposed scheduling algorithm is that the joint optimization is employed in a distributed fashion, decreasing the computational complexity at each node and increasing the possibility of deployment. The experimental results demonstrate that significant performance enhancement can be achieved by using the proposed algorithm.

5.2 Related Work

In [62], Delmastro presented a performance evaluation of Pastry system [64] running on a real ad hoc network. An optimized solution called CrossROAD was defined to exploit the cross-layer architecture to reduce the communications overhead introduced in Pastry. By providing an external data sharing module known as Network Status (NeSt), the system is able to store all routing information in a single routing table, including the logical address of all nodes taking part in the overlay, and the optional information about each node's behavior. It directly exploits the network routing protocol that collects every topology change by periodically sending its Link State Update (LSU) packets, and directly updates its own routing table and the related abstraction in NeSt. Therefore, CrossROAD becomes aware of topology changes with the same delays of the routing protocols.

Further, Counti *et al.* [63] developed and tested a Group-Communication application

on top of different P2P substrates. They highlighted the advantages of a solution based on cross-layer optimization and demonstrated the limitations of legacy P2P systems in Mobile Ad Hoc Networks (MANETs). In the proposed system, CrossROAD exploits cross-layer interactions with a proactive routing protocol (OSLR) to build and maintain the Distributed Hash Table (DHT). These interactions are handled by the Network State module, which provides well-defined interfaces for cross-layer interactions throughout the protocol stack. Specifically, each node running CrossROAD piggybacks advertisements of its presence in the overlay into routing messages periodically sent by OLSR. Thus, the node in the network becomes aware of the other peers in the overlay network.

Barbera *et al.* [65] proposed an approach to carry out P2P video streaming performance analysis and system design by jointly considering both a P2P overlay network and the underlying packet networks. A fluid-flow approach has been adopted to simulate the behavior of the network elements supporting a simulated overlay network, which consists of implementations of P2P clients. The P2P SplitStream video streaming protocol is considered as a case study to demonstrate that the tool is able to capture performance parameters at both the overlay and the packet network levels.

Si *et al.* [66] presented a distributed algorithm for scheduling the multiple senders for multi-source transmission in wireless mobile P2P networks that maximize the data rate and minimize the power consumption. The wireless mobile P2P networks was formulated as a multi-armed bandit system. The Gittins index-based optimal policy was used to increase the receiving bit rate and lifetime of the network.

Mastronarde *et al.* proposed a distributed framework for resource exchanges that enables peers to collaboratively distribute available wireless resources for P2P delay-sensitive networks based on the quality of service requirements, the underlying channel conditions and the network topology [67]. Also, a scenario with multiple pairs of peers transmitting scalably encoded video to each other over a shared infrastructure was considered. Distributed algorithms were designed for P2P resource exchanges, including collaborative admission

control, path provisioning and air-time reservation at intermediate nodes.

Nevertheless, most of the current works on cross-layer P2P networks either only address issues under the wireline environment or simply ignore the time-varying channel quality that takes place at the physical layer, which is not suitable for video applications in wireless networks. Further, only heuristic solutions can be found in the literature, lacking mathematical quantification or algorithmic formulation from the users' perspectives.

5.3 System Model

Generally speaking, two categories of methodologies for overlay construction can be adopted for interactive P2P video streaming, known as tree-based and data-driven [68]. Although the vast majority of proposals to date can be categorized as tree-based approach, they suffer from the problem that the failure of nodes, especially those residing higher in the tree, may disrupt delivery of data to a large number of users and thus potentially result in poor video transmission performance. Therefore, in this chapter we adopt the data-driven overlay design which does not construct or maintain an explicit structure for data delivery. Furthermore, we adopt the gossip-based protocols in the system for group communications, which have attractive scalability and reliability properties [69]. In a typical gossip-based protocol, a node sends a newly generated message to a set of randomly selected nodes. These nodes do similarly in the next round, and so do other nodes until the message is spread to all nodes. Gossip-based algorithms do not need to maintain an explicit network structure.

At the overlay layer, we consider the scheduling algorithm for the missing segments. We adopt the mechanism of Buffer Map (BM) [58] representation to exchange information of missing segments between the neighboring nodes in the overlay network. A scheduling algorithm is to find the optimal node from the partner list to fetch the expected missing segments, given the BMs of the node and those of its partners. These missing segments need to be made available before the playback deadline.

In the following subsections, we describe the key functions that are jointly considered in the proposed distributed P2P scheduling algorithm and their interactions.

5.3.1 Video Distortion

In P2P wireless environments, channel conditions are highly asymmetrical and heterogeneous, so video transmission mainly suffers from unreliable channel conditions and excessive delays. We calculate the video distortion at the application layer. Usually, the performance of a given P2P video streaming system in terms of user-perceived video quality is evaluated by estimating the distortion of a set of frames decoded at the receiver node, that is,

$$\mathcal{E}[\mathcal{D}] := \sum_{f} \mathcal{E}[d_f], \tag{5.1}$$

where $\mathcal{E}[d_f]$ is the expected distortion of frame f. At the video encoder, which resides in the application layer in the proposed system, we consider the quantization step size (QP) q for each packet as the target variable for optimization.

In video encoder, each video frame is divided into 16×16 macroblocks (MB), which are numbered in scan order. The packets are constructed in such a way that each packet consists of exactly one or several rows of MBs and can be independently decodable [2]. When a segment is missing and the scheduling algorithm fails to find it from the node's partners or cannot be fetched within the playback deadline constraint, temporal replacement is adopted as the error concealment strategy. The motion vector of a missing MB is estimated as the median of the motion vectors of the nearest three MBs in the preceding row. If the previous row is also lost, the estimated motion vector is set equal to zero. The pixels in the previous frame, pointed to by the estimated motion vector, are used to replace the missing pixels of the current frame.

Given the dependencies introduced by the error concealment scheme, the expected end-to-end (e2e) distortion of packet i of video frame n can be calculated at the encoder by

using the ROPE method as [8]

$$\begin{aligned}\mathcal{E}[\mathcal{D}_{n,i}^{e2e}] &= (1-\rho_{n,i})\mathcal{E}[\mathcal{D}_{n,i}^{r}] \\ &+ \rho_{n,i}(1-\rho_{n,i-1})\mathcal{E}[\mathcal{D}_{n,i}^{lr}] + \rho_{n,i}\rho_{n,i-1}\mathcal{E}[\mathcal{D}_{n,i}^{ll}]\end{aligned} \tag{5.2}$$

where $\rho_{n,i}$ is the loss probability of packet i. $\mathcal{E}[\mathcal{D}_{n,i}^{r}]$ is the expected distortion of packet i when it is successfully received and $\mathcal{E}[\mathcal{D}_{n,i}^{lr}]$ and $\mathcal{E}[\mathcal{D}_{n,i}^{ll}]$ are the expected distortions after concealment when packet i is lost but packet $(i-1)$ is received and lost, respectively. Thus, the expected end-to-end video distortion is accurately calculated by ROPE assuming knowledge of the instantaneous network conditions and becomes the objective function in the proposed optimized system [17]. For a given video packet i, the expected packet distortion only depends on packet error rates $\rho_{n,i}$ and $\rho_{n,i-1}$and QP q. Considering the fact that the individual contribution of each path is continuously updated, these parameters are updated after each packet is encoded. The prediction and calculation of packet loss rate $\rho_{n,i}$ will be discussed in the following subsection.

It is known that an error-resilient transcoder can improve video quality in the presence of errors while maintaining the input bit rate over wireless channels [70]. Thus, we use transcoding and re-quantization at the intermediate nodes when necessary. The increased computational complexity is small because a node only fetches the missing segments from the neighboring nodes, usually limiting the transcoding frequency to only one or two times for each scheduling period. The scheduling only happens when a given segment is detected missing. Additionally, for the ROPE algorithm to work, we regard the videos on the node that the segment might be fetched from as the reference. Usually, this implies that the original video at the source node is used. However, for the intermediate nodes, we use the currently available videos at the neighboring nodes to achieve the "relative" reference. In actuality, these neighboring nodes are currently serving as the "source" nodes. Table 5.1 evaluates the accuracy of this "relative" reference model, using the same video environment as what is described in Section 6.6, where QP is set to 25. Considering the fact that

Packet Loss Rate (PLR) is usually less than 5%, this method provides reasonable distortion predication. On the other hand, when PLR increases, the "relative" reference model leads to degraded expected video quality quantified by peak signal-to-noise ratio (PSNR), which will be utilized by the proposed scheduling algorithm (Algorithm 2 in Section 5.5.1) to effectively eliminate its opportunity of being fetched. In this way, we can guarantee to always fetch the videos with better quality.

Table 5.1 : Performance Evaluation of "relative" reference for the ROPE algorithm

Packet Loss Rate (%)	1	2	3	4	5
ROPE PSNR	36.4858	35.9595	35.3681	34.9538	34.4246
"Relative" Reference PSNR	35.9376	34.9507	34.0096	33.2573	32.5349

5.3.2 AMC and Truncated ARQ

Adaptive Modulation and Coding (AMC), is adopted for link adaptation to increase the overall system capacity. The link quality is characterized by the received signal-to-noise ratio (SNR) γ. Perfect Channel State Information (CSI) is available at the receiver and the corresponding mode selection is fed back to the transmitter without error or latency. The general Nakagami-m model is used to describe γ statistically with a probability density function (pdf) [9]

$$\rho_\gamma(\gamma) = \frac{m^m \gamma^{\gamma-1}}{\bar{\gamma}^m \Gamma(m)} exp(-\frac{m\gamma}{\bar{\gamma}}), \tag{5.3}$$

where $\bar{\gamma} := E(\gamma)$ is the average received SNR, $\Gamma(m) := \int_0^\infty t^{m-1}e^{-t}dt$ the Gamma function, and m the Nakagami fading parameter ($m \geq 1/2$).

The instantaneous varying channel quality adaptation is the major issue to be addressed for P2P scheduling. However, for longer intervals, channel quality can be assumed to be

stable. These two are not in conflict because of different time granularity. Furthermore, we assume that error detection based on CRC is perfect, provided that sufficiently reliable error detection CRC codes are used.

By using AMC at the physical layer, the combination of different constellation of modulation and different rate of error-control codes are chosen based on the time-varying channel condition. We list all the AMC schemes (s) adopted in this work in Table 2.1, with each scheme consisting of a pair of modulation scheme a and FEC code c as in 3GPP, HIPERLAN/2, IEEE 802.11a, and IEEE 802.16 standards [9, 10, 20, 21]. To simplify the AMC design, we employ the following approximate Bit Error Rate (BER) expression:

$$\epsilon_s(\gamma) = \frac{x_s}{e^{\gamma \times y_s}}, \tag{5.4}$$

where s is the scheme index and γ the received SNR. Parameters x_s and y_s are obtained by fitting (5.4) to the exact BER. In this chapter, we will refer to the term "scheme s" to imply a specific choice of modulation and coding scheme. Then, we adopt the following model to achieve the corresponding packet loss rate $\rho^s_{n,i}(\gamma)$

$$\rho^s_{n,i}(\gamma) = 1 - (1 - \epsilon^s(\gamma))^{L^s_{n,i}}, \tag{5.5}$$

where $L^s_{n,i}$ is the packet length of packet i of frame n. Denote S the number of available AMC schemes. Thus, the average packet error rate (PER) can be represented as

$$\begin{aligned} \bar{\rho}^s_{n,i} &= \frac{\sum_{s=1}^{S} \int_{\gamma_s}^{\gamma_{s+1}} \rho^s_{n,i}(\gamma) p(\gamma) d\gamma}{\int_{\gamma_1}^{+\infty} p(\gamma) d\gamma} \\ &= \frac{\sum_{s=1}^{S} \int_{\gamma_s}^{\gamma_{s+1}} [1 - (1 - \epsilon^s(\gamma))^{L^s_{n,i}}] p(\gamma) d\gamma}{\int_{\gamma_1}^{+\infty} p(\gamma) d\gamma}, \end{aligned} \tag{5.6}$$

where $\int_{\gamma_1}^{+\infty} p(\gamma) d\gamma$ is the probability that the channel has no deep fades and at least one AMC scheme can be adopted.

Due to the fact that the underlying bit streams represent highly correlated image contents, truncated ARQ is adopted at the data link layer. If an error is detected in a packet, a retransmission is needed. Otherwise, no retransmission is necessary. Because only finite

delays and buffer sizes can be afforded in practice, the maximum number of ARQ retransmissions should be bounded [71]. Denote the maximum number of retransmissions allowed per packet as N_r^{max}, which can be specified by dividing the maximum allowable system delay over the round trip delay required for each retransmission. Therefore, if a packet is not received correctly after N_r^{max} retransmissions, the packet is dropped ($N_r^{max} = 0$ means no retransmission).

Finally, the packet loss rate at the data link layer can be expressed as

$$\rho_{n,i}^s = (\bar{\rho}_{n,i}^s)^{N_r^{max}+1}. \tag{5.7}$$

5.3.3 Cross-layer Interactions

The interactions of network functions residing in different network layers will jointly affect the user-perceived video quality, thus can be utilized during scheduling. As shown in Figure 5.3, the fed-back channel quality affects the choice of Modulation and Channel Coding (MCC) scheme, while the video encoding determines the packet length by using the appropriate encoding parameters such as quantization step size (QP) q and intra- or inter- prediction mode. Then, the used MCC scheme and packet length jointly determine the packet loss rate ρ. Further, the transmission rate r and ARQ are affected by the MCC scheme and packet loss rate ρ, respectively. Also, the transmission delay can be achieved by the joint effect of transmission rate r and retransmission, while the estimated video distortion is jointly affected by packet loss rate ρ and video encoding. Finally, the transmission delay and estimated video distortion $\mathcal{E}[\mathcal{D}]$ interact with each other to achieve the final video quality. Therefore, by fine-tuning the system parameters residing in different layers in a systematic way during scheduling, video performance can be greatly improved.

5.4 Problem Formulations

In this section, we first present the P2P scheduling problem, and then we discuss the distributed cross-layer optimization, which is the core of the scheduling problem.

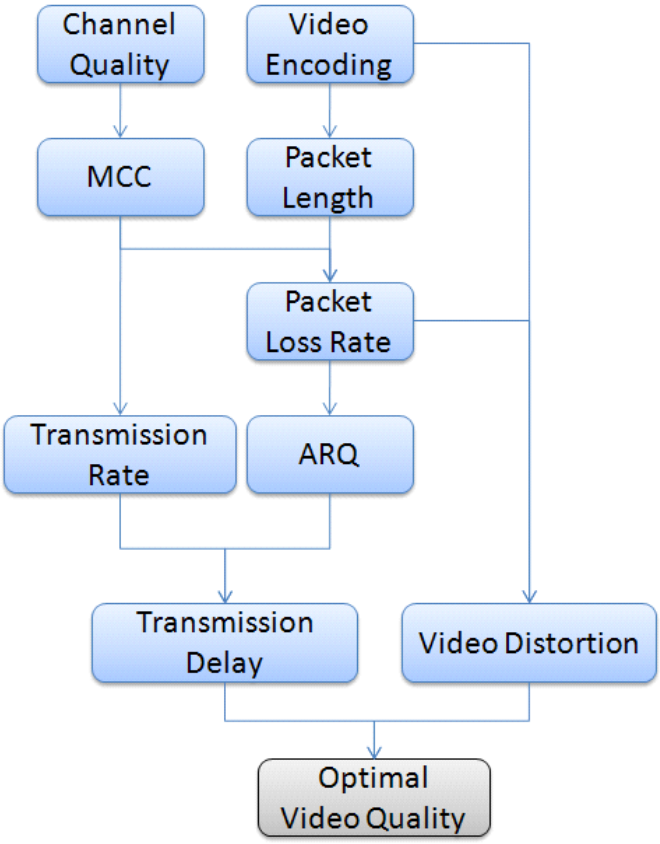


Figure 5.3 : The interactions among P2P video streaming network functions residing in different network layers that can jointly affect the user-perceived video quality in wireless networks.

5.4.1 P2P Scheduling Problem

As shown in Figure 5.1, when a missing segment is detected, the node needs to fetch it from the neighboring nodes. To make this work, additional signaling is needed so that information among peer nodes can be exchanged. The problem here is how to exchange information between the current working node and its neighboring nodes. Usually, when the current working node finds out a missing segment, it will send out "request" messages to its neighboring nodes. The neighboring nodes will perform the cross-layer optimization independently and then send back the results through "response" messages. After gathering these "response" messages, the working node will choose the best node to fetch the missing

segment from. Furthermore, the proposed algorithm should strike to meet two constraints for scheduling: the user-perceived video quality and the playback deadline for each video packet.

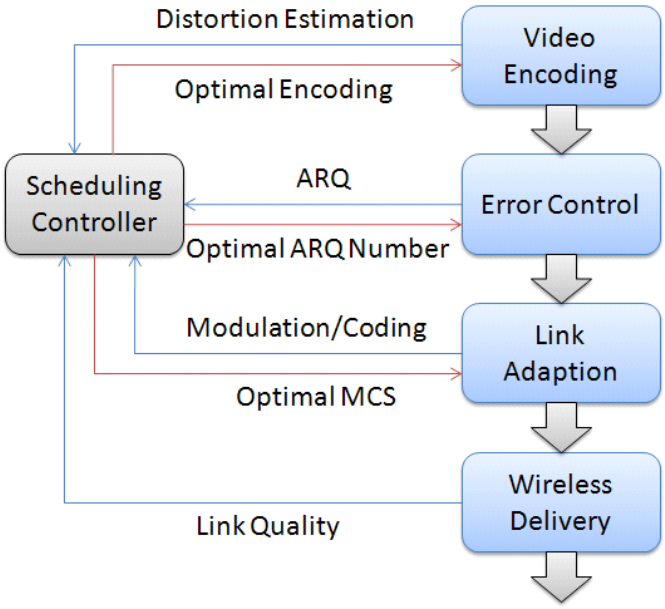


Figure 5.4 : The system platform for the proposed quality-driven distributed scheduling of P2P video streaming services over wireless networks.

The core of the scheduling algorithm is the cross-layer optimization that is distributed and it is performed by each neighboring node. As shown in Figure 5.4, we consider video encoding, ARQ, and link adaptation. The scheduling controller is able to adjust the network behavior by equipping it with key system parameters of each network layer, such as the quantization step size q at the application layer, the retransmission N_r^{max} at the data link layer, and the modulation and channel coding (MCC) scheme s at the physical layer. The cross-layer optimization algorithm is implemented by the controller of each neighboring node [72].

In the following subsection, we will concentrate on the formulation of the major challenge of this scheduling problem: how to perform cross-layer optimization to determine the

optimal system parameters on each neighboring node. We model the transmission delay and queuing delay in terms of bandwidth and key system parameters of different network layers under the design constraint, and then we formulate it into a distortion-delay problem.

5.4.2 Cross-layer Optimization on Each Neighboring Node

Real-time video streaming has strict delay requirements. In this chapter, we formulate the end-to-end packet delay, including the transmission delay at the MAC layer and the queuing delay at the network layer. Most of the previous studies focus only on the transmission delay, while in many cases queuing delay accounts for a significant portion of the total delay over a hop. Sometimes, the delay through a node with many packets in queue but short transmission time could be larger than through the one with fewer packets in the queue but longer transmission delay.

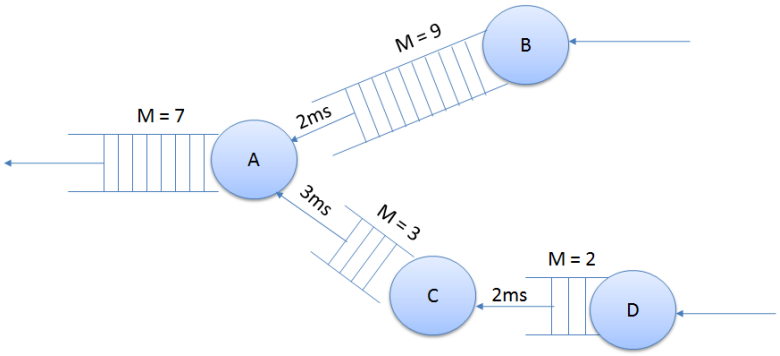


Figure 5.5 : The impact of queuing delay and transmission delay on P2P scheduling over wireless networks.

Consider the example shown in Figure 5.5, where the impact of queuing delay and transmission delay is illustrated. The number M denotes the number of packets in the queue of the network layer, waiting to be served by the MAC layer. Suppose that the bandwidth of each link is 10 Mbits/second and the packet length is 1000 bytes. When a packet is missing at node A, both node B and D have this packet. Without considering

queuing delay, path $D \rightarrow C \rightarrow A$ gives a transmission delay of 5ms, while path $B \rightarrow A$ gives a transmission delay of 2ms. Therefore, it appears that node B is a better node for fetching the missed packet. However, when queuing delay is also considered, path $D \rightarrow C \rightarrow A$ gives an end-to-end delay of 10ms while path $B \rightarrow A$ gives 11ms. Thus, node D is actually the better choice.

Thus, the expected end-to-end packet delay includes transmission delay, queueing delay and propagation delay. For any packet at hop κ, the average transmission delay for one attempt can be expressed as

$$\bar{t}^s_{n,i}(\kappa) = \frac{\sum_{s=1}^{S} \int_{\gamma_s}^{\gamma_{s+1}} \frac{L^s_{n,i}}{R_s \times B_w} p(\gamma) d\gamma}{\int_{\gamma_1}^{+\infty} p(\gamma) d\gamma}, \tag{5.8}$$

where B_w is the bandwidth. Furthermore, using N_r^{max}-truncated ARQ, the average packet transmission attempts can be calculated as

$$\begin{aligned} \bar{N}^r_{n,i}(\kappa) &= 1 + \bar{\rho}^s_{n,i} + (\bar{\rho}^s_{n,i})^2 + ... + (\bar{\rho}^s_{n,i})^{N_r^{max}} \\ &= \frac{1 - (\bar{\rho}^s_{n,i})^{N_r^{max}+1}}{1 - \bar{\rho}^s_{n,i}}, \end{aligned} \tag{5.9}$$

where N_r^{max} is the maximum allowed retransmissions. Thus, the packet transmission delay can be expressed as

$$E(\mathfrak{tr}^s_{n,i}(\kappa)) = \bar{t}^s_{n,i}(\kappa) \times \bar{N}^r_{n,i}(\kappa) \tag{5.10}$$

We use an exponentially distributed services time at the MAC layer. It is known that an exponentially distributed services time has the memoryless property of the packet service time, as the head-of-line packet only needs to finish a residue packet service time when the new packets arrive [73]. Thus, if there are M_κ packets in the queue when a new packet reaches node κ, the End-to-End Delay (EED) metric can be defined as

$$E(\mathfrak{t}^s_{n,i}(\kappa)) = \sum_{p_\kappa=1}^{M_\kappa} E_{p_\kappa}(\mathfrak{tr}^s_{n,i}(\kappa)) \tag{5.11}$$

which means that the total delay passing through hop κ equals the MAC service time of those packets in the queue ahead of the current packet (the ith packet of frame n) plus the MAC service time of the current packet itself.

Consider an end-to-end path including $\mathbb{H}$ hops. The end-to-end video distortion estimation metric for the path can be defined as

$$E(D_{n,i}^{e2e}) = \sum_{\kappa=1}^{\mathbb{H}} E(\mathfrak{t}_{n,i}^{s}(\kappa)) \tag{5.12}$$

To formulate the cross-layer based optimization problem that is distributed to each neighboring node, we denote a vector $\mu_{n,i}^{s} := [q_{n,i}, N_{n,i}, s_{n,i}]$ for packet i of frame n, which includes the quantization step size QP q, the number of retransmissions N and the AMC scheme s. Then $(\mu_{n,i}^{m})^*$ is the optimal vector that can ensure the best video quality under the constraint of the playback deadline. Denote by Γ all possible choices of μ, and then the distributed utility-based scheduling problem can be formulated as the following delay-distortion problem

$$\begin{aligned} (\mu_{n,i}^{s})^* &= arg \min_{\mu \in \Gamma} E[D_{n,i}^{e2e}(\mu_{n,i}^{s})] \\ s.t. \qquad & E(\mathfrak{t}_{n,i}^{e2e}(\mu_{n,i}^{s})) \leq \mathcal{T}_{n,i}^{dd}, \end{aligned} \tag{5.13}$$

where $\mathcal{T}_{n,i}^{dd}$ is the playback deadline for the missing packet i of frame n. Thus, the cross-layer optimization problem turns into choosing the best partner node for fetching the missing packet i, by jointly optimizing system parameters residing in different network layers.

5.5 Problem Solutions

In this section, we first propose the algorithm for the distributed utility-based P2P scheduling, and then we provide the solution to the cross-layer optimization problem, followed by the complexity analysis.

5.5.1 The Proposed P2P Scheduling Algorithm

Denote $\mathbf{P}_{\kappa}$ the parter list of node κ. The overall procedure of the proposed distributed pull-based scheduling algorithm is shown in Algorithm 2.

One of the deciding factors for the value of timer to_{κ} at the current node κ is the delay deadline. Also, the time spent on all the delays needs to be factored in. Here,

1. Node κ detects a missing segment before playback ;
2. Node κ starts a timer to_κ and sends initial request messages to the nodes in $\mathbf{P}_\kappa$, with delay requirement $\mathcal{T}_{n,i}^{dd}$;
3. **if** *Node* p_κ *(*$p_\kappa \in \mathbf{P}_\kappa$*) does not have the requested segment* **then**
   p_κ forwards the request message to its partner nodes in $\mathbf{P}_{\kappa+1}$;
   else
   p_κ performs cross-layer optimization to determine the optimal system parameters ;
   end
4. Node p_κ sends the scheduling results and selected system parameters back to node κ;
5. Timer to_κ at node κ triggers or all partner nodes respond;
6. Node κ chooses the optimal node κ_o;
7. Node κ sends a final request message to κ_o to fetch the missing segment;
8. Node κ_o sends the requested segment back to κ.

Algorithm 2: The overall scenario of the proposed distributed pull-based scheduling algorithm for P2P video streaming over wireless networks.

the scheduling algorithm runs for each missing segment. Specifically, whenever a missing segment is detected, the scheduling algorithm will be performed to find the best node to fetch from.

As shown in Algorithm 2, the most difficult part of the algorithm is the cross-layer optimization shown in step 3. In the following subsection, we will present the solution to this subproblem.

5.5.2 Solution for the Cross-layer Optimization Problem

Since the formulated problem at each neighboring node in (5.13) is actually a constrained minimization problem, it can be solved by Lagrangian relaxation. That is, it can be represented as the following Lagrangian cost function:

$$L_\lambda = arg \min_{\mu\in\Gamma}\left\{\mathcal{E}[\mathcal{D}_{n,i}^{e2e}] + \lambda\big(E(\mathfrak{t}_{n,i}^{e2e}) - \mathcal{T}_{n,i}^{dd}\big)\right\} \tag{5.14}$$

Thus, to solve (5.13) is to solve (5.14) and to find the optimal λ^* such that

$$L_{\lambda^*}((\mu_{n,i}^s)^*) = arg\ min\ L_{\lambda^*}.(\mu_{n,i}^s) \tag{5.15}$$

According to [74], this λ^* exists. Based on (5.13), the video distortion can be expressed as a function of the delay deadline as $f(t^{dd})$, where t^{dd} is the corresponding delay deadline. To achieve the optimal λ^*, we first describe the following theorem.

Theorem: $f(t^{dd})$ is a non-increasing function.

Proof: Let $t_1^{dd} < t_2^{dd}$. Then, $\mu_{n,i}^{1*} := [q_{n,i}^{1*}, N_{n,i}^{1*}, m_{n,i}^{1*}]$ and $\mu_{n,i}^{2*} := [q_{n,i}^{2*}, N_{n,i}^{2*}, m_{n,i}^{2*}]$ are the optimal parameters for packet i of frame n under the two playback deadlines, respectively. Due to $t_1^{dd} < t_2^{dd}$, $\mu_{n,i}^{1*}$ is also a possible solution for (5.13) when the playback deadline is t_2^{dd}. However, $\mu_{n,i}^{2*}$ is the optimal solution in this case. Therefore, $f(t_1^{dd}) > f(t_2^{dd})$.

The parameter λ ranges from zero to infinity. Hence, its optimal value λ^* can be obtained by using a fast convex recursion based on the bisection algorithm illustrated in Algorithm 3.

With the bisection algorithm, the initial values of λ_1 and λ_2 are chosen heuristically based on experience. Thus, $\mathcal{T}_{n,i}^{dd}(\lambda)$ is made recursively close to $\mathfrak{t}_{n,i}$ until the optimal λ^* is

Input: $\epsilon, \mathcal{T}_{n,i}^{dd}, \lambda_1, \lambda_2$, where

$\lambda_1 \neq \lambda_2$ and

$\mathfrak{t}_{n,i}(\lambda_1) < \mathcal{T}_{n,i}^{dd}(\lambda) < \mathfrak{t}_{n,i}(\lambda_2)$

Output: λ^*

repeat

  $\lambda_t \leftarrow \frac{\lambda_1+\lambda_2}{2}$;

  if $\mathcal{T}_{n,i}^{dd}(\lambda_t) > \mathcal{T}_{n,i}^{dd}$ **then**

    $\lambda_1 \leftarrow \lambda_t$;

  end

  if $\mathcal{T}_{n,i}^{dd}(\lambda_t) < \mathcal{T}_{n,i}^{dd}$ **then**

    $\lambda_2 \leftarrow \lambda_t$;

  end

until $|\mathcal{T}_{n,i}^{dd}(\lambda_t) - \mathcal{T}_{n,i}^{dd}| \leq \epsilon$;

$\lambda^* = \lambda_t$;

return λ^* ;

Algorithm 3: Calculating the optimal λ^*.

found. With this λ^*, the optimal parameters $(q^*_{n,i}, N^*_{n,i}, s^*_{n,i})$ can be finally obtained for any packet i of frame n.

5.5.3 Complexity Analysis

In the proposed scheduling algorithm, additional signaling is involved for the system to work. However, the propagation delay is normally negligible compared to the transmission and queueing delay, while additional queuing delay is minimal since signaling messages usually take higher priority. Based on Algorithm 2, the time cost due to signaling is

$$T_o = V(to_\kappa) + T^{propa}, \tag{5.16}$$

where $V(to_\kappa)$ is the value of timer to_κ and T^{propa} the propagation delay for the final segment request. This value is jointly determined by the delay bound, channel quality SNR, and the number of peers in the list. Thus, the increased complexity resulting from additional signaling is negligible.

The cross-layer optimization part of the scheduling algorithm is distributed to every peer node. Normally, the node with missing segments only needs to wait until the timer triggers to decide the optimal peer node to fetch the packet from. Thus, the proposed scheduling can effectively offload the computational intensity of each peer node.

5.6 Experiments and Performance Analysis

In this section, we introduce the experimental environment and analyze the performance enhancement due to the proposed optimization. The extensive experiments we have conducted using the proposed P2P scheduling algorithm for interactive video streaming services over wireless networks have shown its significant performance improvement.

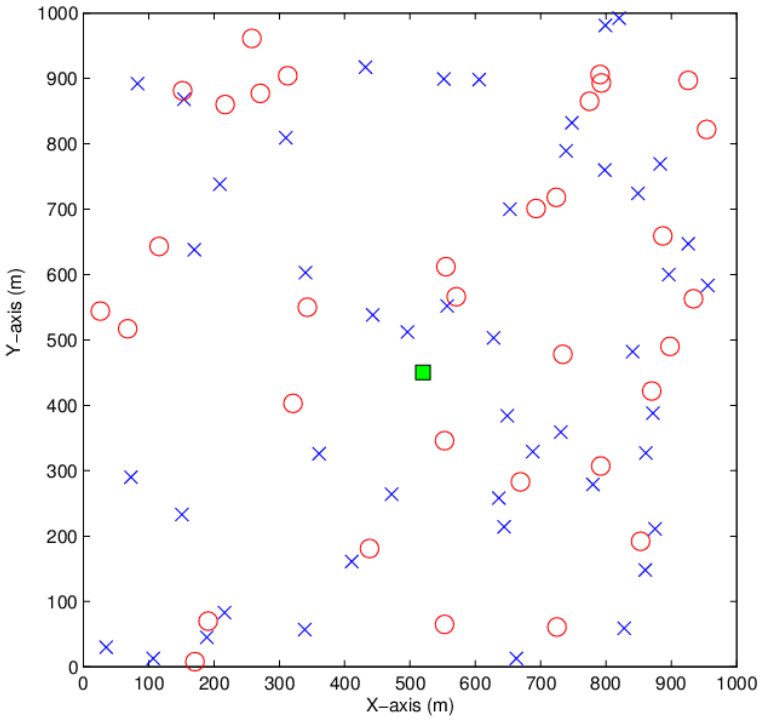


Figure 5.6 : The experimental topology of P2P video streaming in wireless networks.

5.6.1 The Experimental Environment

During the design of experiments, we randomly distributed 80 nodes in an area of $1000m \times 1000m$. We set the transmission radius of every node to be $150m$. Further, we allow the nodes to randomly join or leave the overlay networks. However, their locations are fixed. One snapshot of the adopted topology is illustrated in Figure 5.6. All nodes shown in the figure are physical nodes that exist in the networks. However, the nodes represented by a *cross* only exist at the underlying layers, while the nodes represented by a *circle* also exist at the overlay layers. Further, the *square* node in the center is the streaming server.

In this chapter, we assume that the mesh network topology is fixed over the duration of the video session and that each video flow can reserve a predetermined transmission opportunity interval prior to the transmission, and thus contention-free access to the medium is

provided. This reservation mechanism can be performed based on the Hybrid Coordination Function (HCF) Controlled Channel Access (HCCA) protocol of IEEE 802.11e [75].

Each P2P node maintains two buffers, which are the synchronization buffer and the cache buffer. A received packet is first placed into the synchronization buffer for the corresponding frame. We set the buffer size at each node to be large enough to hold 150 packets. The packet size is corresponding to one slice of the same frame. The packets in the synchronization buffer will be dispatched into one stream when packets with continuous sequence numbers have been received. A Buffer Map (BM) is used to represent the availability of the latest packets of different frames in the buffer. This information is exchanged periodically among neighboring nodes in order to determine which node has the missing packet. With the exchange of BM information, the newly joined node can also obtain the video availability information from a set of randomly selected nodes in the list. We do not address the free-rider issue [76] in this chapter, so every P2P node is open to share its video segment as long as there is an incoming request.

The proposed scheduling algorithm runs on every P2P node. At the streaming server, we use the QCIF (176×144) sequence "Foreman" as the video streaming content, which is coded at a frame rate of 30 frames/second. We repeat the first 100 frames of the video clip to form a longer video clip, where every I frame is followed by 9 P frames. In the experiments, each communication link has an average SNR $\bar{\gamma}$ based on (5.3). The propagation delay on each link is set to $10\mu s$, while the average bandwidth is randomly set to $150K, 200K$ and $250K$ (symbols/second) to simulate a heterogenous network environment.

To avoid prediction error propagation, a 10% macroblock level intra-refreshment is used. When a packet is lost during transmission, we use the temporal-replacement error concealment strategy. The motion vector of a missing MB is estimated as the median of the motion vectors of the nearest three MBs in the preceding row. If that row is also lost, the estimated motion vector is then set equal to zero. The pixels in the previous frame, pointed to by the estimated motion vector, are used to replace the missing pixels in the current

frame. At the streaming server, the values of QPs are chosen from 1-50. The AMC scheme is chosen from Table 2.1. We use PSNR as the performance metric to compare the video quality achieved by using the proposed distributed P2P scheduling algorithm with that achieved by using the scheduling methods without cross-layer optimization or using only partial cross-layer optimization on each peer node, given the same wireless environment. Specifically, for the cross-layer optimization of the compared scheduling algorithms, we set some or all of the system parameters that we have considered in this chapter to fixed values. To verify the performance enhancement of different videos with different slow/medium/fast video frame rates, we also verify the video performance improvement under different packet delay bounds, ranging from 5ms to 50ms.

5.6.2 Performance Analysis

As shown in the experimental topology of Figure 5.6, the circle nodes consist of the overlay network, while the cross nodes represent the underlying nodes that are hidden from the overlay layer. It can be observed that some nodes at the overlay layer are not even reachable without consideration of the hidden connections in the underlying layers. In addition, the quality of all transmission links are varying with time.

We first analyze the amelioration of scheduling failure rate by using the proposed distributed scheduling algorithm. A scheduling failure event means that the node fails to find any partner node that contains the missing packet within the constraint of the playback deadline. When this happens, the node has to fetch the packet from the streaming server or re-construct the packet by using error concealment as described in Section 5.3.1. Therefore, a scheduling failure brings about performance degradation. As illustrated in Figure 5.7, the x-axis represents the playback deadline in ms, while the y-axis is the scheduling failure rate percentage-wise. We can observe that the proposed algorithm can significantly reduce the average scheduling failure rate, especially in the deadline-stringent environment. This is because the proposed scheduling algorithm can dynamically adapt the encoding param-

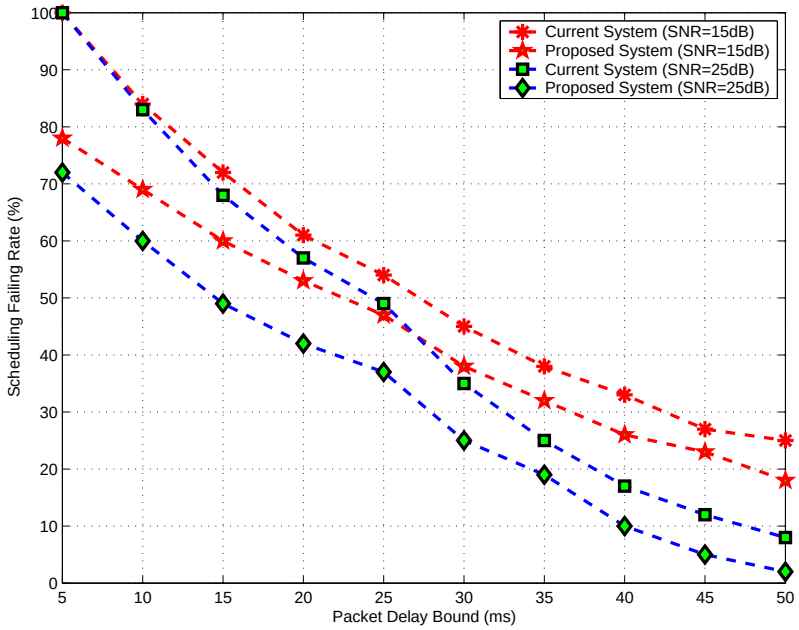


Figure 5.7 : Scheduling failure rate comparison between the existing P2P scheduling algorithm and the proposed distributed P2P scheduling algorithm.

eter, retransmission and AMC scheme based on the instantaneous channel and network conditions.

In the following, we compare the average frame PSNR difference of the video sequence using the two different scheduling algorithms as shown in Figure 5.8, where the proposed scheduling algorithm can achieve 4-14dB performance gain. As $\mathcal{T}_{n,i}^{dd}$ becomes more stringent, the performance gain increases as well. Thus, the proposed scheduling algorithms is especially suitable for real-time multimedia applications. In this figure, we also quantify the performance impact of different system parameters on the receiver end video performance. Dynamic modulation and coding is identified as the most important design parameter that has the biggest performance contribution on the proposed design, which is shown in the figure by using a fixed AMC scheme ($s = 3$). This points to the fact that the proposed scheduling algorithm is especially useful in wireless networks.

Next, we evaluate the performance impact of the design of timer to_κ as shown in Fig-

ure 5.9. In the experiments, we do not consider the transmission time for signaling. The purpose of this demonstration is to show how the timer value can affect the video performance. However, the result shown here can still give us a great insight considering the fact that the proposed algorithm can be easily extended by changing the configurable value of to_κ. It is known from the figure that the timer with a small value leads to performance degradation, which is reasonable because the introduced signaling for distributed scheduling needs some time for propagation. If the timer value is too small, it may not have enough time for the signaling propagation before optimization can be finished. We can also observe that the timer with a very large value can also lead to degradation, which is also reasonable because the more time is spent on signaling, the less time will remain for video transmission.

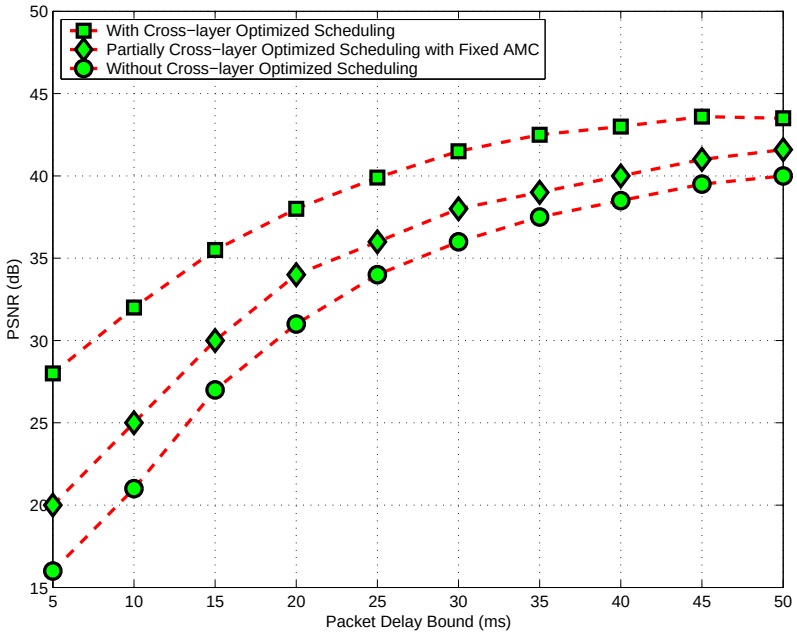


Figure 5.8 : Average Frame PSNR comparison between the existing P2P scheduling algorithm and the proposed distributed P2P scheduling algorithm.

We also present the evaluation of the number of P2P nodes on the overall video performance in Figure 5.10. As we can observe from this figure, the overall performance

increases with the increase of the nodes joining in the overlay networks. More important, the proposed scheduling algorithm also works well when the number of P2P network nodes decreases. This is possible because the proposed scheduling algorithm performs link adaptation by dynamically fine-tuning parameter values at different layers, so that the packet loss rate can be effectively decreased.

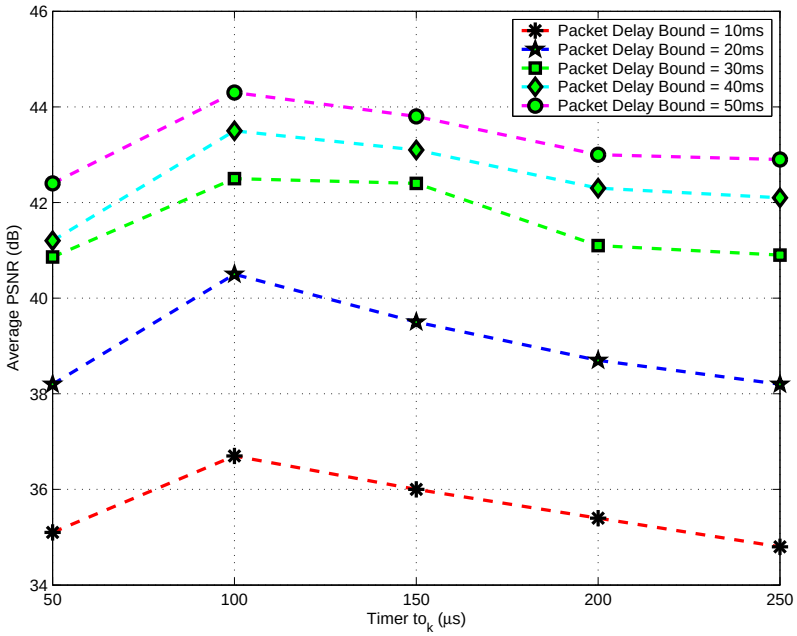


Figure 5.9 : Average Frame PSNR comparison between the existing P2P scheduling algorithm and the proposed distributed P2P scheduling algorithm with different timer to_κ values.

At last, we present the visual quality difference in Figure 5.11. In this figure, (a) is the video frame that uses the existing P2P scheduling algorithm, while (b) is the video frame when the proposed distributed P2P scheduling algorithm is used. It can be concluded that the existing system experiences more losses and thus more frequent use of error concealment, leading to degraded video quality. Our experiments also showed that this user-perceived performance enhancement can be even more significant when the playback deadline is more stringent or/and the channel quality is worse.

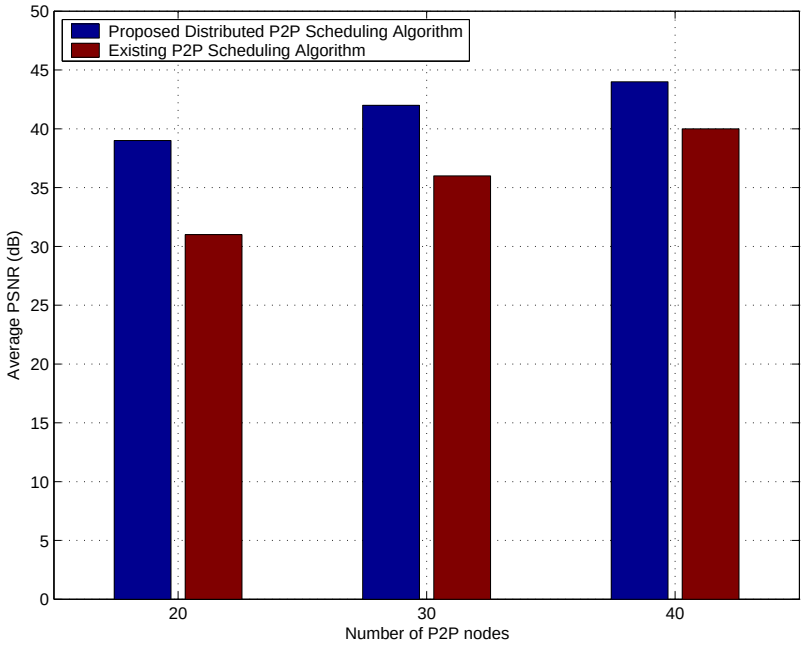


Figure 5.10 : Average Frame PSNR comparison between the existing P2P scheduling algorithm and the proposed distributed P2P scheduling algorithm with different numbers of P2P nodes.

Figure 5.11 : User-perceived video quality comparison between the existing scheduling algorithm and the proposed distributed P2P scheduling algorithm.

5.7 Summary

In this chapter, a distributed utility-based scheduling algorithm for interactive P2P video streaming applications over wireless networks has been proposed and studied. The present-

ed scheduling algorithm performs integrated cross-layer optimization, which is distributed to each peer node to determine the best system parameters residing in different network layers such as video codec, retransmission, network condition, and modulation and coding. This essential part of the scheduling algorithm has been formulated to minimize the total distortion constrained by the playback deadline for a given missing packet on the neighboring nodes. The computational complexity on each node has thus been effectively decreased through the distributed algorithm. Extensive experiments have been conducted for design verification, which have demonstrated that the proposed distributed P2P scheduling algorithm can achieve significant performance enhancement for video applications in P2P networks over the existing P2P scheduling algorithm.

Chapter 6

Context-aware Multimedia Communications

In this chapter, we extend the concept of dynamic adaptation of the wireless channel variation to context-aware services, where the context information includes channel variation, location, weather, user preference and so on. We also study a context-aware platform for wireless multimedia communications.

6.1 Introduction

With the development of mobile devices such as netbook computers, PDAs, smart phones etc., ubiquitous or pervasive systems have gained increasing popularity. Context-aware service has thus emerged as one of the most important fields of pervasive computing [77], which can adapt the system operations to the current context without explicit user intervention and thus increase the system usability and effectiveness by taking environmental context into account. With the development of computing technologies, the needed context data can also be retrieved in a variety of ways, such as applying sensors, network information, device status, user profiles and other external sources. When it comes to multimedia services for mobile devices, it is desirable that programs and services react specifically to their current location, time and other environment attributes and adapt their behaviors according to the changing context data. With intelligent services, customers can be better served.

Many definitions on context can be found in literature. One of the most generic definitions is: *Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves* [5]. Based on this definition, context data can be anything that is relevant to an application and its

set of users, as long as it can be used to characterize the situation of a participant in an interaction. For example, a user's location is often used to characterize the user's situation.

Context-aware services are concerned with the acquisition of context (e.g. using sensors to perceive a situation), the abstraction and understanding of context (e.g. matching a perceived sensory stimulus to a context), and the application behavior based on the recognized context (e.g. triggering actions based on context) [7]. As the user's activity and location are important for many applications, context-awareness has been focused more deeply in the research fields of location awareness and activity recognition in the past. However, with the development of computing technologies in recent years, more information can be integrated into context data and be used to broaden the scope of pervasive computing. Context-aware systems are thus becoming more and more complex. On the other hand, the demand on context-aware services has remained on the rise. Actually, context-awareness is regarded as an enabling technology for ubiquitous computing systems [78].

Gu *et al.* in [79] presented a Service-oriented Context-Aware Middleware (SOCAM) architecture for the building and prototyping of context-aware mobile services in an intelligent vehicle environment. Pessoa *et al.* in [80] discussed the suitability of using ontologies for modeling context information and presented a usage scenario where the system can send either a Short Message Service (SMS) or Multimedia Messaging Service (MMS) message to the tourist (the user), depending on whether he or she has a mobile device or a smart phone. In [81], an OSGi-based [82] infrastructure for context-aware multimedia services in a smart home environment was presented, which supports multimedia content filtering, recommendation, and adaptation according to the changing context. Further, in [83], the authors proposed a middleware for ad hoc networks that instantiates a new networking plane called the Information Plane (InP), which is a distributed entity to store and disseminate information concerning the network, its services and the environment, orchestrating the collaboration among cross-layer protocols, automatic management solutions and context-aware services. Thus, protocols and services may improve their operations by using

algorithms that take into account context, service and network information.

However, most of the previous studies on context-aware multimedia applications only focus on very limited context information such as the end equipment type. Context data that are specific and also essential to wireless video transmission such as varying channel quality, available energy on the end equipment and application Quality of Services (QoS) are not considered. These have posed a very strict limit upon the applicability of the presented systems. In this chapter, we propose a novel and general context-aware based adaptive wireless multimedia system that is divided into two stages. First, it can dynamically adapt to the user's location, time and weather context based on the user's profile to choose the interested videos for the end user. Then, more context data are integrated into the system, such as the end equipment type and its resource information, the varying network conditions and the application QoS requirements, to dynamically perform media adaptation so that the video quality can be highly improved for the end user.

6.2 Typical User Cases

The proposed design focuses on creating ontologies that are suitable for building pragmatic context-aware based wireless multimedia communications systems. Thus, the presented system can be used in many scenarios, for example, mobile tour guide for cities or museums, smart homes, campus visits, online learning, etc.

- City Tour Guide. Imagine a tourist is touring around a new city. He or she may want to be informed of the nearest tourist attractions of interest through video streaming on the available mobile client. The proposed context-aware based wireless multimedia communications system can provide this service to the user on the basis of the retrieved location data, the time and weather information, and the user's profile. Then, the end equipment context (e.g., equipment type, screen size, available energy), the wireless network conditions (e.g. wireless channel quality, network congestion), and the application QoS (e.g., video frame rate) are combined to dynamically perform

media adaptation to determine the optimal encoding parameters and the transmission schemes, so that the video quality displayed at the user's end equipment can be greatly improved. The test bench used in this chapter is based on this application scenario.

- Smart Homes. In this case, a user's preference for the multimedia service, such as the media subtitle, the audio volume, the video brightness/contrast and the time stamp, can be combined with the user's location (the specific room) to enable identical service environment when the user moves from one room to another, or even to his or her courtyard [84]. With a sharing platform, it enables different electronic devices to share multimedia between different products and brands, thus users can share digital information on PC, TV, set-top box, printer, stereo, mobile phones, PDA and DVD player through wireless networks. The proposed context-aware system can also be used to adapt the source and channeling coding of the multimedia to provide better video quality to the users based on the changes of habits, the usage of multimedia, and the context data of the electrical appliances such as the end equipment type, screen size, CPU, available energy etc.

6.3 System Modeling

The design objective of the proposed context-aware based wireless multimedia services can be generalized as the following problem – given a set of media options and context variables, how to use all the context data to dynamically choose the proper video streaming sources and ensure the best video transmission quality at the receiver side through wireless networks. To achieve this, different adaptation mechanisms can be combined.

- Structural adaptation. The video streaming content can be changed according to the user's location, time and weather context and the user's profile. For example, a user who is interested in performing arts may be recommended an arts museum on a rainy

day.

- Spatial adaptation. Based on the user's end equipment context, the spatial resolution of the video can be changed. For example, when the user switches from a laptop to a PDA, the screen size needs to be changed to adapt to the size of PDA to ensure high video quality.

- Quality adaptation. The video quality of each frame can be changed based on the varying channel quality and the network congestion status. For example, Adaptive Modulation and Coding (AMC) can be used to ensure high video quality by dynamically adapting to the varying channel conditions [4].

- Format adaptation. Based on the network condition and end equipment context, the encoder parameters can be dynamically changed to maintain high video quality. For example, in H.264 codec, with good wireless channel conditions, lower Quantization Step (QP) size can be used to improve the overall video quality.

As shown in Figure 6.1, the context-aware application is located on top of the architecture, and thus it can make use of the different levels of context data and adapt their behaviors according to the current context. In the figure, the context providers abstract contexts from different sources such as the client, the network, the external weather information and the server and then convert them to OWL [85] representation so that contexts can be shared and reused by other components. The context reasoning engine provides the context reasoning services, which includes inferring deduced contexts, resolving context conflicts and maintaining the consistency of the context knowledge base. Different inference rules can be specified and input into the reasoning engines. The context knowledge base provides the service that other components can query, add, modify or delete context knowledge in the context database. Furthermore, the media adaptation makes use of different levels of contexts and adapt the way they behave according to the current context data.

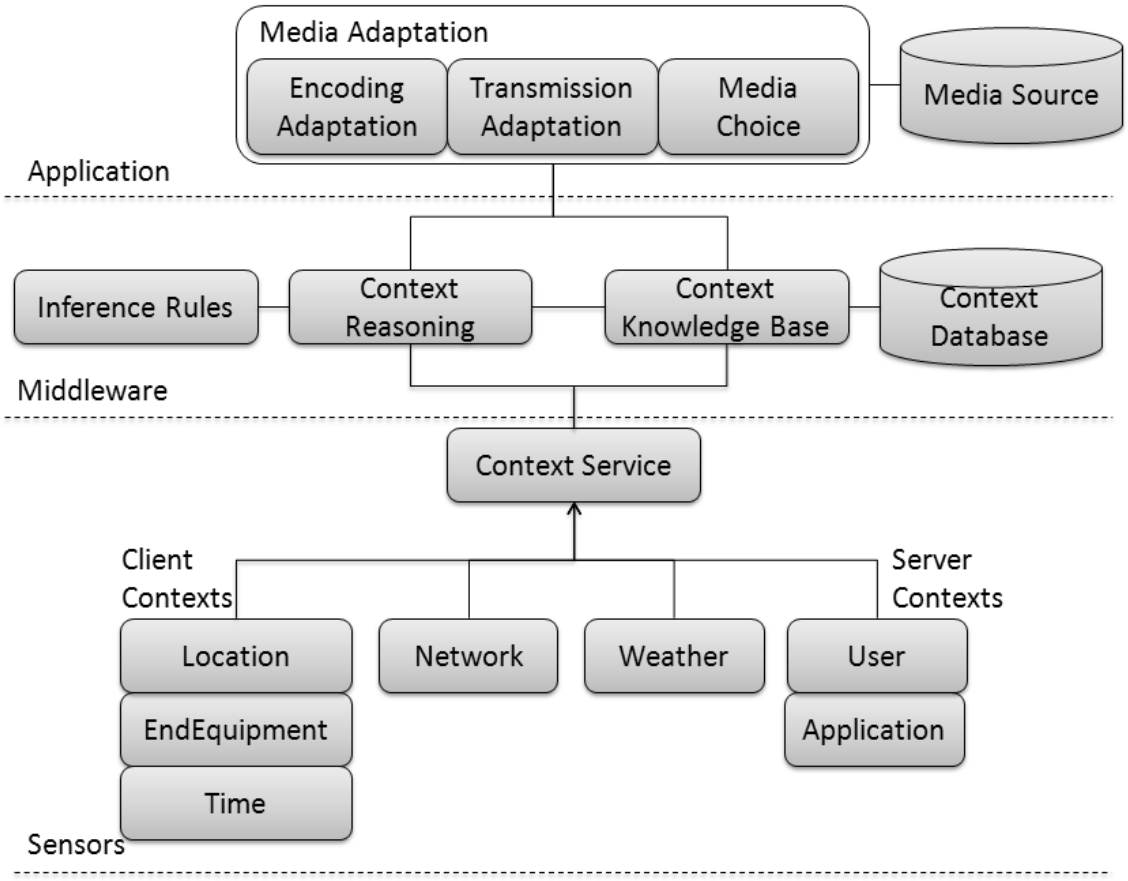


Figure 6.1 : The overall architecture and data flows of the proposed adaptive wireless multimedia communications with context-awareness.

6.4 Ontology-based Context Modeling

To make contextual data usable and sharable by applications, it is necessary to model sensor data values. Most current systems use their own methods to model context, making exchange of context and inter-operability between existing context-aware systems very difficult. To facilitate the development of extensible and inter-operable context-aware applications, it is essential to have a set of well-defined, uniform context models and protocols based on some principles for specifying any given context from any domain. The proposed ontology-based context modeling is illustrated in Figure 6.2.

From the figure, it should be noted that sensors include both physical and logical sensors.

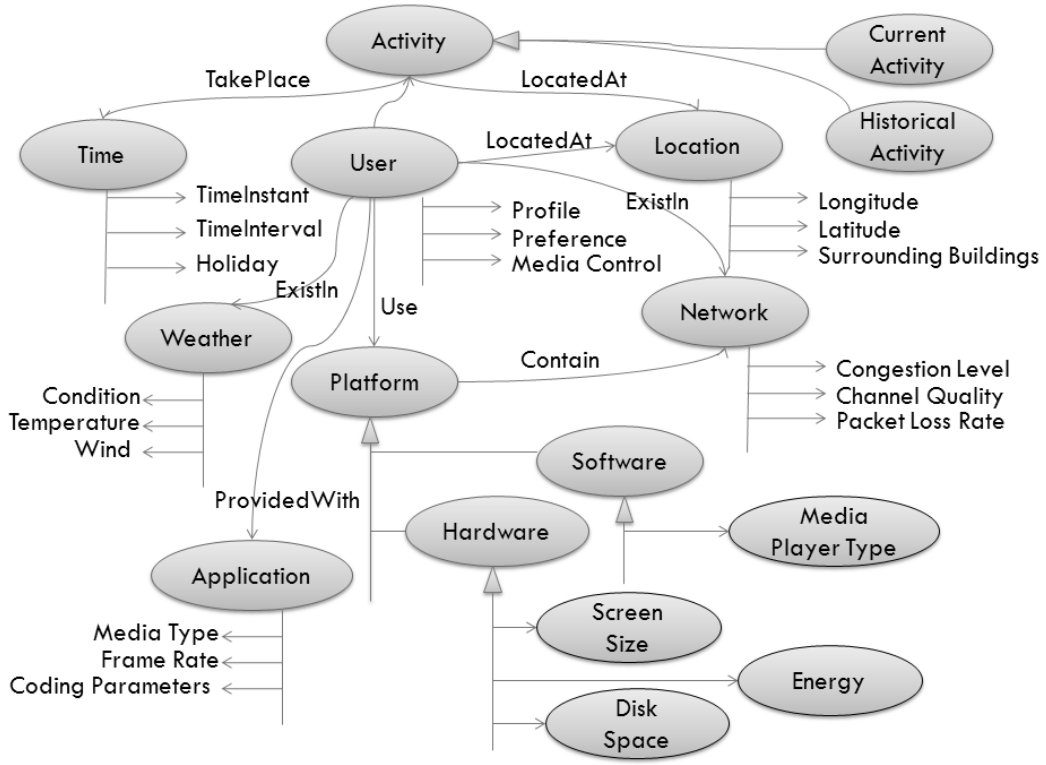


Figure 6.2 : The ontology-based context modeling of the proposed adaptive wireless multimedia communication system with context-awareness.

Logical sensors can make use of one or many context sources with additional information from databases to perform inference. For example, the user profile context, weather context, location context and time information can be combined to determine the appropriate video source.

As shown in Figure 6.2, we consider four major classes of context information in this chapter, which include contexts from the client side, the network, context data from the media server and external context sources such as the weather forecast. More context data can be easily added to the system. The client context class encompasses the end equipment context, the location context and the time context. The network context includes the information fed back from the network, representing the transmission status and network conditions. Further, the server context provides information regarding the user profile and

Table 6.1 : Context classes and properties of the proposed adaptive wireless multimedia communications system.

Context Type	Context Class	Context Subclasses and Properties
Client	End Equipment	End Equipment Type, Available Energy, CPU, Memory
Client	Location	Altitude, Longitude, Surrounding Buildings, Geospatial Information
Client	Time	Weekday, Weekend, Holiday, Specific Time
Network	Network	Channel Quality, Congestion Status, Transmission Delay
Server	User Profile	Preference, Historical Visits, Audio Volume
Server	Application	Encoding Parameters, Media Type, Frame Rate
External	Weather	Conditions, Temperature, Wind, Humidity

the multimedia application QoS requirements. Some of the possible classes and properties that can be integrated into the system are shown in Table 6.1.

Furthermore, context data are divided into static context information and dynamic context information. The static context information, such as end equipment type, hardly changes during the service usage. However, dynamic context is constantly changing, e.g. the channel quality in a wireless environment. It should be noted that location context is considered as static, although the user may be moving. This is because when the user is located within certain range of a location, the recommended videos remain the same. In this chapter, we handle the static and dynamic context data at different stages. First, the static context information such as location, end equipment, weather and time information are used to determine the streaming content, type and size. Then, the varying context data such

as wireless channel quality and network congestion status are used to dynamically perform source coding and channel coding to ensure highly-improved video streaming quality.

6.5 Context Reasoning and Middleaware Design

Different types of context data have different levels of confidence and reliability. For example, a GPS-based location sensor may have a 95% accuracy rate while an RFID-based location sensor only has an 80% accuracy rate. Thus, by reasoning context classification information based on the proposed ontology-based context model, context conflicts can be effectively detected and solved. Furthermore, we also define our own reasoning based on first order logic to achieve higher flexibility. The following are two examples of the reasonings at the two proposed stages used in our experiments.

The middleware components in our proposed design also act as a context provider, as they provides high-level context by interpreting low-level contexts. The context knowledge base, which contains context ontologies in a subdomain and their instances, provides an interface for other service components to query, add, delete or modify context knowledge. These instances may be specified by users in case of defined contexts or acquired from various context providers in case of sensed contexts. The context ontologies and their instances of defined contexts are pre-loaded into the context knowledge base during system initiation, while the instances of sensed contexts are loaded during runtime.

$$
\begin{aligned}
&LocatedAt(?u, PacificSt\&DodgeSt) \wedge \\
&EquipmentType(?u, PDA) \wedge \\
&Preferred(?u, Arts) \wedge \\
&WeatherCond(Weather, Rainy) \wedge \\
&Time(Timeinstant, 10:00am) \models \\
&StreamingSources(?u, ListOfArtsMuseums)
\end{aligned}
$$

$Network(Congestion, Good) \wedge$

$EquipmentType(?u, iPhone3GS) \wedge$

$AvailableEnergy(?u, Good) \wedge$

$ChannelQuality(SNR, Good) \wedge$

$Application(MediaType, YUV) \wedge$

$Application(FrameRate, 30 frames/second) \models$

$SmallQP \ and \ HighRateChannelCoding$

6.6 Experimental Analysis

As a test bench of the proposed adaptive multimedia communications streaming system, we consider the following experiments. As shown in Figure 6.3, we allow two users P_1 (with an iPhone 3GS as the mobile client) and P_2 (with a LG INCITE CT 810 as the mobile client) to walk from location A, through B and C, to D and stop at each location for several minutes, respectively. The entire area is covered by a WIFI-based wireless mesh network with a central controller in the wired network. We suppose P_1 prefers science and natural scenery, while P_2 prefers animals and entertainment activities. We then configure some videos for different tourist attractions for each stop as shown in Table 6.2. H.264 codec is used to encode these videos to provide streaming services. For media adaptation, we consider the context data of varying wireless channel quality (the fed-back SNR), monitored network congestion status, available energy at the end equipment, and the application video frame rate. To dynamically adapt to the changes of these context data, Quantization Step (QP) size and AMC mode [4] are changed on a frame-by-frame basis. The optimized QP value and AMC mode under the different contexts are achieved through a distortion-delay framework and are pre-loaded into the knowledge base on the basis of our previous research works [4, 86].

When P_1 and P_2 walk from A to D, we assume no matter where the users are, there is

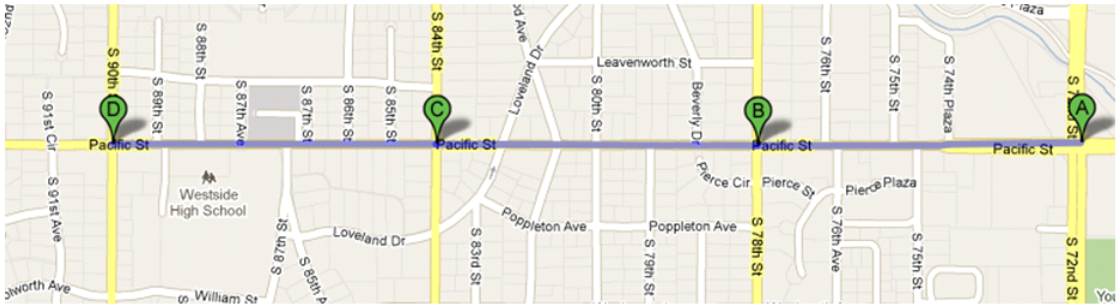


Figure 6.3 : Verification of location-awareness of the proposed adaptive wireless multimedia communications system, where the users walk from A to D (through B and C) and stop at each location for several minutes.

Table 6.2 : Configuration of videos of tourist attractions at different locations used in the experiments.

Locations	Tourist Attractions (Videos)
A	Elmwood Park, Arts Museum, Henry Doorly Zoo
B	Restaurant, History Museum, Memorial Park
C	Public Library, Farm, Air Show
D	Shopping Mall, Bar, Botanical Garden

at least one tourist attraction that will be recommended to the users. The user falls into the range of a location i ($i \in \{A, B, C, D\}$) as long as the distance between the current user and location i is smaller than that between him/her and any other location. As shown in Figure 6.4, the proposed system can choose the interested video source for the user according to the location context, the user profile, the time and weather context data. On a sunny day, when user P_1 is at around location A, Elmwood Park is recommended. When P_1 moves to the range of location C, the recommended tourist attraction will be changed to the air show. Similarly, when P_2 is at around location A, the zoo visit is recommended. When he/she moves to the range of location C, the recommendation is changed to the farm

visit.

(a) User P_1 at around location A is recommended Elmwood Park.

(b) User P_1 at around location C is recommended the Air Show.

(c) User P_2 at around location A is recommended the Henry Doorly Zoo.

(d) User P_2 at around location C is recommended the farm.

Figure 6.4 : The recommended tourist attractions of users with different mobile clients using the proposed adaptive wireless multimedia system with context-awareness.

We then demonstrate the improvement of video quality perceived by users with different mobile clients as shown in Figure 6.5. In this figure, we can observe that around 2dB-4dB performance improvement, in terms of PSNR [4], can be yielded by using the proposed system. Another observation is that the mobile client used by P_2 usually performs not as well as P_1's mobile client does, possibly due to the overall reduction in a good receiving signal resulting from cell phone differences. However, with the proposed context-aware based system, we can achieve even higher performance gain for user P_2.

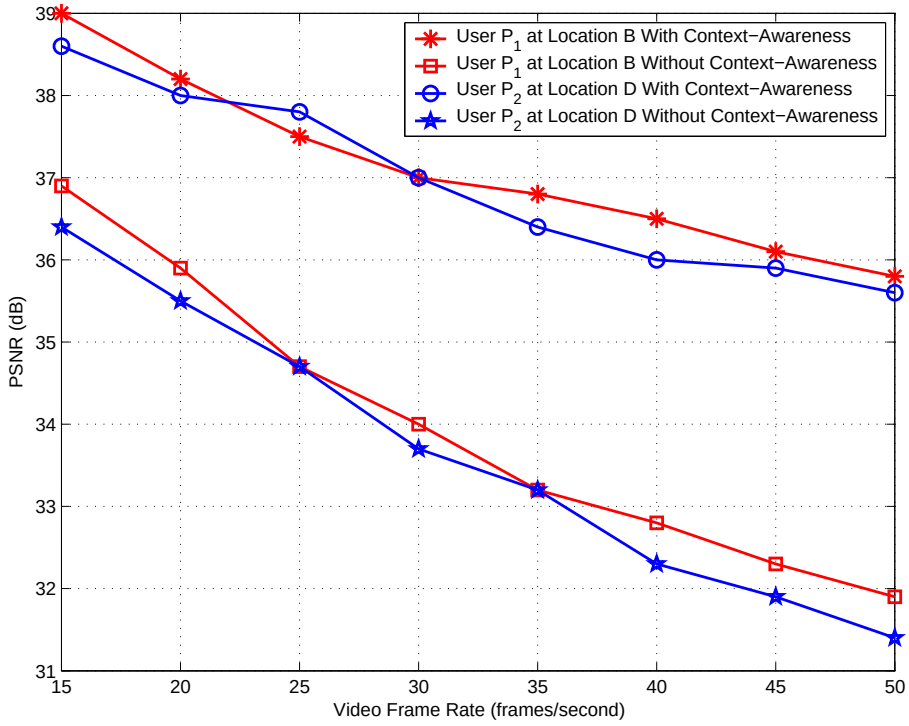


Figure 6.5 : The video quality improvement using the proposed adaptive wireless multimedia communications system with context-awareness for users P_1 and P_2 at locations B and D, respectively.

6.7 Summary

In this chapter, a context-aware adaptive wireless multimedia communications system using ontology-based models has been proposed, which can be used as an intelligent tour guide and in museum visits, smart homes, online learning, etc. The adopted context modeling and the overall system architecture have been discussed with some of the implementation details. Moreover, we have described a test bench based on the tour guide application that has proved the effectiveness and applicability of the proposed system.

Chapter 7

Conclusions

In this chapter, we summarize the contributions of this dissertation and present some potential research topics for the future.

7.1 Summary of Research Contributions

In this dissertation, we have investigated the current problems that are intrinsic in multimedia communications over wireless networks. We started with the discussion on the differences between the Internet-based multimedia applications and the wireless multimedia communications that affect the overall video performance. Then, with these specific problems, we have provided our solutions from both theoretical and practical perspectives to solve the issues, significantly improving the user-perceived video quality, against limited resources in wireless networks.

Specifically, we have proposed a novel joint source and distributed error control algorithm for video streaming services in wireless multi-hop networks to achieve globally optimal joint source and channel coding decisions for video transmission over wireless multi-hop networks. Then, within the context of an e-healthcare system, we have presented a content-aware communications algorithm that can accurately identify the region of interest (ROI) and efficiently utilize the limited network resources based on the achieved ROI information to significantly improve the overall user-perceived video quality. Further, we analyze how cross-layer optimization can be used in wireless networks to improve video quality by studying the scheduling algorithm for the P2P multimedia communication networks. Finally, by integrating more rapidly and/or slowly changed context information, we have provided our research on context-aware multimedia communications.

It is worth mentioning that all these techniques that have been discussed in this dissertation to improve video performance are general methodologies and can be easily extended to other application scenarios. For example, content-aware communications have been used in [87,88]. The cross-layer optimization techniques have been successfully used for cognitive radio networks [89], 3GPP Long Term Evolution (LTE) [90], and TCP Friendly Rate Control (TFRC) [91]. Further, these techniques can be combined together to further improve the video performance in wireless networks.

7.2 Future Work

7.2.1 Game Theory

Game theory is a discipline aimed at modeling situations in which decision-makers have to make specific actions that have mutual, and possibly conflicting, consequences. It has been primarily used in economics to model competition between companies. For example, should a given company enter a new market, considering that its competitors could make similar (or different) moves [92, 93]?

Game theory has also been applied to other areas, including politics and biology. Not surprisingly, it can and has also been applied to networking, in most cases to solve routing and resource allocation problems in a competitive environment. In recent years, it has been applied by an increasing number of researchers to resolve traditional issues in wireless networks. The common scenario is that the decision makers in the game are rational users or networks operators who control their communication devices. These devices have to cope with a limited transmission resource (i.e., the radio spectrum) that imposes a conflict of interests. In an attempt to resolve this conflict, they can make certain moves such as transmitting now or later, changing their transmission channel, or adapting their transmission rate.

During the last few years, some research has been carried out to extend the application

of game theory to wireless multimedia communications. The research result is preliminary, but the initiation is exciting. Felegyhazi *et al.* in [94] summarized how game theory can be used to model the radio communication channel. By leveraging four simple running examples, the authors in that paper have introduced the most fundamental concepts of non-cooperative game theory.

Wireless multimedia communications, especially in large-scale networks, is a complicated issue. Using game theory to solve the issues facing wireless multimedia communication will shed new insight on wireless technologies and streaming applications.

7.2.2 Energy-aware Multimedia Communications

With the increasing popularity of mobile computing devices, one of the most important issues that affects the mobile multimedia communications is energy consumption. Mobile devices, especially the nodes in wireless sensor networks, are usually powered by batteries and have very limited energy supply [95]. In most cases, these batteries are not rechargeable due to tough environments. On the other hand, video decoding and delay-sensitive playback cost a lot of energy. Thus, one of the potential research topics in this area is to provide the best tradeoff between the user-perceived quality and the lifetime of the given energy. Cross-layer optimization can be used in this respect [96].

7.2.3 GENI

The Global Environment for Network Innovations (GENI) is a facility concept being explored by the U.S. computing community with support from the National Science Foundation (NSF). The goal of GENI is to enhance experimental research in networking and distributed systems, and to accelerate the transition of this research into products and services that will improve the economic competitiveness of the United States. It is expected that the research performed by GENI will lead to capabilities beyond the Internet as we know it today [97].

GENI planning efforts are organized around several focus areas, including facility architecture, the backbone network, distributed services, wireless/mobile/sensor subnetworks, and research coordination amongst these.

Since the research on GENI is still at the initial stage, it is desirable to run some wireless multimedia applications and perform the cross-layer based optimization research based on the GENI concept.

7.2.4 Intelligent Agent

An intelligent agent (IA) is an autonomous entity which observes and acts upon an environment and directs its activity towards achieving goals. Intelligent agents may also learn or use knowledge to achieve their goals. They may be very simple or very complex: a reflex machine such as a thermostat is an intelligent agent, as is a human being, as is a community of human beings working together towards a goal [98].

By using the intelligent agent in wireless P2P networks, each node can learn the available resources and the history of its neighboring nodes. This knowledge can then be used for P2P scheduling, routing or even video transcoding. In this way, the available resources can be better managed and allocated, and thus the overall performance of the whole system can be improved.

7.2.5 Cognitive Radio Networks

Cognitive radio is an intelligent wireless communication system that is aware of its surrounding environment and uses the methodology of understanding-by-building to learn from the environment. Then, its internal states are adapted to respond to statistical variations of the incoming RF stimuli by changing certain operating parameters in real-time. Here, there are two primary objectives: 1) to provide highly reliable communications whenever and wherever needed; and 2) to achieve efficient utilization of the radio spectrum [99].

In cognitive radio networks, only primary users are authorized to use the radio spectrum.

Thus, secondary users have to search the idle channels to use at the beginning of every slot by performing channel sensing. Based on the sensing outcomes, secondary users will decide whether or not to access the sensed channels. It has been reported that some frequency bands in the radio spectrum are largely unused, while some are heavily used. In particular, while a frequency band is assigned to a primary wireless system/service at a particular time and location, the same frequency band is unused by this wireless system/service in other times and locations. This results in spectrum holes (a.k.a spectrum opportunities) [100]. Therefore, by allowing secondary users to utilize these spectrum holes, spectrum utilization can be improved substantially .

To achieve the best user-perceived video quality at the receiver side for secondary users of real-time wireless video transmission over cognitive radio networks, cross-layer optimization can be used to provide a quality-driven cross-layer system for joint optimization of system parameters residing in the entire network protocol stack. Thus, time variations of primary network usage and wireless channels can be modeled, based on which the encoder behavior, cognitive MAC scheduling, transmission, and modulation and coding can be jointly optimized for secondary users in a systematic way under a distortion-delay framework for the best video quality perceived by secondary users [101].

Bibliography

[1] I. Richardson, "White Paper: An Overview of H.264 Advanced Video Coding," *Available from http://www.vcodex.com/files/H.264_overview.pdf*, 2007.

[2] A. Argyriou, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264—ISO/IEC 14496-10 AVC)," *Elsevier Signal Processing: Image Communication*, pp. Available from ftp://ftp.imtc–files.org/jvt–experts/2003–03–Pattaya/JVT–G50r1.zip, May 2003.

[3] "Video Processing and Communications," *IEEE Communication Magazine*, vol. 42, no. 10, pp. 74–80, Oct. 2004.

[4] H. Luo, S. Ci, D. Wu, and H. Tang, "End-to-end optimized TCP-friendly rate control for real-time video streaming over wireless multi-hop networks," *Journal of Visual Communication and Image Representation (JVCI)*, vol. 21, no. 2, pp. 98–106, Feb. 2010.

[5] K. Dey and G. Abowd, "Towards a better understanding of context and context-awareness, workshop on the what, who, where, when, and how of context- awareness," *Computer Networks*, vol. 52, pp. 2961–2974, Jul. 2008.

[6] H. Luo, S. Ci, D. Wu, and A. Argyriou, "Joint source coding and network-supported distributed error control for video streaming in wireless multi-hop networks," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1362–1373, Nov. 2009.

[7] A. Schmidt, "Ubiquitous computing - computing in context," *PhD dissertation, Lancaster University*, pp. 2961–2974, 2003.

[8] R. Zhang, S. Regunathan, and K. Rose, "Video Coding with Optimal Inter/Intra-Mode Switching for Packet Loss Resilience," vol. 18, no. 6, pp. 966–976, Jul. 2003.

[9] M. S. Alouini and A. J. Goldsmith, "Adaptive Modulation over Nakagami Fading Channels," *Kluwer J. Wireless Communications*, vol. 13, pp. 119–143, May 2000.

[10] A. Doufexi, S. Armour, M. Butler, A. Nix, D. Bull, J. McGeehan, and P. Karlsson, "A Comparison of the HIPERLAN/2 and IEEE 802.11a Wireless LAN Standards," *IEEE Communication Magazine*, vol. 40, pp. 172–180, May 2002.

[11] "Advanced video coding for generic audiovisual services," ITU-T Rec. H.264/ISO/IEC 14496-10(AVC), Jul. 2005.

[12] D. Wu, T. Hou, W. Zhu, H.-J. Lee, T. Chiang, Y.-Q. Zhang, and H. J. Chao, "On end-to-end architecture for transporting MPEG-4 video over the Internet," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 923–941, Sep. 2000.

[13] G. Cote, 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, no. 6, pp. 952–965, Jun. 2000.

[14] A. Argyriou, "Real-time and rate-distortion optimized video streaming with TCP," *Elsevier Signal Processing: Image Communication*, vol. 22, no. 4, pp. 374–388, Apr. 2007.

[15] Z. He, J. Cai, and C. W. Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 511–523, Jun. 2002.

[16] Y. Zhang, W. Gao, Y. Lu, Q. Huang, and D. Zhao, "Joint source-channel rate-distortion optimization for h.264 video coding over error-prone networks," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 445–454, Apr. 2007.

[17] H. Luo, D. Wu, S. Ci, A. Argyriou, and H. Wang, "Quality-Driven TCP Friendly Rate Control for Real-Time Video Streaming," *IEEE GLOBECOM*, Dec. 2008.

[18] D. Wu, S. Ci, and H. Wang, "Cross-layer optimization for video summary transmission over wireless networks," *IEEE J. Select. Areas Commun.*, vol. 25, no. 4, pp. 841–850, May 2007.

[19] ——, "Cross-Layer Optimization for Packetized Video Communication over Wireless Mesh Networks," *IEEE ICC*, May 2008.

[20] (2002) IEEE Standard 802.16 Working Group, IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems.

[21] (2004) 3GPP TR 25.848 V4.0.0, Physical Layer Aspects of UTRA High Speed Downlink Packet Access (release 4).

[22] M. van der Schaar and P. Chou, *Multimedia over IP and Wireless Networks.* Academic Press, 2007.

[23] A. Argyriou, "Distributed resource allocation for network-supported FGS video streaming," *Packet Video Workshop*, Nov. 2007.

[24] J. Yan, K. Katrinis, M. May, and B. Plattner, "Media- and TCP-friendly congestion control for scalable video streams," *IEEE Trans. Multimedia*, vol. 8, no. 2, Apr. 2006.

[25] M. Chen and A. Zakhor, "Transmission protocols for streaming video over wireless," *IEEE International Conference on Image Processing (ICIP)*, 2004.

[26] L. Gao, Z.-L. Zhang, and D. Towsley, "Proxy-assisted techniques for delivering continuous multimedia streams," *IEEE/ACM Transactions on Networking*, vol. 11, no. 6, pp. 884–894, Dec. 2003.

[27] J. Chakareski and P. Chou, "Radio edge: Rate-distortion optimized proxy-driven streaming from the network edge," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1302–1312, Dec. 2006.

[28] G. Cheung and A. Zakhor, "Bit allocation for joint source/channel coding of scalable video," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 340–356, Mar. 2000.

[29] L. Kondi and A. Katsaggelos, "Joint source-channel coding for scalable video using models of rate-distortion functions," *ICASSP*, 2001.

[30] F. Zhai, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos, "Rate-distortion optimized hybrid error control for real-time packetized video transmission," *IEEE Trans. Image Processing*, vol. 15, no. 1, pp. 40–53, Jan. 2006.

[31] J. P. M. Handley, S. Floyd and J. Widmer, "TCP friendly rate control (TFRC): Protocol specification," *RFC 3448*, Jan. 2003.

[32] Q. Zhang, Z. Ji, W. Zhu, and Y.-Q. Zhang, "Power-Minimized Bit Allocation for Video Communication over Wireless Channels," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 398–410, Jun. 2002.

[33] M. Westerlund and S. Wenger, "RTP Topologies," *draft-ietf-avt-topologies-05.txt*, Jul. 2007.

[34] N. Thomos and P. Frossard, "Collaborative video streaming with Raptor network coding," *IEEE International Conference on Multimedia and Expo (ICME)*, 2008.

[35] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *Microsoft Research Technical Report MSR-TR-2001-35*, 2001.

[36] D. Bertseka and R. Gallager, *Data Networks*. Prentice-Hall, Upper Saddle River, NJ, 1987.

[37] O. B. Akan and I. F. Akyildiz, "ARC: The Analytical Rate Control Scheme for Real-Time Traffic in Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 634–644, Aug. 2004.

[38] A. Mukherjee, "On the dynamics and significance of low frequency components of Internet load," *University of Pennsylvania, Tech. Rep. MS-CIS-92-83*, 1992.

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

[40] A. Mukherjee, "JVT reference software," *Available from http://bs.hhi.de/ suehring/tml/download/.*

[41] Y. Andreopoulos, N. Mastronade, and M. van der Schaar, "Cross-layer optimized video streaming over wireless multi-hop mesh networks," *IEEE J. Select. Areas Commun.*, vol. 24, no. 11, pp. 2104–2115, Nov. 2006.

[42] S. Kompella, S. Mao, Y. Hou, and H. Sherali, "Cross-layer optimized multipath routing for video communications in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 25, no. 4, pp. 831–840, May 2007.

[43] F. Hartanto and H. R. Sirisena, "Hybrid error control mechanism for video transmission in the wireless IP networks," *IEEE Tenth Workshop on Local and Metropolitan Area Networks (LANMAN)*, 1999.

[44] H. Seferoglu, Y. Altunbasak, O. Gurbuz, and O. Ercetin, "Rate distortion optimized joint ARQ-FEC scheme for real-time wireless multimedia," *ICIP*, 2005.

[45] F. Zhai, "Optimal cross-layer resource allocation for real-time video transmission over packet lossy networks," *Ph.D. dissertation, Northwestern University*, 2004.

[46] Y. Shan, I. Bajic, S. Kalyanaraman, and J. Woods, "Overlay multi-hop FEC scheme for video streaming over peer-to-peer networks," *International Conference on Image Processing (ICIP)*, Oct. 2004.

[47] A. C. Begen and Y. Altunbasak, "An adaptive media-aware retransmission timeout estimation method for low-delay packet video," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 332–347, Feb. 2007.

[48] D. Nguyen, T. Nguyen, and X. Yang, "Multimedia Wireless Transmission with Network Coding," *Proc. of Packet Video*, Nov. 2007.

[49] M. Piccardi, "Background subtraction techniques: a review," *Systems, Man and Cybernetics*, vol. 4, pp. 3099 – 3104, Oct 2004.

[50] C. Stauffer and Eric, "Learning patterns of activity using real-time tracking," *IEEE Trans. Comm.*, vol. 46, pp. 595–602, May 1998.

[51] Q. Jackson and D. Landgrebe, "Adaptive Bayesian Contextual Classification Based on Markov Random Fields," *IEEE Trans. Geosci. Remote Sensing*, vol. 11, pp. 2454–2463, Nov. 2002.

[52] R. Kinderman and J. L. Snell, "Markov random fields and their applications," *Amer. Math. Soc.*, vol. 1, pp. 1–142, 1980.

[53] D. Chai and K. N. Ngan, "Face segmentation using skin-color map in videophone applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 551–564, Jun. 1999.

[54] D. Wu, H. Luo, S. Ci, H. Wang, and K. Katsaggelos, "Quality-Driven Optimization for Content-Aware Real-Time Video Streaming in Wireless Mesh Networks," *IEEE GLOBECOM*, pp. 1–5, Dec. 2008.

[55] E. Setton and B. Girod, *Peer-to-Peer Video Streaming.* Springer Science+Business Media, LLC, 2007.

[56] S. Petrovic and P. Brown, "Fluid Model for eMule File Sharing System," *Universal Multiservice Networks*, pp. 273–282, Feb. 2007.

[57] S. Baset and H. G. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol," *IEEE INFOCOM*, pp. 1–11, Apr. 2006.

[58] X. Zhang, J. Liu, B. Li, and T. Yum, "CoolStreaming/DONet: A Data-Driven Overlay Network for Efficient Live Media Streaming," *IEEE INFOCOM*, vol. 3, pp. 2102–2111, Mar. 2005.

[59] L. Zhao, J. Luo, M. Zhang, and S. Yang, "Gridmedia: A practical peer-to-peer based live video streaming system," *Multimedai Signla Processing*, pp. 1–4, Oct. 2005.

[60] J. Jia, C. Li, and C. Chen, "Characterizing PPStream across Internet," *Network and Parallel Computing Workshops*, pp. 413–418, Sept. 2007.

[61] "LiveStation Website," *http://www.livestation.com.*

[62] F. Delmastro, "From Pastry to CrossROAD: CROSS-layer Ring Overlay for AD hoc networks," *IEEE PerCom*, pp. 60–64, Mar. 2005.

[63] M. Conti, J. Crowcroft, F. Delmastro, and A. Passarella, "P2P support for Group-Communication Applications: a Cross-Layer Approach for MANET Environments," *IEEE INFOCOM*, pp. 23–29, Apr. 2006.

[64] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," *IFIP/ACM Middleware*, Nov. 2001.

[65] M. Barbera, A. G. Busa, A. Lombardo, and G. Schembra, "CLAPS: A Cross-Layer Analysis Platform for P2P Video Streaming," *IEEE ICC*, pp. 50–56, Jun. 2007.

[66] P. Si, F. R. Yu, H. Ji, and V. C. M. Leung, "Distributed sender scheduling for multimedia transmission in wireless mobile peer-to-peer networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 9, pp. 4594–4603, Sep. 2009.

[67] N. Mastronarde, D. S. Turaga, and M. van der Schaar, "Collaborative resource exchanges for peer-to-peer video streaming over wireless mesh networks," *IEEE J. Select. Areas Commun.*, vol. 25, no. 1, pp. 108–118, Jan. 2007.

[68] J. Liu, S. Rao, B. Li, and H. Zhang, "Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast," *Proceedings of the IEEE*, vol. 96, pp. 11–24, Jan. 2008.

[69] A. J. Ganesh, A. Kermarrec, and L. Massoulie, "Peer-to-Peer Membership Management for Gossip-Based Protocols," *IEEE Trans. Comput.*, vol. 52, no. 2, pp. 139–149, Feb. 2003.

[70] I. Ahmad, X. Wei, Y. Sun, and Y. Zhang, "Video transcoding: An overview of various techniques and research issues," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 793–804, Oct. 2005.

[71] Q. Liu, S. Zhou, and G. Giannakis, "Cross-Layer combining of adaptive Modulation and coding with truncated ARQ over wireless links," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1746–1755, Sept. 2004.

[72] H. Luo, S. Ci, and D. Wu, "A cross-layer optimized distributed scheduling algorithm for peer-to-peer video streaming over multi-hop wireless mesh networks," *IEEE SECON*, pp. 1–9, Jun. 2009.

[73] A. Abdrabou and W. Zhuang, "Service time approximation in IEEE 802.11 single-hop ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 1, pp. 305–313, Jan. 2008.

[74] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. Image Processing*, vol. 2, pp. 160–175, Apr. 1993.

[75] Y. Andreopouls, N. Mastronarde, and M. van der Schaar, "Cross-layer optimized video streaming over wireless multihop mesh networks," *IEEE J. Select. Areas Commun.*, vol. 24, no. 11, pp. 2104–2115, Nov. 2006.

[76] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-riding and whitewashing in peer-to-peer systems," *Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, pp. 228–236, 2004.

[77] M. Weiser, "The computer for the twenty-first century," *Scientific American*, pp. 92–99, 1991.

[78] A. Schmidt, M. Beigl, and H. Gellersen, "There is more to context than location," *Computers & Graphics Journal*, vol. 23, no. 6, pp. 893–902, Dec. 1999.

[79] T. Gu, H. K. Pung, and D. Q. Zhang, "A middleware for builing context-aware mobile services," *In Proceedings of IEEE Vehicular Technology Conference (VTC)*, 2004.

[80] R. M. Pessoa, C. Z. Calvi, J. G. P. Filho, C. R. G. de Farias, and R. Neisse, *Dependable and Adaptable Networks and Services.* Springer Berlin / Heidelberg, 2007.

[81] Z. Yu, X. Zhou, Z. Yu, D. Zhang, and C. Chin, "An osgi-based infrastructure for context-aware multimedia services," *IEEE Communications Magazine*, vol. 44, no. 10, pp. 136–142, 2006.

[82] "OSGi consortium website," *http://www.osgi.org/.*

[83] D. F. Macedo, A. L. dos Santos, J. M. S. Nogueira, and G. Pujolle, "A distributed information repository for automatic context-aware manets," *IEEE Trans. on Networks and Service Management*, vol. 6, no. 1, pp. 45–55, Mar. 2009.

[84] C. Lai and Y. Huang, "Context-aware multimedia streaming service for smart home," *Proceedings of the International Conference on Mobile Technology, Applications, and Systems*, 2008.

[85] "OWL Web Ontology Language Overview," *http://www.w3.org/TR/owl-features*, 2004.

[86] H. Luo, A. Argyriou, D. Wu, and S. Ci, "End-to-end optimized TCP-friendly rate control for real-time video streaming over wireless multi-hop networks," *IEEE Trans. Multimedia*, vol. 11, no. 7, pp. 1362–1372, Nov. 2009.

[87] D. Wu, H. Luo, S. Ci, H. Wang, and A. Katsaggelos, "Quality-Driven Optimization for Content-Aware Real-Time Video Streaming in Wireless Mesh Networks," in *Proc. IEEE GLOBECOM*, New Orleans, LA, Dec. 2008.

[88] H. Luo, S. Ci, D. Wu, K. Siu, and N. Stergiou, "Content-aware Wireless Real-time Streaming for Marker-less Human Motion Tracking," *IEEE Wireless Commun. Mag.*, vol. 17, no. 1, pp. 37–43, Feb. 2010.

[89] H. Luo, S. Ci, D. Wu, and H. Tang, "Cross-layer design for real-time video transmission in cognitive wireless networks," in *Proc. IEEE INFOCOM-2010 Workshop on Cognitive Wireless Communications and Networking*, Mar. 2010.

[90] H. Luo, S. Ci, D. Wu, K. Siu, and N. Stergiou, "Quality-Driven Cross-Layer Optimized Video Delivery over LTE," *IEEE Commun. Mag.*, vol. 48, no. 2, pp. 102–109, Feb. 2010.

[91] H. Luo, D. Wu, and S. Ci, "Cross-layer optimized end-to-end tcp friendly rate control for real-time video streaming," *ISpecial Issue of Network Technologies for Emerging Broadband Multimedia Services of Journal of Visual Communication and Image (JVCI)*, 2009.

[92] D. Fudenberg and J. Tirole, *Game Theory.* MIT Press, 1991.

[93] R. Gibbons, *A Primer in Game Theory.* Prentice Hall, 1992.

[94] M. Felegyhazi and J.-P. Hubaux, "Game theory in wireless networks: A tutorial," *EPFL technical report, LCA-REPORT-2006-002*, Feb. 2006.

[95] Y. Yao, Y. Fan, and H. Luo, "A distributed relay node placement strategy based on balanced network lifetime for wireless sensor networks," pp. 306–310, Jun. 2010.

[96] H. Luo, S. Ci, D. Wu, and H. Tang, "Adaptive wireless multimedia communications with context-awareness using ontology-based models," *IEEE GLOBECOM*, pp. 1–5, 2010.

[97] "GENI website:," *http://www.geni.net/.*

[98] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. pper Saddle River, New Jersey: Prentice Hall, 2003.

[99] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 201–220, Feb 2005.

[100] R. Tandra, S. Mishra, and A. Sahai, "What is a Spectrum Hole and What Does it Take to Recognize One?" *Proceedings of the IEEE*, Apr. 2009.

[101] H. Luo, S. Ci, D. Wu, and H. Tang, "Cross-layer design for real-time video transmission in cognitive wireless networks," *IEEE INFOCOM 2010 Workshop on Cognitive Wireless Communications and Networking*, Mar. 2010.