12-2-2010

Cross-Layer Optimized Wireless Multimedia Networking

Dalei Wu
University of Nebraska-Lincoln, dwu@huskers.unl.edu

Follow this and additional works at: http://digitalcommons.unl.edu/ceendiss
Part of the Digital Communications and Networking Commons

http://digitalcommons.unl.edu/ceendiss/6

This Article is brought to you for free and open access by the Computer & Electronics Engineering, Department of at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in Dissertations & Student Research in Computer Electronics & Engineering by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.
CROSS-LAYER OPTIMIZED WIRELESS MULTIMEDIA NETWORKING

by

Dalei Wu

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

November, 2010
CROSS-LAYER OPTIMIZED WIRELESS MULTIMEDIA NETWORKING

Dalei Wu, Ph.D.

University of Nebraska, 2010

Advisor: Song Ci

Multimedia communications, especially real-time video communications, is expected to be the major application of the next-generation wireless networks. However, bringing delay-sensitive and loss-tolerant multimedia services based on the current wireless Internet is a very challenging task. In this dissertation, we address cross-layer optimized wireless multimedia networking from both theoretical and practical perspectives.

In the first part of the dissertation, we propose cross-layer optimization frameworks for real-time video communications over wireless networks, where the expected received video quality is adopted as the objective function. With the user-centric objective function, we first study content-aware video communications in single-hop wireless networks by exploring the transmission of video summaries. Then, we investigate the routing issue for real-time video streaming over multi-hop wireless networks. Lastly, we study the performance of video summary transmission over cooperative wireless networks to exploit the spatial diversity of cooperative communications. Extensive theoretical and experimental results demonstrate that significant performance gains are obtained by our solutions.

In the second part of the dissertation, we theoretically study the methodology of cross-layer design and optimization. Despite rich literature in cross-layer design and optimization schemes, most current research on cross-layer design has been carried out in various piece-meal approaches and lacks a methodological foundation to gain in-depth understanding of complex cross-layer behaviors. We focus on the quantitative analysis of the interactions among design variables towards to the design objective. The interaction measure is calculated based on the non-additive measure theory with network observation data. We conduct a case study on cross-layer optimized wireless multimedia communications to illustrate the major cross-layer design tradeoffs and validate the proposed theoretical framework. The proposed framework can significantly enhance our capability for cross-layer behavior characterization and provide insights for future design.
Acknowledgments

First of all, I would like to express my genuine gratitude to my advisor, Prof. Song Ci, for his excellent supervision and encouraging discussions. His extensive expertise and experience combined with great enthusiasm in research have made working on this dissertation a wonderful experience. I also value his influences on my vision of my professional and personal developments.

I am grateful to my committee members: Prof. Hamid Sharif, Prof. Yaoqing (Lamar) Yang, Prof. Lisong Xu, and Prof. Hai-Feng Guo. I would like to thank them for serving in my committee and their critique on this dissertation. My special thanks are due to Prof. Hamid Sharif for his kind encouragement and invaluable advice during my doctoral study.

I am indebted to Dr. Haohong Wang, Prof. Aggelos Katsaggelos, Prof. Hai-Feng Guo, Prof. Zhu Han, and Dr. Guan-Ming Su. They provided very constructive comments and discussions in the development of this research.

I would like to thank all members in the Intelligent Ubiquitous Computing Lab for useful discussions. I also owe my thanks to all colleagues at Peter Kiewit Institute, past and present, for creating a friendly, pleasant, and stimulating atmosphere over these years.

Finally, this dissertation is dedicated to my wife and parents for their love, sacrifice, and support.
To my wife Fengjie and my parents
for their love and support
Contents

Abstract ii
Acknowledgments iv
List of Figures ix
List of Tables xiii

1 Introduction 1
1.1 Motivation ................................................... 1
1.2 Research Scope ............................................. 4
1.3 Dissertation Outline ....................................... 5

2 Cross-Layer Optimized Video Summary Transmission over Single-Hop Wireless Networks 8
2.1 Introduction ................................................... 8
2.2 Problem Statement ......................................... 11
   2.2.1 Background ........................................... 11
   2.2.2 Problem Formulations ................................. 13
2.3 System Model ............................................... 16
2.4 Link Adaptation ............................................. 19
2.5 Algorithm of the Controller ............................... 23
   2.5.1 Optimal Solution ......................................... 23
   2.5.2 Implementation Considerations ....................... 25
2.6 Experimental Results ..................................... 26
2.7 Summary ..................................................... 31

3 Cross-Layer Optimized Routing for Video Streaming over Multi-Hop Wireless Networks 32
3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
3.2 Problem Description . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
  3.2.1 Proposed System Model . . . . . . . . . . . . . . . . . . . . . . . . . 36
  3.2.2 Problem Formulation . . . . . . . . . . . . . . . . . . . . . . . . . . 41
3.3 Solution Procedures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
3.4 Proposed Routing Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . 44
  3.4.1 Routing Metric: Packet-Delay-Deadline-Aware Expected Distortion 45
  3.4.2 Optimal Path Selection for Individual Packets . . . . . . . . . . . . . 50
3.5 Convergence and Complexity Discussion of the Proposed Framework . . . 56
  3.5.1 Convergence of the Proposed Routing Algorithm . . . . . . . . . . . . 56
  3.5.2 Computational Complexity of the Proposed Framework . . . . . . . 58
3.6 Experimental Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
3.7 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68

4 Cross-Layer Optimized Video Summary Transmission over Cooperative Wireless Networks 69
  4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69
  4.2 System Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72
    4.2.1 Video Summary Distortion Model . . . . . . . . . . . . . . . . . . . 72
    4.2.2 Cooperative Video Summary Transmission Model . . . . . . . . . . 73
  4.3 The Proposed Scheme . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
    4.3.1 The Proposed DPF Scheme . . . . . . . . . . . . . . . . . . . . . . . 75
    4.3.2 Problem Formulation . . . . . . . . . . . . . . . . . . . . . . . . . . 76
  4.4 Solution Procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
    4.4.1 Optimal Solution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
    4.4.2 Implementation Considerations . . . . . . . . . . . . . . . . . . . . . 80
  4.5 Experimental Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
  4.6 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 84

5 Interaction Measure and Sensitivity Analysis in Cross-Layer Design 85
  5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
5.2 The Proposed Theoretical Framework for Quantitative Interaction Measure

5.2.1 Problem Description .................................................. 88
5.2.2 Interaction Measure .................................................... 91
5.2.3 The Proposed Non-Linear Regression Model for the Interaction
Measure ................................................................. 94

5.3 The Proposed Theoretical Framework for Sensitivity Analysis with
Incommensurable Observation Data .................................. 97

5.3.1 Derivative-Based Sensitivity Analysis ............................ 99
5.3.2 Pseudo-Derivative-based Sensitivity Analysis ................. 102

5.4 A Case Study: Cross-Layer Optimized Multimedia Delivery over Wireless
Networks ............................................................................. 105

5.4.1 Experiment Environment .............................................. 105
5.4.2 Illustration of Major Cross-Layer Design Tradeoffs ......... 108
5.4.3 Validation of the Proposed Theoretical Framework ........ 112

5.5 Summary .......................................................................... 116

6 Summary and Future Work ............................................. 118

6.1 Summary of Research Contributions ............................... 118
6.2 Future Work .................................................................... 121

6.2.1 Imperfect CSI and Channel Feedback Delay ............... 121
6.2.2 Multi-Users and Multiple Antennas ............................ 121
6.2.3 Content-Aware Video Communications ....................... 121
6.2.4 Power Consumption on Video Consumption and Transmission .... 122
6.2.5 Derive the Analytical Expression of the Design Objective Function .. 122

Bibliography ................................................................. 123
## Figures

2.1 An example of a 10-frame video summary of the Stefan sequence.  
2.2 Video clip reconstruction at the receiver side with video summary (labelled by gray color).  
2.3 The system model of cross-layer optimization for video summary transmission.  
2.4 Path graph of a 3-frame toy video summary transmission.  
2.5 Distortion vs. delay comparison between QP adaptation and no QP adaptation with 30 summary frames.  
2.6 Distortion vs. delay comparison between QP adaptation and no QP adaptation with 60 summary frames.  
2.7 Distortion vs. average SNR comparison between QP adaptation and QP=50 with different $N_{\text{max}}$ for a 30-frame video summary.  
2.8 Distortion vs. average SNR comparison between QP adaptation and QP=50 with different $N_{\text{max}}$ for a 60-frame video summary.  
2.9 Comparisons of distortion vs. delay budget with different $P_L$.  
2.10 Distortion vs. delay budget comparison with different summary frame number.
3.1 The system model of the proposed application-centric routing for video transmission over multi-hop wireless networks. Here, $s$ is the source node and $t$ the destination node. All of the dotted arrowed lines and the solid arrowed lines in the network refer to the available network connectivity obtained from a proactive routing protocol such as *optimized link state routing* (OLSR) [81]; the solid arrowed lines connecting $s$ with $t$ refer to the optimal routing path calculated by the proposed application-centric routing approach.

3.2 Trellis for the joint source coding and path selection problem for a group of slices.

3.3 The total packet loss probability $P^{(u,v)}_{i,j}$ over link $(u,v)$ mainly comprises of packet drop probability $p^{(u,v)}_{i,j}$ due to packet delay deadline expiration during queueing at node $u$ and the packet loss probability $p^{(u,v)}_{i,j}$ including error probability $p^{(u,v),err}_{i,j}$ due to signal fading over link $(u,v)$ and the packet collision probability $p^{(u,v),col}_{i,j}$ due to contention access to the medium.

3.4 An example of a multi-hop wireless network.

3.5 The steps used in computing the optimal path for packet $\pi^j_i$ from source $s$ to destination $t$. The algorithm begins from step (a) and ends at step (f). The arrows indicate the working node. The optimal path is

$s \rightarrow a \rightarrow c \rightarrow d \rightarrow f \rightarrow t$.

3.6 The DAG-modelled connectivity structure between the source node $s$ and the destination node $t$ determined by the OLSR protocol under a given topology in a 100-node network.

3.7 Packet loss probability vs. packet arrival rate $\lambda^u$ under different packet delay deadlines $T^{\text{budget}}$.

3.8 Average received PSNRs using different routing approaches in different sizes of networks. (a) 30 nodes, (b) 100 nodes.

3.9 Frame PSNRs of the received “Mobile” sequence in a 100-node network with four different routing approaches.
3.10 The slice QP selection in a transmission of the “Coastguard” sequence in a 100-node network. .................................................. 66
3.11 Comparison of the received video quality generated by different routing approaches. ........................................... 67

4.1 An illustration of cooperative communications with source node A, relay nodes B and C, and destination node D. ............................... 70
4.2 The proposed system model. ................................................... 73
4.3 Path map for the shortest path algorithm when \(z = 2\). ............. 79
4.4 Performance of the proposed DPF scheme for video summary delivery over cooperative wireless networks. ............................... 82
4.5 Comparison of the received video quality generated by different transmission schemes. ....................................................... 83

5.1 The illustration of temporal-spatial behaviors of cross-layer design by using dynamic programming. The horizontal axis shows the temporal adaptation in cross-layer design, where \(T\) is the adaptation interval and \(\tau\) is the temporal resolution. The vertical axis shows the spatial adaptation, where the continuous values of the design variables are coarsely or finely discretized, resulting in different sets of operating points with different discretization levels. ....................................................... 90
5.2 Illustration of the proposed interaction measure of a cross-layer design system with three design variables \(x_1, x_2,\) and \(x_3\). ................................................................. 93
5.3 The algorithm to calculate the interaction measure \(\mu_k\) in cross-layer design. ................................................................. 96
5.4 Illustration of different sensitivity of the performance of design objective \(Y\) to different design variables \(x_1\) and \(x_2\). Here \(\frac{\Delta Y}{\Delta x_1} \ll \frac{\Delta Y}{\Delta x_2}\). ................................................... 98
5.5 Illustration of the derivative calculation of the design objective \(Y\) with respect to the design variable \(x_n\) by using the observation vectors \(x_n\) and \(y\). ......................................................... 100
5.6 The algorithm for sensitivity analysis by using the derivative-based method. ................................................................. 101
5.7 The dense swarms of observations of design objective $Y$ and design variable $x_n$ ................................................................. 103
5.8 The algorithm for sensitivity analysis in cross-layer design with dense swarms of observations of design objective $Y$ and design variables $x_1, x_2, \ldots, x_N$ ................................................................. 103
5.9 The sparse swarms of observations of design objective $Y$ and design variable $x_n$ ................................................................. 104
5.10 The algorithm for sensitivity analysis in cross-layer design with sparse swarms of observations of design objective $Y$ and design variables $x_1, x_2, \ldots, x_N$ ................................................................. 105
5.11 The system model of the cross-layer optimized multimedia transmission over wireless networks. ........................................... 106
5.12 The temporal relations among frame, slice and macroblock. .......... 109
5.13 The comparison of the received video quality through optimizing QP and MCS on different temporal resolutions of adaptation. ................. 109
5.14 The impact of SINR feedback delay on cross-layer optimized video transmission with different video frame rates under different wireless channels.110
5.15 The comparison of the received video quality of the cross-layer designs with finely-discretized and coarsely-discretized QP operating point sets. ............. 111
5.16 The received video quality with different operating points of both QP and MCS under different channel conditions. ............................ 113
5.17 The received video quality with different operating points of both QP and MCS under the good channel. ............................... 114
5.18 The received video quality with different operating points of both QP and MCS under the bad channel. ................................. 115
5.19 Comparison of video frame quality. ................................. 116
## Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>AMC Modes at the Physical Layer</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>Proposed Optimization Algorithm</td>
<td>45</td>
</tr>
<tr>
<td>3.2</td>
<td>The average accumulating packet delay at each hop along a 6-hop path under different packet delay deadlines $T_{budget}$</td>
<td>61</td>
</tr>
<tr>
<td>3.3</td>
<td>Average video quality achieved by different routing approaches under the two different packet delay deadlines in a 100-node network (To conserve space, the video sequence names are abbreviated as “Ca”: “Carphone”, “F”: “Foreman”, “Co”: “Coastguard”, “M”: “Mobile”).</td>
<td>65</td>
</tr>
<tr>
<td>3.4</td>
<td>Comparison of the proposed application-centric routing with the existing end-to-end path selection algorithm.</td>
<td>66</td>
</tr>
<tr>
<td>4.1</td>
<td>Simulation parameter values</td>
<td>81</td>
</tr>
<tr>
<td>5.1</td>
<td>Operating points of the design variable MCS.</td>
<td>107</td>
</tr>
<tr>
<td>5.2</td>
<td>The calculated interaction measures of the design variables QP and MCS in the case study.</td>
<td>113</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

Wireless networking has been one of the fastest growing industry sector in the past few decades, especially in last ten years. Many new standards and technologies have been proposed and developed, such as IEEE 802.11-based wireless local area networks (WLAN), IEEE 802.16-based Wireless Metropolitan Area Networks (WMAN), IEEE 802.15-based Wireless Personal Area Networks (PAN), the 3rd/4th generation (3G/4G) mobile telecommunication networks, and the 700MHzWhitespace WiFi. With the increasing bandwidth efficiency, energy efficiency, and mobility, wireless networks have enabled various applications and changed in many ways how we communicate.

At the meantime, multimedia communications has significantly facilitated and enriched people’s daily life. People have witnessed the fast development of various wireless multimedia applications, such as video content distribution (e.g., Youtube, P2P streaming by PPStream, PPLive, etc.) and live video communications (e.g., video conferencing by skype video, MSN, etc.). Significant advances in video compression have made it possible to deliver high-quality video at relatively low bit rates. Recent advances in video compression technique are to make video compression algorithms more efficient, more flexible, and more robust against errors. The state-of-the-art video coding standard H.264/AVC has achieved a significant improvement in rate-distortion efficiency relative to existing standards [1]. When used well together, the features of H.264/AVC provide approximately a 50% bit rate savings for equivalent perceptual quality relative to the performance of prior standards. Scalable video coding (SVC) [2, 3] intends to provide simple adaptations for various needs or preferences of end users as well as to varying terminal capabilities or network conditions. As
new paradigm for video compression, distributed video coding [4] reverses the traditional balance of complex encoder and simple decoder and enables a simple encoder at the expense of a more complex decoder, which holds great promise for new generations of mobile video cameras, such as wireless PC cameras and smartphone cameras.

Because of the success of both wireless networks and video compression technology, multimedia applications are expected to become the main theme of the next-generation wireless Internet. It is fully expected that there will be a stronger user demand for bringing multimedia streaming services to the ubiquitous computing devices in the coming years. However, bringing delay-sensitive and loss-tolerant multimedia services based on the current wireless Internet is a very challenging task due to the facts: 1) the original design goal of the Internet, based on the layer concept of the open systems interconnection (OSI) model, is to offer simple delay-insensitive and loss-sensitive data services with little quality of service (QoS) consideration; 2) the current Internet architecture was designed for communication among computers, not communication among humans, making it very challenging to deliver human- and content-oriented services via the Web; 3) Due to the time-varying and fading nature of wireless channels and the limited spectral resources, layer-separated design can no longer guarantee an optimal end-to-end performance for multimedia delivery over wireless networks. Therefore, these facts urge us to rethink the current Internet architecture and develop a new design methodology for multimedia communications over the current and future wireless Internet.

Cross-layer design has been thought as one of the most effective and efficient ways to provide quality of service (QoS) over wireless networks, and it has been receiving many research efforts. Cross-layer design methodology in communications was first studied in adaptive modulation and coding (AMC) [5, 6] to significantly enhance throughput under time-varying channel quality. Adaptive modulation and coding has been advocated at the physical layer by many standard wireless networks, such as 3GPP/3GPP2 [7, 8], HIPER-LAN/2 [9], and IEEE 802.11/16 [10]. Inspired by the big success of adaptive modula-
tion and coding, researchers have applied cross-layer optimization into many other areas, such as cross-layer networking architecture [11–13], joint optimal video analysis and delivery [14–16], joint source-channel coding optimization [17–19], application-QoS provisioning [20,21], video distortion-driven routing [22,23], queueing and scheduling [22,24–26], link adaptation [6,27], and energy efficiency [28,29].

Despite the large amount of research work proposed in recent years, the following research issues for cross-layer design for wireless multimedia communications still remain open problems:

- Most existing work on media streaming focuses on video streaming applications with pre-encoded videos. However, pre-encoded video streaming does not allow for the video coding adaptation to dynamic network conditions and user requirements which is feasible to be performed in real-time video applications for performance improvement. Here, real-time video applications refer to the scenarios where videos are captured, encoded, transmitted, and displayed to end users on-the-fly. Such applications are expected to significantly increase in the coming years, for example, in the form of real-time video surveillance and monitoring, video telephony, live video broadcasting, etc. However, due to the real-time requirement, compared to the pre-encoded video streaming, providing real-time video applications faces more challenges, such as the higher required computational capability and larger memory size of the devices for video coding, the higher required transmission bandwidth, and the required better trade-off between compression-induced distortion and transmission-induced distortion. Therefore, to achieve the optimal performance of cross-layer design for real-time wireless multimedia communications, video coding and transmission should be jointly considered. Although video coding has been considered in some cross-layer design work, such as joint source-channel coding [17–19], critical issues related to video transmission over wireless networks has been superficially addressed due to the use of oversimplified network models in these work, such as, how to satisfy the delay
constraint of real-time video communications, how to achieve the optimal received end-to-end video quality in a multi-hop wireless network, and how to maximize network resource utilization.

- Current research on cross-layer design for wireless multimedia communications has been carried out in various piecemeal approaches with different specific design objectives. For example, some work focuses on the maximization of network throughput, the minimization of transmission delay, or the user-received video distortion. Piece-meal approaches may cause so-called design paradoxes such as the *Ellsberg Paradox* [30], in which each individual design variable residing at certain network layer makes the “best” decision to maximize the design objective at the local scope, but the overall system performance may be worse than that of not doing any optimization. Therefore, current research on cross-layer design lacks a theoretical study of the methodology of cross-layer optimization. Although some researchers have realized the needs for quantitative analysis of the design of cross-layer solutions [31] and have proposed some formal methods [31–41] for the cross-layer modeling and optimization of wireless multimedia networks, there is still lacking of a rigorous mathematical model to gain in-depth understanding of cross-layer design. Therefore, research is still needed to provide a theoretical formulation and quantitative analysis on cross-layer design issues, which will be crucial to avoid design pitfalls in existing design and provide insights for future design.

### 1.2 Research Scope

In this research, we investigate the performance of cross-layer optimization solutions for real-time video communications over three types of wireless networks, i.e., single-hop wireless networks, multi-hop wireless networks, and cooperative wireless networks. We consider the joint optimization of the video coding parameters of an H.264 codec at the application
layer and the transmission parameters at lower layers in a cross-layer manner. From the point of view of end users, the end-to-end video quality is the most straightforward and reasonable objective function in any optimization framework for video streaming. Therefore, we implement the recognized “recursive optimal per-pixel estimate” (ROPE) method in the H.264 video codec to calculate the expected received video distortion under the video transmission delay constraint. With the user-centric objective function, we proposed several cross-layer optimization frameworks for wireless multimedia networking: 1) we study content-aware video communications in single-hop wireless networks by exploring the transmission of video summaries, where a good content coverage is achieved by the proposed link adaptation schemes [42]; 2) the routing issue is considered for real-time video streaming over multi-hop wireless networks since path selection in video transmission significantly affects the end user experience of video applications [43–45]; and 3) we investigate video summary transmission over cooperative wireless networks to exploit the spatial diversity of cooperative communications and enhance the network resource utilization [46]. Our research has produced fundamental theories and practical algorithms for the cross-layer design and optimization of wireless multimedia networking. Significant performance gains are obtained by our solutions.

To gain in-depth understanding of complex cross-layer behaviors such as multiscale temporal-spatial behavior, we propose a framework to quantitatively and systematically study the cross-layer interactions based on the non-additive measure theory [47–49]. We conduct a case study on cross-layer optimized wireless multimedia communications to verify the proposed framework. Our research can significantly enhance our capability for cross-layer behavior characterization and provide insights for future design [50].

1.3 Dissertation Outline

The remainder of this dissertation is organized as follows.

In Chapter 2, we present the study on cross-layer optimization for video summary trans-
mission over single-hop wireless networks [42, 51]. We first conduct a literature overview of existing research on wireless video summary transmission. Then, we describe the system model and present the problem formulation. To achieve a good video content coverage, a link adaptation scheme combining adaptive modulation and coding (AMC) and automatic repeat request (ARQ) will be discussed [52–55]. We present the solution to the formulated problem as a combination of Lagrangian relaxation and dynamic programming. To further validate the benefit of the proposed framework, we conduct simulation by using H.264/AVC JM 10.2 and MATLAB, and report the experimental results.

In Chapter 3, we extend our research on cross-layer optimized video communications to multi-hop wireless networks by investigating the routing issue of wireless real-time video streaming [43–45, 56]. We first introduce existing work in the field and motivate the proposed application-centric routing. Then, after describing the system model, we formulate the problem as to find the optimal video coding parameters and transmission paths to maximize the expected end-to-end video distortion under a video transmission delay constraint. The solution procedures and the proposed routing algorithm will be detailedly discussed, including the analysis on the computational complexity of the proposed routing framework. Finally, we report the extensive experiments in verifying the superior performance of the proposed framework.

In Chapter 4, we present a generic cross-layer optimization framework for video summary transmission over cooperative wireless networks [46]. The motivation is to inherit both advantages from cooperative communications and video summarization. We propose a novel decode-process-and-forward (DPF) scheme by equipping relay nodes with the video processing capability. We first formulate a problem as to jointly optimize the source coding, relay video processing parameters, power allocation among the source and relay nodes, and error concealment strategy to achieve the best video quality under the constraints of transmission delay and power consumption. Then, we present the solution to the problem. Finally, we report the experiments and present the results of the comparison between the
proposed DPF scheme and conventional transmission schemes.

In Chapter 5, we present our theoretical study on the methodology of cross-layer design and optimization [47–49]. First, we give the definition of the interaction measure and introduce the quantitative model of the interaction measure in cross-layer design. Then, we describes different methods of sensitivity analysis based on quantitative interaction measure. Finally, a case study on cross-layer optimized wireless multimedia communications is conducted to illustrate the major cross-layer design tradeoffs and to validate the proposed framework.

We present our summary of the research contributions and future research directions in Chapter 6.
Chapter 2

Cross-Layer Optimized Video Summary Transmission over Single-Hop Wireless Networks

2.1 Introduction

In recent years, universal multimedia access (UMA) [57,58] is emerging as one of the most important components for the next generation of multimedia applications. The basic idea of UMA is universal or seamless access to the multimedia content by automatic selection or adaptation of content following user’s interaction. As mobile phones have grown in popularity and capability, people have become enthusiastic about watching multimedia content using mobile devices and personalizing the content, for example, summarizing the video for real-time retrieval or for easy transmission. In general, the video summarization algorithm will generate a still-image storyboard, which is composed of a collection of salient images extracted from the underlying video sequence, as shown in Figure 2.1.

Although plenty of works on video summarization can be found in literature [14,59–61], the transmission issue of video summary has gained little attention. [62] extends the work of [14] into the wireless video streaming domain, however the packet loss factor due to unsatisfactory wireless channel conditions has not been considered in the framework. In [15], packet loss is considered in the video summary transmission, and the key frames that minimize the expected end-to-end distortion are selected as the summary frames. However, the source coding has not been optimized in the optimization framework, which might directly impact the perceptual quality of the results. In addition, the algorithm does not guarantee a good content coverage aspect of the selected frames because potential packet loss penalty heavily biases the selection process.

In wireless networks, packet loss is mainly due to the fading effect of time-varying
wireless channels. Adaptive modulation and coding (AMC) have been studied extensively and advocated at the physical layer, in order to match transmission rates to time-varying channel conditions. For example, to achieve high reliability at the physical layer, one has to reduce the transmission rate using either small size constellations, or powerful but low-rate error-correcting codes [5, 63, 64]. An alternative way to decrease packet loss rate is to rely on the automatic repeat request (ARQ) protocol at the data link layer, which requests retransmissions for those packets received in error. Obviously, by allowing a very large retransmission number, ARQ can guarantee a very low packet loss rate. However, to minimize delays and buffer sizes in practice, truncated ARQ protocols have been widely adopted to limit the maximum number of transmissions [65].

A variety of techniques have been proposed to address the problem of multimedia delivery over lossy networks. [66] shows that the problem of rate-distortion optimized streaming of an entire presentation can be reduced to the problem of error-cost optimized transmission of an isolated data unit. Based on this observation, a general framework for rate-distortion optimized streaming of packetized media over a lossy packet network is set up for various transmission scenarios. In [66], distortion-rate performance is measured in an average sense, and practical streaming using window and rate control is proposed to overcome a possible large instantaneous rate. [67] and [68] propose a proxy-driven rate-distortion optimized streaming over a lossy packet network, exploiting a proxy located at the edge of the
backbone network to coordinate the streaming process. Traffic load brought from packets lost in the last hop is relieved and end-to-end performance is greatly improved.

In this chapter, within an expected rate-distortion framework where expectation are taken over channel realizations as in [66], [67] and [68], we focus our study on the cross-layer optimization of the video summary transmission over lossy networks. We assume a video summarization algorithm that can select frames based on some optimality criteria is available in the system. Therefore, a cross-layer approach is proposed to jointly optimize the AMC parameters at the physical layer, the ARQ parameters at the data link layer, and the source coding parameters at the application layer to achieve the best video quality of reconstructed video clips from received video summaries. Clearly, due to the spectacular characteristics of video summary data, the general cross-layer optimization schemes recently proposed for normal video sequences [69] and [70] do not automatically cover the summary data transmission. As an example, the neighboring summary frames have typically less correlation in order to cover the content variation of the video clip, and thus the normal temporal-based error concealment algorithm considered in the [69] and [70] would not be efficient in the current scenario. In [71], a cross-layer multi-objective optimized scheduler for video streaming over 1xEV-DO system is presented. With the usage of decodability and semantic importance feedback from the application layer to the scheduler, [71] focuses on determining the best allocation of channel resources (time slots) across users, however, the joint optimization of source coding and transmission parameters has not been considered in the framework.

In the proposed framework, we assume the existence of a system controller, whose responsibility is to communicate with each layer and dynamically determine the corresponding parameters that guarantee the best output video quality, and then drive the application system to perform efficiently. In addition, the framework tries to maintain a good content coverage by providing tunable parameters to avoid cases that a number of consecutive summary frames being lost simultaneously. This task is not trivial due to the complexity of the
underlying wireless link protocols. Compared to the existing systems, the novelty of this
work is in twofold: first, this is the first cross-layer optimization framework proposed for
the coding and transmission of video summary data; second, AMC and ARQ are jointly
considered in the cross-layer design, which gives the controller more flexibility in delivering
the summary frames.

The remainder of this chapter is organized as follows. Section 2.2 provides a brief
description of the background techniques adopted in this chapter and then highlights the
problem formulation. Section 2.3 provides the system model of our framework. Section
2.4 describes the link adaptation principles, while Section 2.5 details the algorithm of the
controller. Section 2.6 provides experimental results and shows the effectiveness of the
proposed framework. Finally, concluding remarks are given in Section 2.7.

2.2 Problem Statement

The section first provides a brief explanation of background techniques used in this chapter,
including video summarization at the application layer, link adaptation by combining ARQ
at the data link layer and AMC at the physical layer. Then a cross-layer optimization
problem is formulated for video summary transmission by jointly determining the optimal
parameters for each layer.

2.2.1 Background

Clearly in wireless communication applications, video transmission suffers mainly from un-
reliable channel conditions and excessive delays. In source coding, setting a finer coding
parameter will directly improve the coded video quality, however, might increase the trans-
mission time and increase the chance of getting corrupted by the transmission error. Video
summary is a special format of the video clip whose correlation between frames are not
as high as normal clips, but losing consecutive or continuous summary frames might cause
severe damage for understanding the summary content. The summary could be automatic
generated or selected with user's interactions. In this research how to generate the video summary is not within the scope of our framework.

At the data link layer, ARQ is widely used to mitigate channel fading and decrease packet error rate (PER). Once some information packets are lost in transmission, retransmission requests are activated and those packets are sent out again. Obviously, by allowing a very large retransmission number, ARQ can guarantee a very low PER. However, large retransmission number means large delay, especially when the round trip time (RTT) of wireless channels is large. Considering the requirement on the allowable maximum delay for transmitting one video summary frame, we need to set a prescribed maximum transmission times for one summary frame.

At the physical layer, AMC has been advocated to enhance the throughput of future wireless data communication systems. In AMC, different size constellations and different rate error-control codes are chosen based on different time-varying channel conditions. For example, in good channel conditions, AMC schemes with large size constellations and high rate error-control codes can increase the system transmission rate while guaranteeing a good reliability. This means that AMC can effectively decrease transmission delay while satisfying some PER constraint. In this research, we combine AMC with ARQ to yield a link adaptation to achieve a desirable delay-PER tradeoff for video summary transmissions. With ARQ correcting occasional packet errors at the data link layer, the stringent error control requirement is alleviated for the AMC at the physical layer. In turn, with less stringent error control requirement, AMC schemes with large size constellations and high rate error-control codes are more likely to be chosen, leading to higher transmission rate and smaller transmission delay.

Clearly, the tradeoff among the selected parameters in these layers are mixed, for example, to maintain a reasonable delay, the source coding might choose a coarser parameter, or AMC chooses a larger size constellation or a higher rate FEC channel code, which increases the vulnerability of coded frames and will cause unacceptable video quality. However, if
source and channel coding use more bits, with the increase of packet length, the probability of packet loss increases, and ARQ might have to increase the number of retransmission trials to reduce the quality problem due to packet loss, and then will result in excessive delay. Therefore, cross-layer optimization approach is a nature solution to improve the overall system performance.

2.2.2 Problem Formulations

In this chapter, we propose a cross-layer framework that optimizes the parameter selection in AMC at the physical layer, ARQ at the data link layer and source coding at the application layer to achieve the best video quality of reconstructed video clip from the received video summary.

The following notation will be used. Let us denote by $n$ the number of frames of a video clip $\{f_0, f_1, \ldots, f_{n-1}\}$, and $m$ the number of frames of its video summary $\{g_0, g_1, \ldots, g_{m-1}\}$.

Let $S_i$ and $B_i$ be the coding parameters and the resultant consumed bits of the $i$th ($i = 0, 1, \ldots, m-1$) video summary frame in lossy source coding. The summarization with different coding parameters will produce summary frames with different frame lengths. Large size frames will be fragmented into multiple packets for transmission at lower layers. Let $Q_i$ denote the number of fragmented packets of the $i$th summary frame. Let $N_{i,q}$ and $F_{i,q}$ be the number of transmissions and the packet size for the $q$th packet of the $i$th summary frame, respectively. To improve channel utilization, AMC is designed to update the transmission mode for every transmission and retransmission of each packet. Let $R_{i,q,n}(A_{i,q,n}, C_{i,q,n})$ be the rate (bits/symbol) of AMC mode used at the $n$th transmission attempt when transmitting the $q$th packet of the $i$th summary frame, where $A_{i,q,n}$ and $C_{i,q,n}$ are the corresponding modulation order and coding rate. We assume that the transmission rate of the physical layer channel is fixed, denoted by $r$ (symbols/second). Clearly, the delay in transmitting
the whole summary can be expressed by
\[ T = \sum_{i=0}^{m-1} \sum_{q=1}^{Q_i} \sum_{n=1}^{N_i} \left[ \frac{F_i(q, B_i)}{R_i(q, n, A_i, C_i, \gamma_{i,q,n})} \right] + T_{RTT} \]  
(2.1)

where \( T_{RTT} \) is the maximum allowed RTT to get the acknowledgement packet via the feedback channel before a retransmission trial.

Let \( l_i \) be the index of the summary frame \( g_i \) in the video clip. At the receiver side, the video clip is reconstructed by substituting missing frames with the corresponding summary frames. Let \( \tilde{f}_k \) denote the displayed \( k \)th frame from the received summary at the receiver side. Let \( \tilde{g}_i \) be the reconstructed the \( i \)th summary frame. Specifically, for the example shown in Figure 2.2, if the summary frame \( \tilde{g}_{i+1} \) is successfully transmitted, video frames \( \{\tilde{f}_{l_i+1}, \tilde{f}_{l_i+1+1}, \tilde{f}_{l_i+1+2}, \tilde{f}_{l_i+1+3}\} \) are reconstructed by \( \tilde{g}_{i+1} \) (denoted by solid lines); otherwise, they will be reconstructed by \( \tilde{g}_i \) (denoted by dotted lines). If \( \tilde{g}_i \) is also lost, \( \tilde{g}_{i-1} \) will be considered. This process continues until the closest correctly received summary frame to \( f_k \) is available. Note that any summary frame is possible to get lost during video transmission. However, in order to simplify the problem formulation, we assume the first summary frame would guarantee to be received.

Let \( \rho_i(S_i, B_i, N_{max}, A_{i,q,n}, C_{i,q,n}, \gamma_{i,q,n}) \) be the loss probability of the \( i \)th summary frame, where \( N_{max} \) is the maximum transmission number for one packet and \( \gamma_{i,q,n} \) is the instantaneous channel SNR. Based on the aforementioned video reconstruction process at the

Figure 2.2: Video clip reconstruction at the receiver side with video summary (labelled by gray color).
receiver side, the expected distortion of the video clip can be calculated by

\[
E[D] = \sum_{k=0}^{n-1} E[D(f_k, \tilde{f}_k)]
\]

\[
= \sum_{i=0}^{m-1} \sum_{l_i} \sum_{b=0}^{l_i-1} \left\{ (1 - \rho_{l-i}) d[f_j, \tilde{g}_{\min(i, L-1)}] \cdot b \prod_{a=0}^{b-1} \rho_{l-a} \right\}
\]

(2.2)

where function \(d(\ )\) is the distortion between two frames. In this work we use the mean squared error (MSE) between the two frames as the metric for calculating the distortion. The same distortion measure for the video summary result has been used in [14,15,62].

The problem at hand can be formulated as

\[
\text{Min } E[D], \quad \text{s.t. : } T \leq T_{\text{max}}
\]

(2.3)

where \(T_{\text{max}}\) is a given delay budget for delivering the whole video clip.

In this research, we consider the content coverage issue of the received summary. In other words, if a chunk of continuous summary frames are lost due to the channel error, then the coverage of the received summary for the original clip would be degraded significantly. To avoid such a problem but still keep the problem as general as possible, we define \(L\) such that the case of \(L\) or more than \(L\) consecutive summary frames being lost will never happen. For instance, if \(L = 2\) then no neighboring summary frames can be lost together during transmission. So that the distortion can be calculated by

\[
E[D] = \sum_{k=0}^{n-1} E[D(f_k, \tilde{f}_k)]
\]

\[
= \sum_{i=0}^{m-1} \sum_{l_i} \sum_{b=0}^{l_i-\min(i, L-1)} \left\{ (1 - \rho_{l-i}) d[f_j, \tilde{g}_{\min(i, L-1)}] \cdot b \prod_{a=0}^{b-1} \rho_{l-a} \right\}.
\]

(2.4)

It is important to realize that the value of \(L\) is a programmable constant by the system, and the introduction of \(L\) does not narrow down the original problem. As you may notice, when we set \(L = m\), the Eq. (2.4) is equal to Eq. (2.2).

If we set \(L = 2\) in (2.4), it is very clear that for the \(i\)th summary frame, there are only two possibilities: either it is received or it is lost but its previous summary frame is received.
Table 2.1: AMC Modes at the Physical Layer

<table>
<thead>
<tr>
<th>Modulation</th>
<th>Mode1</th>
<th>Mode2</th>
<th>Mode3</th>
<th>Mode4</th>
<th>Mode5</th>
<th>Mode6</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QPSK</td>
<td>1/2</td>
<td>1/2</td>
<td>3/4</td>
<td>9/16</td>
<td>3/4</td>
<td>3/4</td>
</tr>
<tr>
<td>Coding Rate $C_m$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R_m (bits/sym.)</td>
<td>0.50</td>
<td>1.00</td>
<td>1.50</td>
<td>2.25</td>
<td>3.00</td>
<td>4.50</td>
</tr>
<tr>
<td>$a_m$</td>
<td>1.1369</td>
<td>0.3351</td>
<td>0.2197</td>
<td>0.2081</td>
<td>0.1936</td>
<td>0.1887</td>
</tr>
<tr>
<td>$b_m$</td>
<td>7.5556</td>
<td>3.2543</td>
<td>1.5244</td>
<td>0.6250</td>
<td>0.3484</td>
<td>0.0871</td>
</tr>
</tbody>
</table>

Let us denote by $G_i$ the chance of the $i$th summary frame being not lost, so $G_i = 1$ means it is guaranteed to be received, otherwise it is not guaranteed. Based on the constraint, we need $\max(G_i, G_{i-1}, \ldots, G_{i+1-L}) = 1$ for all $i \in [0, m-1]$. $G_i$ can be guaranteed and derived by link adaptation, which will be discussed in Section 2.4.

For delay issue, we hope that the total delay $T$ of delivering all summary frames satisfies $T \leq T_{\text{max}}$. Therefore, the problem is

$$\text{Min} E[D], \quad \text{s.t.:} \quad T \leq T_{\text{max}}, \quad \text{and} \quad \max(G_i, G_{i-1}, \ldots, G_{i+1-L}) = 1, \quad i \in [0, m-1]. \quad (2.5)$$

Once we work out the problem (2.5), the optimal parameter combinations, i.e., source coding parameter $S_i$, AMC modulation order $A_{i,q,n}$ and channel coding rate $C_{i,q,n}$, and ARQ transmission number $N_{\text{max}}$ are obtained to transmit the $i$th summary frame, which minimizes the whole clip distortion and satisfies certain predefined delay constraint.

2.3 System Model

The system model of the proposed framework is shown in Figure 2.3, which consists of a 3-layer structure and a controller.

At the application layer, summarization is performed on the target video clip and large size summary frames are fragmented into multiple packets for transmission at lower layers. At the data link layer, ARQ protocol is adopted. If an error is detected in a packet, a
retransmission request is generated by the receiver, and is sent to the transmitter via a feedback channel. The transmitter arranges retransmission of the requested packet. If a packet is not received correctly after $N_{\text{max}}$ transmission attempts, we will declare packet loss, then the summary frame to which the lost packet belongs is also regarded as lost. At the physical layer, we assume that multiple transmission modes are available as shown in Table 2.1, with each mode consisting of a specific modulation and FEC code pair as in 3GPP, HIPERLAN/2, IEEE 802.11a, and IEEE 802.16 standards [7,9,10]. Based on channel state information (CSI) from the channel estimator, the transmitter updates the AMC mode for the next packet transmission. Coherent demodulation and maximum-likelihood (ML) decoding are used at the receiver. The decoded bit streams are mapped to packets, which are pushed upwards to the data link layer. If all fragmented packets of the summary frame $g_i$ are correctly delivered, the summary frame $g_i$ is saved into the buffer, and the video clip frames $f_i$ through $f_{i+1}-1$ are reconstructed with $g_i$. If $g_i$ does not reach the receiver after some fixed time, the video clip frames $f_i$ through $f_{i+1}-1$ will be reconstructed with the previously received summary frame. Obviously, the receiver only need a buffer that can
contain one summary frame, i.e., the latest received summary frame.

From above description, it is obvious that AMC combined with ARQ performs a link adaptation in a joint approach. For a fixed video summary, say \{g_0, g_1, \cdots, g_{m-1}\}, the link adaptation can guarantee the constraint \(\max(G_i, G_{i-1}, \ldots, G_{i+1-L}) = 1\) in problem (2.5), and produce the summary frame error rate (FER) \(\rho_i\) and transmission time \(T_i\) for each summary frame \(g_i\). The detailed link adaptation and close-form expressions for \((\rho_i, T_i)\) will be clarified in the Section 2.4.

The controller is the most important part of the system, which is equipped with all possible values of the key parameters of each layer. These parameters include the coding parameter \(S\) at the application layer, the allowed maximum transmission number \(N_{\text{max}}\) at the data link layer, and the available AMC modes with modulation order and FEC code rate pair \((A, C)\). Note that here \(S, A,\) and \(C\) are parameter allocation vectors for \(m-1\) summary frames, for example, \(S = \{S_1, S_2, \cdots, S_{m-1}\}\).

The following is a brief list of performing flows of our proposed framework.

- When there is a video clip to transmit, based on the current average SNR \(\Gamma\) from the channel estimator, from all possible values of parameter set \(\{S, N_{\text{max}}, A, C\}\), the controller first calculates all possible theoretical values of the pair \((\rho_i, T_i)\) for all possible summary frames by using the close-form expressions of link adaptation performance with the constraint \(\max(G_i, G_{i-1}, \ldots, G_{i+1-L}) = 1, i \in [0, m-1]\).

- With the total delay budget \(T_{\text{max}}\), the controller use all possible \((\rho_i, T_i)\)s for the whole summary to solve problem (2.5). The group values of \(\{S, N_{\text{max}}, A, C\}\) corresponding to the optimal solution of problem (2.5) are the optimal parameters to transmit the whole video summary.

- The obtained optimal parameters \(\{S, N_{\text{max}}, A, C\}\) are assigned to the corresponding layers, then the whole video summary is sent out frame by frame.
• Corresponding video clip frames are reconstructed with the newly received summary frame.

We next list the operating assumptions adopted in this chapter.

• The channel is frequency flat, remains time invariant during a packet, but varies from packet to packet. Thus, AMC is adjusted on a packet-by-packet basis. In other words, AMC scheme is updated for every transmission and retransmission attempt. The channel quality is captured by a single parameter, namely the received SNR $\gamma$. we adopt Rayleigh channel model to describe $\gamma$ statistically. The received SNR $\gamma$ per packet is thus a random variable with a probability density function (pdf):

$$p_{\gamma}(\gamma) = \frac{1}{\bar{\gamma}} \exp\left(-\frac{\gamma}{\bar{\gamma}}\right)$$

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

• Perfect channel state information (CSI) is available at the receiver. The corresponding mode selection is fed back to the transmitter without error and latency. This assumption could be at least approximately satisfied by using a fast feedback channel with powerful error control information as adopted in IEEE 802.16 [10].

• Error detection based on CRC is perfect, provided that sufficiently reliable error detection CRC codes are used.

2.4 Link Adaptation

In this section, we explain how link adaptation can guarantee the constraint $\max(G_i, G_{i-1}, \ldots, G_{i+1-L}) = 1$, and derive the close-form expression of $(\rho_i, T_i)$.

Actually, it is impossible to strictly guarantee the constraint $\max(G_i, G_{i-1}, \ldots, G_{i+1-L}) = 1$, due to the fading characteristics of wireless channels. Let $P_L$ be the probability that $L$ consecutive summary frames are lost simultaneously. We assume $P_L$ to be a very small value, say $10^{-2}$, to approximate the constraint $\max(G_i, G_{i-1}, \ldots, G_{i+1-L}) = 1$ is satisfied.
Then the goal of link adaptation becomes to guarantee $P_L$ with the least total transmission delay.

Since the processing unit of link adaptation is packet, we need to transform $P_L$ into target packet error rate $P_{\text{target}}$ of lower layers. Let $L_s$, $L_f$ and $L_a$ be summary frame size, fragmentation packet size and the actual packet length of link adaptation, respectively, According to different summary frame sizes, there are two possible cases:

- The summary frame size is smaller than the fragmentation packet size. Since there is no need to do fragmentation, we have $P_{\text{target}} = P_L^{1/L}$ and $L_a = L_s$.

- The summary frame size is larger than the fragmentation packet size, where fragmentation is necessary. A summary frame of length $L_s$ will be fragmented into $N_p = \lceil L_s/L_f \rceil$ packets. $\lceil \cdot \rceil$ is the smallest integer greater than or equal to a given real number. The actual packet size $L_a$ of the first $\lceil L_s/L_f \rceil - 1$ packets equals to $L_f$, and the actual packet size $L'_a$ of the final packet is $L_s - \lfloor L_s/L_f \rfloor \cdot L_f$. The target PER should be as follows:

\[
P_{\text{target}} = 1 - (1 - P_L^{1/L})^{1/N_p}
\]  

(2.7)

In the above both cases, $P_{\text{target}}$ can be regarded as the required PER at the data link layer. Next we explain how to guarantee $P_{\text{target}}$ with transmission packet size $L_a$ by AMC and ARQ. Let us define a PER upper bound $P_{\text{AMC}}$ such that the instantaneous PER is guaranteed to be no greater than $P_{\text{AMC}}$ for each chosen AMC mode at the physical layer. Then the PER at the data link layer after $N_{\text{max}}$ transmissions is no larger than $P_{\text{AMC}}^{N_{\text{max}}}$. To satisfy $P_{\text{target}}$, we need to impose

\[
P_{\text{AMC}}^{N_{\text{max}}} = P_{\text{target}}, \ i.e., \ P_{\text{AMC}} = P_{\text{target}}^{1/N_{\text{max}}}
\]  

(2.8)

We assume each bit inside the packet has the same bit error rate (BER) and bit-errors are uncorrelated, the PER can be related to the BER through

\[
\text{PER} = 1 - (1 - \text{BER})^{L_a}
\]  

(2.9)
for a packet containing $L_a$ bits. For any AMC mode, to guarantee the upper bound $P_{AMC}$, the required BER to achieve is

$$BER_{AMC} = 1 - (1 - P_{AMC})^{1/L_a}$$

(2.10)

Since exact closed-form BERs for the AMC modes in Table 2.1 are not available, to simplify the AMC design, we adopt the following approximate BER expression:

$$BER_m(\gamma) = a_m \exp(-b_m \gamma)$$

(2.11)

where $m$ is the mode index and $\gamma$ is the received SNR. Parameters $a_m$ and $b_m$ are obtained by fitting (2.11) to the exact BER. To guarantee $P_{AMC}$ with the least delay when transmitting a packet, we set the mode switching threshold $\gamma_m$ for the AMC mode $m$ to be the minimum SNR required to achieve $BER_{AMC}$. By (2.11) $\gamma_m$ can be expressed as

$$\gamma_m = \frac{1}{b_m} \ln \left( \frac{a_m}{BER_{AMC}} \right), \quad m = 1, 2, \cdots, M,$$

$$\gamma_{M+1} = +\infty,$$

(2.12)

where $M$ is the total number of AMC modes available.

Since the instantaneous PER is upper-bounded by $P_{AMC}$ in our AMC design, the average PER at the physical layer will be lower than $P_{AMC}$. Taking expectations over channel realizations, the average PER at the physical layer is

$$\overline{P} = \frac{1}{P_T} \sum_{m=1}^{M} \int_{\gamma_m}^{\gamma_{m+1}} PER_m(\gamma)p_\gamma(\gamma)d\gamma$$

$$= \frac{1}{P_T} \sum_{m=1}^{M} \int_{\gamma_m}^{\gamma_{m+1}} \left[ 1 - (1 - a_m \exp(-b_m \gamma))^{L_a} \right] \cdot p_\gamma(\gamma)d\gamma$$

(2.13)

where $P_T = \int_{\gamma_1}^{+\infty} p_\gamma(\gamma)d\gamma$ is the probability that channel has no deep fades and at least one AMC mode can be adopted. Similarly, the average delay for one transmission attempt at the physical layer can be expressed as

$$\overline{T} = \frac{1}{P_T} \sum_{m=1}^{M} \int_{\gamma_m}^{\gamma_{m+1}} \left( \frac{L_a}{R_m \cdot r} + T_{RTT} \right)p_\gamma(\gamma)d\gamma$$

(2.14)
Then the average number of transmission attempts per packet can be found as [6]

\[
N = 1 + P + P^2 + \ldots + P^{N_{\text{max}} - 1} = \frac{1 - P^{N_{\text{max}}}}{1 - P}.
\]

(2.15)

Then the actual PER at the data link layer is

\[
P_{\text{actual}} = P^{N_{\text{max}}},
\]

(2.16)

and the actual transmission delay for each packet at the data link layer is

\[
T_{\text{actual}} = \bar{T} \cdot N.
\]

(2.17)

When to calculate the actual FER \( \rho_i \) and the actual delay \( T_i \) for transmitting the \( i \)th summary frame, two cases should be considered as mentioned before:

- If the summary frame size \( L_s \) is smaller than the fragmentation packet size \( L_f \), we adopt \( L_a = L_s \) to compute \( P(L_a) \) and \( T(L_a) \) and we can have \( \rho_i = P_{\text{actual}}(L_a) \) and \( T_i = T_{\text{actual}}(L_a) \) with (2.13)-(2.17).

- If the summary frame size \( L_s \) is larger than the fragmentation packet size \( L_f \), we adopt \( L_a = L_f \) and \( L_a' = L_a - \lfloor L_s / L_f \rfloor \cdot L_f \) to compute \( P_{\text{actual}}(L_a) \), \( T_{\text{actual}}(L_a) \), \( P'_{\text{actual}}(L_a') \), and \( T'_{\text{actual}}(L_a') \). Then we can have

\[
\rho_i = 1 - (1 - P_{\text{actual}})^{N_p - 1} \cdot (1 - P'_{\text{actual}}),
\]

(2.18)

\[
T_i = (N_p - 1) \cdot T_{\text{actual}} + T'_{\text{actual}}.
\]

(2.19)

The above closed-form expressions of \( \rho_i \) and \( T_i \) will be used by the controller to calculate all possible \((\rho_i, T_i)\) to solve problem (2.5), which we will detail in Section 2.5.
2.5 Algorithm of the Controller

2.5.1 Optimal Solution

Since the problem (2.5) is a constrained minimization problem, it can be solved by La- 
grangian relaxation. So the problem can be converted into

\[
\text{Min}\{E[D] + \lambda T\}, \quad \text{s.t. :}
\]

\[
\max(G_i, G_{i-1}, \ldots, G_{i+1-L}) = 1, \quad i \in [0, m - 1].
\] (2.20)

The target to be minimized can be derived as the following Lagrangian cost function:

\[
J_{\lambda} = E[D] + \lambda T
= \sum_{i=0}^{m-1} \sum_{j=l_i}^{l_{i+1}-1} \sum_{b=0}^{\min(i,L-1)} \left\{ (1 - \rho_{i-b}) d[f_j, \tilde{g}_{i-b}(S_{i-b})] \prod_{a=0}^{b-1} \rho_{i-a} + \lambda T_i \right\}.
\] (2.21)

Let us define a cost function \(H_i(u_i)\) to represent the sum of distortion and delay for up 
to \(i\)th summary frame, where \(u_i\) represents the parameter vector \(\{S_i, N_{\text{max}}, A_{i,q,n}, C_{i,q,n}, \gamma_{i,q,n}\}\). Clearly it can be observed that

\[
H_i(u_i) = H_{i-1}(u_{i-1}) + \sum_{j=l_i}^{l_{i+1}-1} \sum_{b=0}^{\min(i,L-1)} \left\{ (1 - \rho_{i-b}) d[f_j, \tilde{g}_{i-b}(S_{i-b})] \prod_{a=0}^{b-1} \rho_{i-a} + \lambda T_i \right\},
\] (2.22)

which means the process of choosing \(u_i\) for the \(i\)th summary frame is independent of 
\(\{u_0, u_1, \ldots, u_{i-2}\}\), the parameters selected for the first \(i - 1\) summary frames. This is the 
fundamental of dynamic programming (DP). So the optimal solution can be found by a 
shortest path algorithm.

As a toy example, we assume there are three summary frames \(\{g_0, g_1, g_2\}\) to be sent 
and assume \(L = 2\). In addition, we suppose for each summary frame, there are \(k\) different 
source coding options. Then the path graph will be like Figure 2.4. In this figure, each node 
\(u_i^a\) corresponds to a cost value \(H(u_i^a)\). The weight \(h(u_i^a u_{i+1}^b)\) on each branch from node \(u_i^a\)
to $u_{i+1}^b$ corresponds the incremental cost value when transmitting the $(i + 1)$th summary frame with the $b$th source coding option. $h(u_i^a u_{i+1}^b)$ can be computed by the second term of the right hand side of (2.22).

As discussed before, the solution to problem (2.5) is to minimize the average distortion $D$ for a total delay budget $T_{\text{max}}$ in transmitting a whole video summary. With the path graph like above, the goal of the controller is to find the shortest path in the graph with the forward DP. The obtained shortest path has the minimal distortion $D$, and at the same time indicates the optimal choice of parameters $\{S_i, N_{\text{max}}, A_{i,q,n}, C_{i,q,n}\}$ for source coding and transmitting the $i$th summary frame.

For $J_\lambda$ in (2.21), it has been shown [72] that if there is a $\lambda^*$ such that

$$\{S^*, N_{\text{max}}^*, A^*, C^*\} = \arg \min J_{\lambda^*}(S, N_{\text{max}}, A, C)$$

(2.23)

leads to $T(S, N_{\text{max}}, A, C) = T_{\text{max}}$, then $\{S^*, N_{\text{max}}^*, A^*, C^*\}$ is also an optimal solution to (2.5). It is well known that when $\lambda$ sweeps from zero to infinity, the solution to problem (2.23) traces out the convex hull of the distortion delay curve, which is a non-increasing function. Hence $\lambda^*$ can be obtained via a fast convex recursion in $\lambda$ using the bisection algorithm.

Next we list the algorithm to find $\lambda^*$.

---

**Figure 2.4:** Path graph of a 3-frame toy video summary transmission.
• Step 1: We judiciously choose two values of $\lambda$, $\lambda_l$ and $\lambda_u$ with $\lambda_l \leq \lambda_u$ which satisfy the relation:
\[
\sum_i T^*_i(\lambda_u) \leq T_{\text{max}} \leq \sum_i T^*_i(\lambda_l)
\]  
(2.24)
where $\sum_i T^*_i(\lambda)$ is the total delay corresponding to the shortest path found by forward DP. A conservative choice for a solvable problem would be $\lambda_l = 0$ and $\lambda_u = \infty$;

• Step 2: $\lambda_{\text{next}} \leftarrow \frac{\lambda_l + \lambda_u}{2}$.

• Step 3: Perform forward DP through the path graph for $\lambda_{\text{next}}$:
  \[
  \Rightarrow \text{if } \{\sum_i T^*_i(\lambda_{\text{next}}) = \sum_i T^*_i(\lambda_u)\}, \text{ then stop, } \lambda^* = \lambda_u;
  \]
  \[
  \Rightarrow \text{else if } (\sum_i T^*_i(\lambda_{\text{next}}) > T_{\text{max}}), \lambda_l \leftarrow \lambda_{\text{next}}, \text{ Go to Step 2,}
  \]
  \[
  \Rightarrow \text{else } \lambda_u \leftarrow \lambda_{\text{next}}, \text{ Go to step 2.}
  \]

Thus $\sum_i T^*_i(\lambda)$ is made successively closer to $T_{\text{max}}$ and finally we obtain the expected $\lambda^*$. With $\lambda^*$, we perform DP for the last time and obtain the optimal shortest path. The values of $\{S, N_{\text{max}}, A_{i,q,n} C_{i,q,n}\}$ corresponding to the shortest path are just the optimal parameter values for source coding and transmitting the $i$th summary frame.

2.5.2 Implementation Considerations

From the above analysis, we can say that problem (2.5) is converted into a graph theoretic problem of finding the shortest path in a directed acyclic graph (DAG) [73]. The computational complexity of the above algorithm is $O(N \times |U|^L)$, with $|U|$ denoting the cardinality of $U$, which depends on the number of the optional values of parameters $\{S, N_{\text{max}}, A, C\}$, but is still much more efficient than the exponential computational complexity of an exhaustive search algorithm. Clearly for cases with smaller $L$, the complexity is quite practical to perform the optimization. On the other hand, for larger $L$, the complexity can be limited by reducing the cardinality of $U$. The practical solution would be an engineering decision and tradeoff between the computational capability and optimality of the solution. For storage
issue, it is important to emphasize that the problem formulation and the proposed solution are quite generic and flexible for devices with various storage and computational capabilities. For the transmitter with a buffer size that only allows to store some portion of the video clip, the clip has to be divided into a number of segments and problem (2.5) is solved for each segment. In such cases, although the solution is not fully optimal for the video clip, the optimization would still bring sufficient gains compared to those without optimization.

2.6 Experimental Results

In this section, experiments are designed using H.264/AVC JM 10.2 for the video clip called "Glasgow", which is a typical test clip. For comparison, we summarize the first 300 frames into 30 and 60 summary frames, respectively. The case with more summary frames means higher sampling rate thus less distortion. To simplify the problem, we compress the
summary by choosing different QP (quantization step size), and we consider 10 possible QPs (5, 10, 15, 20, 25, 30, 35, 40, 45, 50). According to each QP, the frames have different rates and distortion values. The video summary is coded with intra-coding mode for each summary frame due to the less correlation between neighbor frames. In addition, without loss of generality, we consider the case of $L = 2$, in other words, we impose the constraint $\max(G_i, G_{i-1}) = 1$ ($i \in [0, m - 1]$) which needs to be guaranteed by the link adaptation. Besides parameters to be optimized, we assume fixed channel transmission rate $r = 6 \times 10^6$ symbols/second and fixed round trip time $T_{RTT} = 100$ milliseconds in our experiment.

Figure 2.5 and Figure 2.6 are comparisons between QP adaptation and No QP adaptation. In both figures, the average channel SNR $\bar{\gamma}$ is 25dB and we fix $P_L = 10^{-2}$ and $N_{\max} = 3$. $P_L$ is the target probability that $L = 2$ consecutive frames are being lost simultaneously, which should be small enough to approximately satisfy the constraint $\max(G_i, G_{i-1}) = 1$. In Figure 2.5 where the total summary frame number is 30, the square
nodes show the distortion-delay pairs when the video summary is source coded with the labelled QPs. The ‘v’ nodes refer to the distortion-delay budget pairs with QP adaptation when delay budget is set equal to the delay time that the corresponding labelled single QP takes to transmit the video summary. We can observe that QP adaptation, i.e., the proposed cross-layer framework, has much distortion gain up to 6.2% over fixed QP video transmission when the delay is small. In the case of summary frame number equal to 60 as in Figure 2.6, much more significant distortion gain up to 12% can be obtained in small delay regions.

Figure 2.7 shows the distortion vs. SNR comparisons between QP adaptation and QP=50 with different prescribed maximum transmission number for ARQ. Due to link adaptation performed by ARQ and AMC in a cross-layer fashion, both QP adaptation and QP=50 have a stable distortion level along all SNR values. Of course here the delay difference of different schemes are not considered. We also notice that for either QP adaptation
or QP=50, $N_{max} = 3$ has better performance than $N_{max} = 1$. This is because the case with $N_{max} = 3$ can achieve lower actual PER than with $N_{max} = 1$ even though they both aim to guarantee $P_L = 10^{-2}$. The same conclusion goes to Figure 2.8 where the total summary frame number is 60.

Different distortion vs. delay budget with different $P_L$ ($L=2$ in this experiment) is shown in Figure 2.9. We observe that there is a large distortion-delay difference between $P_L = 10^{-1}$ and $P_L = 10^{-2}$. Once $P_L$ achieves $10^{-2}$, there is no big distortion vs. delay difference even though $N_{max}$ is different. However, in the two cases of with different $P_L$ and same $N_{max} = 3$, the difference in distortion vs. delay is marginal. This is because with larger $N_{max}$, the actual PER is much lower than $P_L$. From this figure, we can conclude that the maximum transmission number impacts much the video transmission quality with our cross-layer optimization framework. With same delay budget, a larger allowed maximum transmission number leads to better video transmission quality.
Figure 2.9: Comparisons of distortion vs. delay budget with different $P_L$.

Figure 2.10: Distortion vs. delay budget comparison with different summary frame number.
Figure 2.10 shows the distortion vs. delay budget with our proposed framework when the summary frame number is 30 and 60 respectively both with $\gamma = 25$dB. With the same delay budget, the case with 60 frames has better performance than the case with 30 frames. This is because with higher sampling rate (that is, using 60 summary frames instead of 30), the similarity and correlation between neighbor summary frames have increased. Therefore, the distortion caused by losing one frame is reduced in this case because the lost frame would be concealed by its neighbor summary frame with higher similarity.

2.7 Summary

In this chapter, we have proposed an optimization framework for delivering video summary frames over wireless networks in a cross-layer fashion. The proposed framework seamlessly integrates the source coding at the application layer, ARQ at the data link layer and adaptive modulation and coding schemes at the physical layer. Within the delay-distortion theoretical framework, all major parameters at each layers are jointly optimized in a way to achieve the best video quality while satisfy the delay budget imposed by the video summary frames. Simulation results show that the proposed optimization framework can achieve more than 10% distortion gain, especially when the delay budget is small.
Chapter 3

Cross-Layer Optimized Routing for Video Streaming over Multi-Hop Wireless Networks

3.1 Introduction

In recent years, there has been an increasing demand for real-time video communication services, such as video telephony, video conferencing, video games and mobile TV broadcasting. These video applications are promoted by two facts: one is the pervasive use of computing devices, such as laptop computers, PDAs, smart phones, automotive computing devices, and wearable computers; and the other is the fast-growing deployment of multi-hop wireless networks to connect these computing devices. However, transmitting video over multi-hop wireless networks encounters many challenges, such as unreliable link quality due to multi-path fading and shadowing, signal interference among nodes, and dynamic connectivity outages. The routing issue significantly affects the end-to-end quality-of-service (QoS) of video applications, raising many questions, such as how to find the optimal path which can maximize the received video quality under stringent delay constraints and how to dynamically and adaptively determine the optimal path which can meet a required QoS and achieve an efficient network resource utilization with time-varying network conditions.

Traditional network-centric routing approaches relying on simple metrics such as hop count, average packet delay, or average packet loss rate fail to achieve the best perceived video quality. For example, the minimum hop-count metric arbitrarily chooses one from different paths of the same minimum length, despite the possible large variance in terms of throughput or packet loss rate existing among those paths. Moreover, the minimum average-packet-delay routing metric may fail due to the fact that the path with the minimum average-packet-delay does not necessarily lead to the minimum video distortion if there are
multiple paths all satisfying the required packet delay deadline. The minimum average-packet-loss-rate routing metric chooses the path with the minimum end-to-end packet loss rate, while ignoring the significant impact of the packetization scheme at the source on the perceived video quality, as well as ignoring the fact that not all the bits of a coded video bitstream are of equal importance in determining the perceived video quality. In recent years, there has been a lot of work on dealing with the multi-criteria routing problem subject to various constraints. A typical formulation to address the QoS routing problem is to choose a network-oriented optimization metric (e.g., maximize the end-to-end path bandwidth) subject to one or more constraints (e.g., delay and packet loss) [74–76]. While these metrics can be directly related to video quality through the rate-distortion theory, they do not consider the impact of error concealment on the perceived video quality, since dependencies among packets are introduced by concealment.

From the point of view of end users, the end-to-end video quality is the most straightforward and reasonable utility function in any optimization framework for video streaming. So far, a number of cross-layer techniques have been proposed to address the routing problem for video transmission over multi-hop wireless networks to maximize the received video quality. In [77], a set of pre-allocated paths is assumed over an overlay network. A framework which considers the path selection, along with the retransmission strategy and the PHY layer transmission scheme, is proposed to maximize the expected video distortion reduction at the application layer. In [22], a distributed Bellman-Ford-like routing algorithm for multi-user video streaming is developed to maximize the expected received video quality based on priority queuing analysis. The expected received video quality is modeled by a rate-distortion model which is a function of a set of parameters used by different video priority classes. In [23, 78], multi-path routing algorithms are developed for video multi-path delivery by utilizing path diversity. A set of paths are determined, one for each video stream/description, such that the received video distortion is minimized. In all aforementioned work, although the application-centric utilities such as video distortion or video
distortion reduction were adopted as routing metrics, they are either pre-calculated \[22,77\] or pre-generated from video distortion-rate models \[23,78,79\], without considering the impacts of the dynamic nature of video coding and error concealment strategies on routing path selection. In fact, most existing works on video routing focus on video streaming applications, where pre-coded data is used. However, pre-encoded video streaming does not allow for dynamic optimal routing path selection adaptive to video compression for real-time video applications. Therefore, to the best of our knowledge, little research has been done on integrating online video coding with dynamic network routing path selection to achieve the best perceived video quality.

In this work, the motivation of integrating application-layer video processing with routing is based on the following two observations. First, the selection of video coding parameters determines the rate-distortion performance of the codec, thus influencing the calculation of the optimal path if the expected distortion is used as the routing metric. For example, different quantization parameters (QPs) used by the source codec lead to different compression-induced distortions. The smaller the adopted QP value, the smaller the resulting compression-induced distortion. On the other hand, different QP values may lead to different transmission-induced distortions. Specifically, smaller QP values will generate larger video packets. Under the same channel conditions and with the same bit error rates (BER), larger packet sizes generally lead to higher packet loss rates, resulting in larger transmission-induced distortions. Thus, the optimal QP value selection is to find the best trade-off between the compression-induced distortion and the transmission-induced distortion. Similarly, such trade-off also exists in finding the optimal choice of other video coding parameters (e.g., intra/inter coding modes, prediction modes in intra/inter coding). Therefore, the optimal path calculation is affected by the selection of the video coding parameters in terms of minimizing the received video distortion. Second, when the network is in bad state (for example, with a median-to-heavy traffic load), no path exists to successfully deliver the video packets within the predefined frame decoding deadline. Then, video frames
should be coarsely coded into a number of short packets (or just skipped [1]) to fit in the limited channel capacity. In other words, network load can be significantly reduced by sending a smaller number of short packets over the network by adjusting the video coding parameters. Similarly, when the network is in good state, there may exist several good paths in it. To fully utilize the network resources, video coding can be tuned such that video frames are finely quantized, coded and transmitted. Although there is a lot of work on developing and optimizing the video coding techniques for wireless multimedia applications, for example, joint source and channel coding techniques (JSCC) [80], video coding optimization in existing works was purely based on the underlying end-to-end network information, such as packet loss rate, throughput, or delay performance.

In this chapter, we develop a routing algorithm by utilizing the user-received video quality as the routing metric, where packets are routed with considerations of higher-layer video processing to find the optimal path for the maximum received video quality. Specifically, path selection and video coding are jointly optimized to adapt to the time-varying network conditions, while ensuring that the end-to-end delay constraint is satisfied. Due to the difficulty in computing the actual video quality perceived by the end users, the received video quality is evaluated as the expected end-to-end distortion. The expected distortion is accurately calculated in real-time at the source node by taking all related parameters into account, such as source codec parameters (e.g., quantization, packetization, and error concealment) and network parameters (e.g., throughput and delay). Therefore, the multi-hop routing problem is formulated as the minimization of the expected end-to-end video distortion constrained by a predefined video packet delay deadline imposed by the video playback streaming application. We design a cross-layer controller at the source node to optimize the performance of the entire system. Compared to existing systems, the novelty of this work lies in: (i) proposing an application-centric routing algorithm for real-time video streaming based on an accurate video quality metric. Unlike network-centric routing metrics, such as, hop count, average delay or average success probability of packet transmission, the expected
video distortion cannot be calculated either additively or multiplicatively in a hop-by-hop fashion, due to error concealment. Therefore, the proposed application-centric routing metric is calculated on-the-fly in the process of routing; (ii) proposing a cross-layer optimization framework which integrates the quality-driven routing with online video coding for real-time video streaming over multi-hop wireless networks.

The remainder of this chapter is organized as follows. In Section 3.2, we formulate a cross-layer optimization problem for the routing issue of video transmission over multi-hop wireless networks. Section 3.3 presents a dynamic programming solution for the optimization problem. The algorithm for determining the optimal path for each packet is discussed in detail in Section 3.4. In Section 3.5, both the convergence and the computational complexity of the proposed framework are analyzed. Section 3.6 shows experimental results and Section 3.7 concludes this chapter.

3.2 Problem Description

In this section, we first describe the system model of the proposed quality-driven routing framework over multi-hop wireless networks. Then, we present the proposed framework of the minimization of the total expected video distortion of a video frame under a packet delay constraint by jointly optimizing the routing path selection and the video encoding parameters.

3.2.1 Proposed System Model

It is important to point out that we mainly focus on multi-hop wireless mesh networks where a collection of wireless nodes are configured to form a network without the aid of any established infrastructure. The system model of the proposed application-centric routing for video transmission over such multi-hop wireless networks is shown in Figure 3.1. In this work, we model a multi-hop wireless network as a directed acyclic graph (DAG) \( G(V, E) \), where \( V \) is the set of vertices representing wireless nodes and \( E \) is the set of arcs representing
Figure 3.1: The system model of the proposed application-centric routing for video transmission over multi-hop wireless networks. Here, $s$ is the source node and $t$ the destination node. All of the dotted arrowed lines and the solid arrowed lines in the network refer to the available network connectivity obtained from a proactive routing protocol such as optimized link state routing (OLSR) [81]; the solid arrowed lines connecting $s$ with $t$ refer to the optimal routing path calculated by the proposed application-centric routing approach.

directed wireless links. We characterize a link $(u, v) \in \mathcal{E}$ between nodes $u$ and $v$ by (1) $\lambda^u$: packet arrival rate at the starting node $u$, (2) $\gamma^{(u,v)}$: Signal-to-Interference-Noise-Ratio (SINR) of link $(u, v)$, (3) $P^{(u,v)}$: packet loss rate on the link $(u, v)$, and (4) $T^{(u,v)}$: packet delay on the link $(u, v)$. The link SINR information can easily be extracted from existing wireless network standards [82, 83]. The calculations of $P^{(u,v)}$ and $T^{(u,v)}$ will be discussed in more detail in Section 3.4.1.

In this work, as shown in Figure 3.1, we design a cross-layer controller at the source node to provide the following functionalities: 1) interact with each layer and obtain the corresponding managerial information, such as the expected video distortion from the encoder
and the network conditions from lower layers; 2) dynamically determine the optimal routing path and the corresponding optimal values of control variables residing in various layers. To calculate the optimal path for each packet, the global network topology and link state information such as link SINRs and average packet arrival rates at intermediate nodes are assumed to be available to the controller. We also assume that the mesh network topology and link status are fixed over the duration of one video frame. From the standpoint of implementation, a link cache/database can be set up at the source node to store the global network topology information and link state information, which can be fed back from other nodes. To make the link state information available to the controller, these information can be disseminated to the source node via a hop-to-hop feedback mechanism at frequent intervals or when the incurred change in network parameters, such as SINRs and packet arrival rates, is larger than a preset threshold. For example, link status information can be piggybacked on the acknowledgement packets to utilize the bandwidth more efficiently. However, in certain cases, feedback from remote hops may arrive with an intolerable delay, and may be deemed inaccurate and unreliable due to the rapidly changing network conditions. For such cases, we may apply the concept of information retrieval horizon [77,84] to the source node, where network information within the horizon of the source node is deemed reliable and can be received in a timely manner, while information beyond the horizon can only be theoretically estimated based on average or previous measurements. Finally, we propose that the quality-driven routing algorithm will be built on top of a proactive routing protocol such as optimized link state routing (OLSR) [81]. Due to its proactive nature, OLSR maintains up-to-date global network topology information in its link-state database. Once there are videos to be transmitted, the controller will retrieve the network information to perform the joint optimization of video encoding and path routing.
Packet Delay Deadline

Let $\Pi = \{\pi_1, \pi_2, \cdots, \pi_I\}$ be the set of $I$ packets that compose the current video frame to be transmitted. Each packet $\pi_i$ is independently decodable and is generated from a slice of the video frame. In the latest video coding standards, such as H.264 [1], a slice could be either as small as a group of macroblocks or as large as the entire video frame. Each slice header acts as a resynchronization marker which allows for the slices to be independently decodable and be decoded correctly at the decoder even when they are transported out of order. To enable a good balance between error robustness and compression efficiency [26], in the rest of this chapter, we assume that the encoded bits of the $i$th slice in the current frame are packetized into packet $\pi_i$. Thus, packet $\pi_i$ and slice $i$ will be used interchangeably in this chapter unless otherwise specified. To provide a smooth video display experience to end users, each frame is associated with a frame decoding deadline $T^{budget}$. We assume that the playback delay (e.g., caused by buffering at the end user) is much smaller than the delay incurred by packets during transmission and queueing along a multi-hop path. Thus, in real-time video communications, the average frame decoding deadline $T^{budget}$ is linked with the frame rate $f$ as $T^{budget} \approx \frac{1}{f}$ [85]. The frame decoding deadline $T^{budget}$ indicates that all the packets needed to decode a frame must be received at the decoder buffer prior to the playback time of that frame, meaning that a delay deadline is imposed on the transmission of each packet composing the frame by $T^{budget}$ [26, 77, 85], i.e.,

$$\max\{T_1, T_2, \cdots, T_I\} \leq T^{budget},$$

(3.1)

where $T_i$ is the end-to-end delay of packet $\pi_i$ transmitted from the source node $s$ to the destination node $t$, which will be discussed in more detail in Section 3.4.1.

Expected End-to-End Distortion

It is challenging to evaluate the received video distortion in multi-hop wireless networks due to the complex network characteristics. Several algorithms for calculating the expected
distortion in single-hop networks have been recently proposed [86–88]. In this work, we employ the ROPE algorithm [88] and extend it to multi-hop networks. The accuracy of ROPE in end-to-end distortion estimation is attributed to its ability to calculate the first and second moments of the decoder reconstructed pixels. To deal with the cross correlation terms in the second moment calculation caused by sub-pixel prediction in H.264/AVC [1] in our experiments, we use the cross correlation approximation method introduced in [89] to calculate the expected end-to-end distortion.

A robust error concealment technique helps avoid significant visible error in the reconstructed frames at the decoder. Given the importance of error concealment in determining the final decoded quality of the transmitted video, we assume that the error concealment scheme is known at both the source node and the destination node. In other words, a protocol in which the error concealment is known to both the controller and the decoder can potentially be highly beneficial in providing significant performance improvement through the proposed application-centric routing approach. In this work, we consider a simple but efficient temporal concealment scheme: a lost macroblock is concealed using the median motion vector candidate of its received neighboring macroblocks (the top-left, top, and top-right) in the preceding row of macroblocks. The candidate motion vector of a macroblock is defined as the median motion vector of all $4 \times 4$ blocks in the macroblock. If the preceding row of macroblocks is also lost, then the estimated motion vector is set to zero and the macroblock in the same spatial location in the previously reconstructed frame is used to conceal the current loss. Note that this concealment strategy is employed both in the controller and at the decoder.

It is important to point out that, although some straightforward error concealment strategies do not cause packet dependencies, as a generic framework, the more complicated scenario is considered here as a superset for the simpler cases. Therefore, given the dependencies introduced by the above error concealment scheme, the expected distortion of
slice/packet $\pi_i$ can be calculated at the encoder as

$$E[D_i] = (1 - P_i)E[D^R_i] + P_i(1 - P_{i-1})E[D^{LR}_i] + P_iP_{i-1}E[D^{LL}_i],$$  \hspace{1cm} (3.2)$$

where $P_i$ is the loss probability of packet $\pi_i$, which will be discussed in more detail in Section 3.4.1; $E[D^R_i]$ is the expected distortion of packet $\pi_i$ if received, and $[D^{LR}_i]$ and $[D^{LL}_i]$ are respectively the expected distortion of the lost packet $\pi_i$ after concealment when packet $\pi_{i-1}$ is received or lost. Based on the additive distortion measure of ROPE, the expected distortion of the whole video frame, denoted by $E[D]$, can be written as

$$E[D] = \sum_{i=1}^{I} E[D_i].$$  \hspace{1cm} (3.3)$$

### 3.2.2 Problem Formulation

Let $S_i \in S$ be the source coding parameters for the $i$th slice of the current frame, where $S$ is the set of all admissible values of $S_i$ and $|S| = J$. Let $\xi_i$ denote the transmission path of packet $\pi_i$. Considering that both $E[D_i]$ and $T_i$ depend on $S_i$ and $\xi_i$, the problem at hand is to choose the optimal transmission path and coding parameter values for all the slices of the current video frame so as to minimize the total expected distortion under the packet delay constraint (3.1) in a lossy multi-hop wireless network, i.e.,

$$\min_{[S_1, \xi_1, \ldots, S_I, \xi_I]} \sum_{i=1}^{I} E[D_i]$$

$$\text{s.t. : } \max\{T_1, \ldots, T_I\} \leq T^{\text{budget}}.$$  \hspace{1cm} (3.4)$$

It is worth noting that the optimization is performed one frame at a time. Nonetheless, this framework can potentially be improved by optimizing the routing and video encoding over multiple buffered frames, which can integrate the packet dependencies caused by both error concealment and prediction [1] in source coding into the optimization framework. Such a scheme, however, would lead to a considerably higher computational complexity.
3.3 Solution Procedures

The solution procedure of the optimization problem in (3.4) consists of the following two steps. First, the controller calculates the optimal path for each possible packet corresponding to each slice coded by different coding options, based on the network topology and link status information. Then, the controller performs global optimization for a group of slices of one video frame, finding the optimal path for each slice. Due to the dependencies between slices introduced by error concealment, we use dynamic programming to find the optimal coding parameters and transmission paths for a group of slices composing a frame in a “trellis” graph. In this section, we first present the global optimization performed by the controller over each group of slices with the assumption that the optimal path for each possible packet has already been computed. The algorithm of how to calculate the optimal path for a given packet by the controller will be discussed in Section 3.4.

Given the dependencies among slices introduced by the decoder concealment strategy, the problem of jointly selecting the pairs of coding parameters and transmission paths for a group of slices can be described by the “trellis” depicted in Figure 3.2. By each coding option $S_i^j \in S \ (j = 1, 2, \cdots, J)$, the coded bits of the $i$th slice will be packetized into a packet denoted by $\pi_i^j$. In other words, the $i$th slice can be compressed into $J$ different versions of packets $\{\pi_i^1, \pi_i^2, \cdots, \pi_i^J\}$ using the $J$ possible coding options. We can send the $i$th slice by transmitting any of the $J$ packets $\{\pi_i^1, \pi_i^2, \cdots, \pi_i^J\}$. Moreover, each packet $\pi_i^j$ has its optimal transmission path $\xi_i^j$ based on the routing algorithm in Section 3.4. However, with the expected distortion as the routing metric, different packets may have different transmission paths and lead to different distortion of the $i$th slice as shown in Figure 3.2, where the weight $E[D_i^{j,j'}]$ of each edge, calculated by Eq. (3.2), represents the resulting expected distortion of the $i$th slice by transmitting packet $\pi_i^j$ over path $\xi_i^j$ with the consideration of its dependency on packet $\pi_{i-1}^{j'}$.

For simplicity, let us denote by $\theta_i = \{\xi_i, S_i\}$ the decision vector for the $i$th slice. Different
error concealment strategies introduce different types of dependencies among slices. For example, given the packetization scheme and the error concealment scheme adopted in this work, that is, a slice is defined as a row of macroblocks and the concealment algorithm uses the motion vectors of the macroblocks above to conceal the lost macroblock, the calculation of the expected distortion of the current slice depends on its previous slices. Without loss of generality, we assume that the current slice depends on its previous $z$ slices ($z \geq 0$) by using a certain concealment strategy. Therefore, the optimization goal in Eq. (3.4) becomes

$$\min_{\theta_1, \ldots, \theta_I} \sum_{i=1}^{I} E[D_i](\theta_{i-z}, \theta_{i-z+1}, \ldots, \theta_i), \quad i - z > 0,$$

(3.5)

where $E[D_i](\theta_{i-z}, \theta_{i-z+1}, \ldots, \theta_i)$ represents the expected distortion of the $i$th slice, which depends on the $z+1$ decision vectors $\{\theta_{i-z}, \theta_{i-z+1}, \ldots, \theta_i\}$ under the packet delay deadline.

To solve the optimization equation (3.5), we define a cost function $\Psi_y(\theta_{y-z+1}, \ldots, \theta_y)$, which represents the minimum expected distortion up to and including the $y$th slice, given

![Trellis for the joint source coding and path selection problem for a group of slices.](image)
that \{\theta_{y-z+1}, \ldots, \theta_y\} are the decision vectors for the slices \(y - z + 1\) to \(y\), i.e.,

\[
\Psi_y^*(\theta_{y-z+1}, \ldots, \theta_y) = \min_{\theta_1, \ldots, \theta_y} \sum_{i=1}^{y} E[D_i](\theta_{i-z}, \theta_{i-z+1}, \ldots, \theta_i).
\] (3.6)

Therefore, \(\Psi_f^*(\theta_{1-z+1}, \ldots, \theta_f)\) represents the minimum total distortion for all the slices of the current frame. Clearly, solving (3.5) is equivalent to solving

\[
\min_{\theta_{1-z+1}, \ldots, \theta_f} \Psi_f^*(\theta_{1-z+1}, \ldots, \theta_f).
\] (3.7)

Following Eq. (3.6), the cost function \(\Psi_y^*(\theta_{y-z+1}, \ldots, \theta_y)\) can be rewritten as

\[
\Psi_y^*(\theta_{y-z+1}, \ldots, \theta_y) = \min_{\theta_{y-z}} \left\{ \min_{\theta_{1-z}, \ldots, \theta_{y-z-1}} \left( \sum_{i=1}^{y-1} E[D_i](\theta_{i-z}, \ldots, \theta_i) + E[D_y](\theta_{y-z}, \ldots, \theta_y) \right) \right\}
\]

\[
= \min_{\theta_{y-z}} \left\{ \min_{\theta_{1-z}, \ldots, \theta_{y-z-1}} \sum_{i=1}^{y-1} E[D_i](\theta_{i-z}, \ldots, \theta_i) + E[D_y](\theta_{y-z}, \ldots, \theta_y) \right\}
\]

\[
= \min_{\theta_{y-z}} \left\{ \Psi_y^*_{y-1}(\theta_{y-z}, \ldots, \theta_{y-1}) + E[D_y](\theta_{y-z}, \theta_{y-z+1}, \ldots, \theta_y) \right\}.
\] (3.8)

The recursive representation of the cost function \(\Psi_y^*(\theta_{y-z+1}, \ldots, \theta_y)\) discussed earlier makes the future step of the optimization process independent from its past steps, forming a dynamic programming problem, which can be further converted into a graphical problem of finding the shortest path in the directed acyclic graph of Figure 3.2 [90]. The complete proposed optimization algorithm is summarized in Table 3.1.

### 3.4 Proposed Routing Algorithm

In Section 3.3, we mentioned that each packet \(\pi_i^j\) is corresponding to the path \(\xi_i^j\), over which the expected end-to-end distortion of the \(i\)th slice coded by \(S_i^j\) will be minimized. However, there are several challenges in determining such a path by the controller at the source node in lossy transmission environment. First, since the controller uses the expected distortion as the routing metric to find path \(\xi_i^j\) for packet \(\pi_i^j\), it is necessary to perform accurate distortion estimation at the source node by considering all the impacts from signal
Table 3.1: Proposed Optimization Algorithm

1) For each slice $i$,
   For each coding option $S_i$,
   Perform the routing algorithm described in Section 3.4, and find the optimal path $\xi^j_i$ for packet $\pi^j_i$ by using the routing metric defined as the expected slice distortion under a certain error concealment scheme with consideration of the packet delay deadline constraint;

2) Perform the DP optimization described in Section 3.3 to find the optimal path and coding parameters values for all slices of one frame.

corruption, packet delay deadline expiration, and error concealment on expected video distortion. Second, to evaluate the effects of signal corruption and packet delay deadline expiration on the distortion of the $i$th slice, the controller needs to calculate the average loss probability due to channel fading and the delay deadline expiration probability incurred by packet $\pi^j_i$ at intermediate nodes, based on the feedbacks from these nodes. To overcome the aforementioned challenges, we propose a routing algorithm to determine the optimal path $\xi^j_i$ for each packet $\pi^j_i$.

3.4.1 Routing Metric: Packet-Delay-Deadline-Aware Expected Distortion

To evaluate the expected distortion with consideration of packet delay deadline, we need to first consider packet loss probabilities and delay performance at intermediate nodes. Although there exist numerous models in the literature for calculating packet loss probabilities and delay performance in multi-hop wireless networks, none of them has taken the expected video distortion into consideration. In the following, we will give a brief discussion on these existing models and focus on how to integrate the packet delay constraint into the estimation of the expected video distortion.
Packet Loss Probability

In wireless environments, as shown in Figure 3.3, the transmission of packet $\pi_i^j$ over each hop of the network can be modeled as follows: packet $\pi_i^j$ arrives at node $u$, waits in the queue of node $u$, and is transmitted/served over link $(u, v)$. If packet $\pi_i^j$ gets lost during transmission, it will be retransmitted until it is either successfully received or discarded because its delay deadline $T_{\text{budget}}$ was exceeded. During the above period, the total packet loss probability $P_{(u,v)}^{(u,v)}$ incurred by packet $\pi_i^j$ mainly consists of two parts: 1) the probability of packet drop $p_{u}^{i,j}$ due to delay deadline expiration when queueing at node $u$, 2) the probability of packet loss $p_{(u,v)}^{(u,v),\text{err}}$ over link $(u, v)$ which is mainly determined by packet error probability $p_{(u,v),\text{err}}^{(u,v)}$ due to signal fading over link $(u, v)$. In a contention-based wireless access network (e.g., a multi-hop 802.11a/e wireless network), $p_{(u,v)}^{(u,v)}$ also includes packet collision probability $p_{(u,v),\text{col}}^{(u,v)}$ due to contention access to the medium. Therefore, without loss of generality, packet loss probability $p_{(u,v)}^{(u,v)}$ over link $(u, v)$ can be written as $p_{(u,v)}^{(u,v)} = 1 - (1 - p_{(u,v),\text{err}}^{(u,v)})(1 - p_{(u,v),\text{col}}^{(u,v)})$. To deal with $p_{(u,v)}^{(u,v)}$ over link $(u, v)$, retransmission mechanisms such as the selective repeat algorithm [91] are commonly used at node $u$ within the packet delay deadline. However, retransmissions of packet $\pi_i^j$ make the packet drop probability $p_{u}^{i,j}$ at the node $u$ implicitly depend on the packet loss probability $p_{(u,v)}^{(u,v)}$ over link $(u, v)$. As a result, the total packet loss probability $P_{(u,v)}^{(u,v)}$ is mainly exhibited as $p_{u}^{i,j}$, i.e., $P_{(u,v)}^{(u,v)} \approx p_{u}^{i,j}$. Next, we will discuss how to calculate $p_{u}^{i,j}$ by the controller.

The packet error probability $p_{(u,v),\text{err}}^{(u,v)}$ depends on the specific channel conditions of link $(u, v)$ and the applied transmission schemes. We assume that each intermediate node implements a certain type of link adaptation scheme to maximize its outgoing link goodput. Specifically, as in [6, 27], node $u$ selects different modulation and channel coding schemes based on the link SINR information in transmitting packets over link $(u, v)$. The packet error probability $p_{(u,v),\text{err}}^{(u,v)}$ for packet $\pi_i^j$ can be modeled by the sigmoid function

$$
p_{(u,v),\text{err}}^{(u,v)} = \frac{1}{1 + e^{\gamma - \delta}}, \quad (3.9)
$$
The total packet loss probability $P_{i,j}^{(u,v)}$ over link $(u,v)$ mainly comprises of packet drop probability $p_{i,j}^{d}$ due to packet delay deadline expiration during queueing at node $u$ and the packet loss probability $p_{i,j}^{(u,v)}$ including error probability $p_{i,j}^{(u,v),err}$ due to signal fading over link $(u,v)$ and the packet collision probability $p_{i,j}^{(u,v),col}$ due to contention access to the medium.

where $\gamma$ is the channel status information (CSI) parameters in term of the detected link SINR, $\zeta$ and $\delta$ are constants corresponding to the used modulation and coding schemes for a given packet of length $L$ [22,92]. The packet collision probability $p_{i,j}^{(u,v),col}$ at steady state can be expressed as [93]

$$p_{i,j}^{(u,v),col} = 1 - (1 - \tau)^{x-1},$$

(3.10)

where $\tau$ is the probability that node $u$ transmits in a randomly chosen slot time, and $x$ is the number of contending nodes.

To evaluate the delay incurred by packets at intermediate nodes, we first need to determine the available throughput for transmitting these packets. Let $w^{(u,v)}$ be the guaranteed bandwidth of link $(u,v)$, e.g., by adopting the Hybrid Coordination Function (HCF) Controlled Channel Access (HCCA) mechanisms of IEEE 802.11e [82]) or the TDMA scheme adopted in WiMAX. Then, the available throughput over link $(u,v)$ is directly determined by the transmission rate $R_{i,j}^{(u,v)}(w^{(u,v)})$ of the corresponding modulation and coding schemes. Thus, the effective transmission rate (goodput) [27] is given by

$$R_{i,j}^{(u,v)} = \frac{R_{i,j}^{(u,v)}(w^{(u,v)})}{1 + e^{-\zeta(\gamma-\delta)}}.$$
for these impacts on the throughput performance, the effective transmission rate can be weighted as

\[ R_{i,j}^{(u,v)} = \frac{\eta \cdot R_{i,j}^{(u,v)}(u,v)}{1 + e^{-c(\gamma - \delta)}}, \tag{3.12} \]

where \( \eta \) is the normalized system throughput defined as the fraction of time the channel is used to successfully transmit payload bits. For example, according to [93], \( \eta \) for 802.11 protocol depends on the wireless channel characteristics (i.e. the propagation delay), the network specification and system parameters (i.e. the minimum and maximum contention window sizes, the average packet size, and the number of contending nodes), and the adopted access mechanisms (i.e., the basic access mechanism, the RTS/CTS mechanism or the hybrid mode of the two).

To derive the packet drop probability \( p_{i,j}^{(u,v)} \) due to the packet delay deadline expiration during the wait in the queue of node \( u \), we need to analyze the queueing model of packet \( \pi_i \) at that node. Let \( T_{i,j}^{u} \) be the current delay incurred by packet \( \pi_i \) when the packet arrives at node \( u \) and enters the queue of that node. The maximum retransmission limit \( \tau_{i,j}^{(u,v)} \) for packet \( \pi_i \) over link \((u, v)\) based on the delay deadline \( T_{\text{budget}} \) can be expressed as [94]

\[ \tau_{i,j}^{(u,v)} = \left\lfloor \frac{R_{i,j}^{(u,v)}(T_{\text{budget}} - T_{i,j}^{u})}{L} \right\rfloor - 1, \tag{3.13} \]

where \( \lfloor \cdot \rfloor \) is the floor operation. To evaluate the expected waiting time for packet \( \pi_i \) spent in the queue of node \( u \), we formulate the service time \( X_{i,j}^{(u,v)} \) for packet \( \pi_i \) over link \((u, v)\) as a geometric distribution [95]. Then the first and second moments of the service time \( X_{i,j}^{(u,v)} \) are given by

\[ E \left[ X_{i,j}^{(u,v)} \right] = \frac{L \left( 1 - (p_{i,j}^{(u,v)})^{\tau_{i,j}^{(u,v)} + 1} \right)}{R_{i,j}^{(u,v)}(1 - p_{i,j}^{(u,v)})} = \frac{L \left( 1 - (p_{i,j}^{(u,v)})^{\tau_{i,j}^{(u,v)} + 1} \right)}{R_{i,j}^{(u,v)}} \tag{3.14} \]

\[ E \left[ (X_{i,j}^{(u,v)})^2 \right] = \frac{L^2(1 + p_{i,j}^{(u,v)})}{(R_{i,j}^{(u,v)})^2(1 - p_{i,j}^{(u,v)})^2}. \tag{3.15} \]
We assume that the arrival traffic at each intermediate node $u$ is from various data sources through all its previous-hop nodes and is assumed to be a Poisson process. This approximation is reasonable if the number of intermediate nodes is large enough and the traffic in the network is relatively balanced. Without loss of generality, we model the queue in the intermediate node $u$ as a M/G/1 queue with an arrival rate $\lambda^u$. The values of $\lambda^u$ can be locally obtained at node $u$ by counting and averaging the total number of incoming packets over a given period of time. Then by queueing analysis [95] (Section 3.5), the average waiting time for packet $\pi^j_i$ at node $u$ can be expressed as

$$E\left[W_{i,j}^{(u,v)}\right] = \frac{\lambda^u E\left[(X_{i,j}^{(u,v)})^2\right]}{2(1 - \lambda^u E[X_{i,j}^{(u,v)}])}. \quad (3.16)$$

Based on the expected waiting time, the probability of packet $\pi^j_i$ being dropped due to the expiration of the packet delay deadline can be calculated by the tail distribution of the waiting time [96]

$$p^u_{i,j} = \text{Prob}\left( E[W_{i,j}^{(u,v)}] + T^u_{i,j} > (T_{\text{budget}}) \right)$$

$$= \lambda^u E[X_{i,j}^{(u,v)}]\exp\left(-\frac{(T_{\text{budget}} - T^u_{i,j})\lambda^u E[X_{i,j}^{(u,v)}]}{E[W_{i,j}^{(u,v)}]}\right). \quad (3.17)$$

Now we can determine the loss probability of packet $\pi^j_i$ traversing a sequence of nodes. Let $\hat{\xi}^v_{i,j}$ be a path from source node $s$ to node $v$ for packet $\pi^j_i$. The loss probability for packet $\pi^j_i$ traversing path $\hat{\xi}^v_{i,j}$ is

$$P^v_{i,j} = 1 - \prod_{(u,v)\in\hat{\xi}^v_{i,j}} (1 - P^{(u,v)}_{i,j}). \quad (3.18)$$

With Equations (3.2) and (3.18), the encoder can calculate the expected distortion on-the-fly which is delivered to the controller to determine the optimal path for packet $\pi^j_i$. 
Packet Delay

Let $T^v_{i,j}$ be the total delay for packet $\pi^j_i$ arriving at node $v$ along path $\xi^v_{i,j}$. Given node $u$ is the previous-hop node of node $v$, we have

$$T^v_{i,j} = T^u_{i,j} + T^{(u,v)}_{i,j},$$

where $T^{(u,v)}_{i,j} := E[W^{(u,v)}_{i,j}] + E[X^{(u,v)}_{i,j}]$ is the total delay incurred by packet $\pi^j_i$ over link $(u, v)$.

3.4.2 Optimal Path Selection for Individual Packets

It is worth noting that in multi-hop wireless networks, to increase the network resilience and robustness against potential problems such as node failures and path failures due to temporary obstacles or external radio interference, network nodes might be connected to each other with redundant paths between each pair of nodes. For example, the emerging wireless mesh networks are densely-connected networks [97]. In such densely-connected networks, exhaustive search to find the optimal path is not feasible due to its exponential computational overhead.

To decrease the computational complexity, contrary to the distributed Bellman-Ford [95] algorithm-like routing approach in [22], we propose a centralized Dijkstra-based [95] labelling algorithm with which the controller calculates the optimal path for a given packet based on the feedback information such as link SINR and average packet arrival rate from all other nodes. However, the proposed algorithm is very different from the classical Dijkstra algorithm, where routing metrics (e.g., packet delay) for path selection are additive. In contrast, in the proposed routing algorithm, the routing metric for path selection is defined as the expected distortion of packet $\pi^j_i$ with considerations of the possible error concealment. Since error concealment is performed only at the destination node $t$, it is not possible to define a routing metric as the expected distortion of packet $\pi^j_i$ with/without the possible error concealment over a single hop. This is because when expected video quality is adopted
as a routing metric, the additive routing cost calculation is no longer feasible. Therefore, explicit link metrics (the expected distortion over each hop along a path) cannot be identified and defined. The expected distortion of packet $\pi_i^j$ at any node $v$ needs to be calculated on-the-fly by the encoder based on the feedback network information about the partial optimal path from source node $s$ to node $v$.

Here the optimal path for a packet is the path over which the transmitted packet has the minimum expected distortion under the given packet delay deadline. The labelling algorithm is summarized as follows:

- Each node is labeled with a quadruple $\{E[D_{i,j}^v|u], P_{i,j}^v, T_{i,j}^v, u\}$ where $E[D_{i,j}^v|u]$, $P_{i,j}^v$, $T_{i,j}^v$ are the expected distortion, the packet loss probability, and the packet delay incurred by packet $\pi_i^j$ traversing along the partial optimal path from source $s$ to node $v$ through the previous-hop node $u$. The motivation for keeping the values of packet loss probability and packet delay in a label is to speed up the calculation of routing path selection, which will be further explained in the later example. Distortion $E[D_{i,j}^v|u]$ at node $v$ is calculated by Eq. (3.2). Moreover, $E[D_{i,j}^v|u]$ becomes the edge weight (distortion) in Figure 3.2 when node $v$ is the destination node. In order to easily reconstruct the whole optimal path later, the quadruple at node $v$ also contains the previous-hop node $u$ through which the partial optimal path passes. Initially, no optimal path exists. Thus, all nodes are labeled with an infinite amount of expected distortion, packet loss probability, and packet delay.

- A label can be marked as either tentative or permanent. A node marked by a tentative label is defined as a tentative node. Likewise, a node marked by a permanent label is defined as a permanent node. As the routing algorithm proceeds, new paths may be found as well as the labels may be changed.

- When a label is found from all tentative labels with the minimum expected distortion of packet $\pi_i^j$, the optimal path for packet $\pi_i^j$ from source node $s$ to the corresponding
node carrying that label will be determined, then the label is marked as permanent and is kept unchanged thereafter. Then, this node becomes the current working node, meaning that the controller will start to calculate the expected distortions incurred by transmitting packet $\pi_i^j$ through all its next-hop nodes of the working node.

- Whenever the label of the destination node is marked as permanent, the optimal end-to-end routing path for packet $\pi_i^j$ will be determined.

Next, we will use the DAG-based network model shown in Figure 3.4 as an example of a multi-hop wireless network to explain the proposed labelling algorithm. As stated before, the network topology is assumed to be known by the controller. As shown in Figure 3.5, the flowchart of finding the optimal path for packet $\pi_i^j$ through the proposed labelling algorithm is described as follows:

Step (a) Initially, as shown in Figure 3.5 (a), no optimal path exists, so all nodes are labeled with an infinite amount of expected distortion, packet loss probability, and packet delay. All labels are tentative, which are marked with circles. Then, the controller starts to mark node $s$ as permanent, which is marked with a filled-in circle. The controller also sets node $s$ as the current working node, which is marked with $\Rightarrow$ and begins calculating the expected distortions if packet $\pi_i^j$ would be transmitted through next-hop nodes $a$ and $e$, respectively.

Step(b) In Figure 3.5 (b), once the expected distortion $E[D_{i,j}^a]$, the probability of loss $P_{i,j}^a$, and
Figure 3.5: The steps used in computing the optimal path for packet $\pi_j$ from source $s$ to destination $t$. The algorithm begins from step (a) and ends at step (f). The arrows indicate the working node. The optimal path is $s \rightarrow a \rightarrow c \rightarrow d \rightarrow f \rightarrow t$. 
the expected delay $T_{i,j}^{a|s}$ are calculated, node $a$ is relabeled by $\{E[D_{i,j}^{a|s}], P_{i,j}^{a|s}, T_{i,j}^{a|s}, s\}$. Similarly, node $e$ is relabeled by $\{E[D_{i,j}^{e|s}], P_{i,j}^{e|s}, T_{i,j}^{e|s}, s\}$. Then, the controller examines all the tentatively labeled nodes, which are $a$, $b$, $c$, $d$, $e$, $f$ and $t$ at this stage in the whole graph. The controller marks the one with the smallest expected distortion as permanent, which will become the new working node. In Figure 3.5 (b), assuming that $E[D_{i,j}^{a|s}] \leq E[D_{i,j}^{e|s}]$, thus node $a$ is marked as permanent and becomes the new working node. Then, the controller begins to calculate the expected distortions if packet $\pi _{i}^{j}$ would be transmitted along the paths $s \rightarrow a \rightarrow b$ and $s \rightarrow a \rightarrow c$, respectively.

**Step (c)** As mentioned earlier, the label of working node can be used to speed up the calculations of the labels of next-hop nodes. For example, when calculating the loss probability $P_{i,j}^{b|a}$ that packet $\pi _{i}^{j}$ would have at node $b$ via the path $s \rightarrow a \rightarrow b$, the controller can directly retrieve the value of $P_{i,j}^{a|s}$ stored in the label of working node $a$ and calculate $P_{i,j}^{b|a}$ as $P_{i,j}^{b|a} = 1 - (1 - P_{i,j}^{a|s})(1 - P_{i,j}^{(a,b)})$ in stead of recalculating $P_{i,j}^{a|s}$. Similarly, the delay $T_{i,j}^{b|a}$ at node $b$ can also be calculated as $T_{i,j}^{b|a} = T_{i,j}^{a|s} + T_{i,j}^{(a,b)}$ by using the stored $T_{i,j}^{a|s}$ in the label of working node $a$. As shown in Figure 3.5 (c), nodes $b$ and $c$ are relabeled by $\{E[D_{i,j}^{b|a}], P_{i,j}^{b|a}, T_{i,j}^{b|a}, a\}$ and $\{E[D_{i,j}^{c|a}], P_{i,j}^{c|a}, T_{i,j}^{c|a}, a\}$, respectively. Then, the controller compares the distortion values of all the left tentative-labeled nodes, which are $b$, $c$, $d$, $e$, $f$ and $t$ at this stage. Assuming that the label of node $c$ has the smallest expected distortion, node $c$ is marked as permanent and becomes the new working node. The controller then begins to calculate the expected distortion $\{E[D_{i,j}^{d|c}]\}$ that packet $\pi _{i}^{j}$ would get at node $d$ if it is transmitted along the path $s \rightarrow a \rightarrow c \rightarrow d$.

**Step (d)** Using the same procedure as discussed above, node $d$ is relabeled with $\{E[D_{i,j}^{d|c}], P_{i,j}^{d|c}, T_{i,j}^{d|c}, c\}$ as shown in Figure 3.5 (d). Then, the controller continues to search for the node which has the smallest expected distortion among all the other tentatively labeled nodes (at this stage, they are $b$, $d$, $e$, $f$ and $t$). Assuming that $E[D_{i,j}^{a|s}]$ is
the smallest among all the compared distortions, node e is marked as permanent and becomes the new working node. Then, the controller begins to calculate the expected distortions that packet $\pi^j_i$ would get at the two next-hop nodes $c$ and $f$ of node $e$ by being passed along the paths $s \rightarrow e \rightarrow c$ and $s \rightarrow e \rightarrow f$, respectively.

Step (e) In Figure 3.5 (e), for node $c$, let $E[D^{|c|}_c]$ be the expected distortion that packet $\pi^j_i$ would incur at node $c$ if it takes the path $s \rightarrow e \rightarrow c$. Assuming that $E[D^{|c|}_c] \geq E[D^{|a|}_c]$, then node $c$ will not be relabeled. For node $f$, the label needs to be updated with $\{E[D^{|e|}_f], P^f_i, T^f_i, e\}$. Then, as shown in Figure 3.5 (e), assuming that node $d$ is identified as having the smallest distortion among all unchecked tentative nodes (at this stage, they are $b$, $d$, $f$ and $t$). Thus, node $d$ is marked as permanent and becomes the new working node. The controller then begins to calculate the expected distortions that packet $\pi^j_i$ would incur at the two next-hop nodes $b$ and $f$ of node $d$ if the packet travels along the paths $s \rightarrow a \rightarrow c \rightarrow d \rightarrow b$ and $s \rightarrow a \rightarrow c \rightarrow d \rightarrow f$, respectively.

Step (f) In Figure 3.5 (f), for node $b$, let $E[D^{|d|}_b]$ be the expected distortion that packet $\pi^j_i$ would incur at node $b$ if it takes the path $s \rightarrow a \rightarrow c \rightarrow d \rightarrow b$. Assuming that $E[D^{|d|}_b] \geq E[D^{|a|}_b]$, then the label of node $b$ will not be updated. For node $f$, let $E[D^{|d|}_f]$ be the expected distortion that packet $\pi^j_i$ would incur at node $f$ if it takes the path $s \rightarrow a \rightarrow c \rightarrow d \rightarrow f$. Assuming that $E[D^{|d|}_f] \leq E[D^{|e|}_f]$, so node $f$ is relabeled by $\{E[D^{|d|}_f], P^f_i, T^f_i, d\}$. Then, the controller needs to find the node with the smallest labeled distortion among the unchecked tentatively labeled nodes (at this stage, they are $b$, $f$ and $t$). Assuming that the label of node $f$ has the smallest expected distortion, node $f$ is marked as permanent and becomes the new working node.

Step (g) Figure 3.5 (g) shows that the only next-hop node $t$ of node $f$ is relabeled with the expected distortion $E[D^{|f|}_t]$ that packet $\pi^j_i$ would incur at node $t$ if it takes the path $s \rightarrow a \rightarrow c \rightarrow d \rightarrow f \rightarrow t$. Then, the controller begins to find the next working node.
between the last two tentative nodes $b$ and $t$. Assuming that the label of node $b$ has a smaller expected distortion value than the label of node $t$ (i.e., $E[D_{i}^{b}] < E[D_{i}^{f}]$), thus, node $b$ is marked as permanent and becomes the new working node. Then, the controller begins to calculate the expected distortion $E[D_{i}^{b}]$ that packet $\pi_i^j$ would incur if it proceeds via the path $s \rightarrow a \rightarrow b \rightarrow t$.

**Step (h)** In Figure 3.5 (h), assuming that $E[D_{i}^{f}] \geq E[D_{i}^{f}]$, thus, node $f$ does not need to be relabeled. Then, node $f$ is the only tentative node and is marked as permanent. Once node $t$ is marked as permanent, the proposed routing algorithm stops. By retrieving all stored previous-hop nodes in the labels from destination node $t$ to source node $s$, the optimal end-to-end path can be reconstructed. In Figure 3.5 (h), the optimal path is $s \rightarrow a \rightarrow c \rightarrow d \rightarrow f \rightarrow t$.

**Remark:** (i) The proposed online distortion estimation method makes this work very different from all other existing works, where distortions are calculated based on predefined rate-distortion functions or models. In particular, the online distortion estimation at the source node makes it practical for the joint optimization of network routing and video source encoding. (ii) The proposed routing algorithm is efficient in dealing with network link breakages and node breakdown. With the proposed routing algorithm, a set of paths can be discovered and maintained simultaneously. Once the optimal path is broken due to either link outage or node breakdown, the suboptimal path with the second smallest distortion will be selected immediately for transmitting, and no path rediscovery needs to be initiated.

### 3.5 Convergence and Complexity Discussion of the Proposed Framework

#### 3.5.1 Convergence of the Proposed Routing Algorithm

Next, we will show that the optimal path can always be determined based on the following Lemmas:
Lemma 1

Once a label/node is marked as permanent, it will never be changed thereafter.

Proof: Recall that in Figure 3.5 (c) node \( c \) is selected to be marked as permanent if \( c \) has the smallest distortion among all tentative nodes \( \{b, c, d, e, f, t\} \). Thus, \( E[D_{i,j}^{c|a}] \geq E[D_{i,j}^{c|t}] \).

In the next step as shown in Figure 3.5 (d), node \( e \) is chosen to be marked as permanent. Then, the controller needs to calculate the expected distortions that the packet \( \pi_i^j \) would incur if the packet goes from node \( e \) to its next-hop nodes \( \{c, f\} \), respectively. Let \( E[D_{i,j}^{e|c}] \) denote the resulting expected distortion of packet \( \pi_i^j \) at node \( c \) passing through node \( e \).

Since \( E[D_{i,j}^{e|c}] \) is the resulting expected distortion of packet \( \pi_i^j \) after passing along an extra link \( (e, c) \) based on \( E[D_{i,j}^{e|t}] \) at node \( e \), then \( E[D_{i,j}^{e|c}] \geq E[D_{i,j}^{e|t}] \). Hence, \( E[D_{i,j}^{e|c}] \geq E[D_{i,j}^{c|a}] \) based on the above two inequalities: \( E[D_{i,j}^{e|c}] \geq E[D_{i,j}^{c|a}] \) and \( E[D_{i,j}^{e|c}] \geq E[D_{i,j}^{c|a}] \). Therefore, the permanent label of node \( c \) will not be updated.

From Lemma 1, we can derive that, for any working node, if its next-hop node is a permanent node, the expected distortion which the packet \( \pi_i^j \) would incur at the next-hop node does not need to be calculated. Thus, compared with the exhaustive search method, the computational overhead can be significantly reduced on path selection.

Lemma 2

Whenever the destination node is marked as permanent, the algorithm terminates, and the optimal path is found.

Proof: According to Lemma 1, for packet \( \pi_i^j \), whenever the destination node is marked as permanent, the label of the destination will not be changed. The permanent label has the smallest expected distortion of packet \( \pi_i^j \). Any other expected distortion at the destination caused by packet \( \pi_i^j \) passing through remaining tentative nodes would be larger than or equal to the distortion in the permanent label of the destination node. Therefore, there is no need to perform further calculations, and the optimal path can be selected by backward tracking the stored nodes in the permanent labels from the destination \( t \) to the source \( s \).
3.5.2 Computational Complexity of the Proposed Framework

The computational complexity of the proposed application-centric routing framework mainly comprises the following two parts:

Computational Complexity of the Routing Algorithm

The computational complexity of the proposed routing algorithm in Section 3.4.2 consists of three parts:

(i) The complexity in terms of comparison operations in calculating the optimal path for each possible packet by the routing algorithm, denoted by $C_R$. Note that the routing algorithm terminates after at most $|V| - 1$ iterations with the destination node being marked as permanent. The number of comparisons per iteration is proportional to $|V|$, where $|V|$ is the total number of nodes in the network. Therefore, in the worst case, the computational complexity is $O(|V|^2)$, comparing favorably with the worst-case estimate $O(|V|^3)$ of the Bellman-Ford algorithm. In fact, given the topology of a network, based on Lemma 1 and Lemma 2, the amount computations required to determine the optimal path from a source node to a destination node is considerably less than $O(|V|^2)$.

(ii) The complexity in calculating the expected video distortion for each packet to perform each comparison operation of (i), denoted by $C_D$. According to the ROPE method [88], the expected distortion is calculated at the video pixel level. Therefore, the value of $C_D$ is proportional to the number of pixels contained in one packet. For the detailed discussions on the values of $C_D$, please refer to [88].

(iii) The complexity in calculating the packet loss probability $P$ for each packet over each sub-path to perform each comparison operation of (i), denoted by $C_P$. From the Eqs. (3.10) - (3.18), we observe that $C_P$ is also a fixed value. Moreover, $C_P$ is much smaller than $C_D$, i.e., $C_P \ll C_D$. 
Therefore, the total computational complexity of the routing algorithm is $O(|V|^2 \times (C_P + C_D))$. Furthermore, to configure the $I$-stage trellis for dynamic programming, we need to run the routing algorithm $J^2$ ($J = |S|$) times for each stage. Thus, the total computational complexity before dynamic programming is performed is $O(I \times J^2 \times |V|^2 \times (C_P + C_D))$.

**Computational Complexity of the Dynamic Programming**

The computational complexity of the DP algorithm in Section 3.3 is $O(I \times J^z)$, which depends directly on the value of $z$. For most cases, $z$ is a small number (for the error concealment strategy used in this work, $z = 1$), so the algorithm is much more efficient than an exhaustive search algorithm with exponential computational complexity.

Note that the proposed solution is an operationally optimal solution. Therefore, based on the above analysis, the operational admissible set size can be adjusted according to the practical computational constraints.

### 3.6 Experimental Results

In this section, we evaluate the performance of the proposed quality-driven routing framework over multi-hop wireless networks with different network sizes and topologies. We consider 30- and 100-node networks both deployed over a 1000m x 1000m rectangular region. The source node and destination node are chosen randomly from the nodes in the network. The connectivity between the nodes is determined by the radio transmission range. The transmission range for each node is assumed to be 150m. The packet arrival rate at each node is taken with equal probability from the set [100, 120, 130, 140, 150] packets per second. For each network size, we generate 50 topologies and run 50 computations to obtain the average results. For a given topology, we assume that the DAG-modeled connectivity structure between the source node and the destination node, as shown in Figure 3.6, has been determined by the so-called proactive routing protocols such as Optimized Link State Routing (OLSR) [81], and is available to the controller. At the link level, we assume that
the link is frequency-flat, meaning that the link quality remains time-invariant during the transmission of a packet but may vary from packet to packet. The bandwidth of every link is set to 1 MHz. Furthermore, adaptive modulation and coding is performed over each link based on the instantaneous received SINR. Thus, according to Eq. (3.11), different links have different goodputs.

We first verify the proposed packet loss probability model proposed in Section 3.4.1. We calculate the end-to-end packet loss probability for packets with packet size $L = 10^4$ bits transmitted over a 6-hop path under different packet delay deadlines. The packet arrival rate $\lambda_u$ at each node $u$ is set to increase from 100 to 150 packets per second. Figure 3.7 shows that the packet loss probability increases with the increasing of packet arrival rate under a fixed packet delay deadline. This is because higher packet arrival rates lead to the longer packet waiting time for a packet during queueing, which increases its probability of
violating the packet delay deadline before it reaches the destination node. From Figure 3.7, we also observe that loose packet delay deadlines result in low packet loss probabilities with a fixed packet arrival rate. Table 3.2 shows the average accumulating packet delay at each hop along a 6-hop path under different packet delay deadlines $T_{\text{budget}}$.

Table 3.2: The average accumulating packet delay at each hop along a 6-hop path under different packet delay deadlines $T_{\text{budget}}$.

<table>
<thead>
<tr>
<th>$T_{\text{budget}}$ (seconds)</th>
<th>hop 1 (seconds)</th>
<th>hop 2 (seconds)</th>
<th>hop 3 (seconds)</th>
<th>hop 4 (seconds)</th>
<th>hop 5 (seconds)</th>
<th>hop 6 (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0333</td>
<td>0.0061</td>
<td>0.0124</td>
<td>0.0175</td>
<td>0.0221</td>
<td>0.0261</td>
<td>0.0312</td>
</tr>
<tr>
<td>0.0400</td>
<td>0.0072</td>
<td>0.0143</td>
<td>0.0239</td>
<td>0.0286</td>
<td>0.0323</td>
<td>0.0371</td>
</tr>
<tr>
<td>0.0500</td>
<td>0.0086</td>
<td>0.0165</td>
<td>0.0247</td>
<td>0.0322</td>
<td>0.0387</td>
<td>0.0473</td>
</tr>
<tr>
<td>0.0667</td>
<td>0.0093</td>
<td>0.0181</td>
<td>0.0294</td>
<td>0.0385</td>
<td>0.0470</td>
<td>0.0542</td>
</tr>
</tbody>
</table>

Four video sequences with varied contents (1 “Carphone”, 2 “Foreman”, 3 “Coastguard”, 4 “Mobile”) in QCIF format are considered in this chapter. The Y-component of the first
120 frames of each video sequence is encoded with H.264 (JVT reference software, JM 12.2 [98]). To explore the performance of the proposed framework under different packet delay deadlines, the video sequences are encoded at frame rates $f_1=15$ f/s and $f_2=30$ f/s (frames per second), which gives the resulting packet delay deadlines of 0.0667 seconds and 0.0333 seconds, respectively. All frames except the first one are coded as inter frames. In encoding inter frames, the selection from all possible sizes ($16 \times 16$, $16 \times 8$, $8 \times 16$, $8 \times 8$, $8 \times 4$, $4 \times 8$, and $4 \times 4$) of inter block search is enabled. To reduce error propagation due to packet loss, 10 random I Macroblocks were inserted into each frame. Constrained intra prediction was used at the encoder. For further error resilience, we also allow the first frame (Intra coded) to be correctly received at the destination node by providing enough time for the packet transmissions of that frame. We choose the quantization step size (QP) as the tunable source coding parameter. The permissible QP values are $[10, 13, 16, 19, 22, 25, 28, 31]$. To obtain a smooth perceptual video quality, the difference of the selected QPs for neighboring slices is limited within a threshold of 3. The frames are packetized such that each packet/slice contains one row of MBs, which enables a good balance between error robustness and compression efficiency.

The simulations compare four different approaches for joint optimization of path routing and video encoding, which are as follows.

1. **Application-Centric** — This is the proposed approach as described in Section 3.2, where the expected distortion is used as the routing metric, and QP is optimized for each slice to adapt video encoding to the underlying network conditions considering the constraint of packet delay deadline.

2. **Average packet delay** — Differently from the first approach, the optimal path is calculated such that the average end-to-end packet delay is minimized. Another difference is that this approach chooses QP from the available QP set for each video frame such that the expected distortion is minimized with the consideration of packet delay.
3. Average packet loss rate (PLR) — This approach calculates the optimal path to minimize the average PLR, taking into account the constraint of packet delay deadline. The same QP optimization and delay performance analysis as adopted in approach 2 are performed.

4. Hop count (HC) — This approach takes the path with the minimum number of hops as the optimal transmission path while satisfying the constraint of packet delay deadline. The same QP optimization and delay performance analysis as adopted in approaches 2 and 3 are also used here.

Figure 3.8 shows the received video quality measured by peak signal-to-noise ratio (PSNR), using the four different approaches. The PSNRs are averaged over all the inter-coded frames of a given video sequence with 50 computations under 50 topologies. Figures 3.8(a) and 3.8(b) show the results derived in the 30-node and 100-node network scenarios, respectively. The packet delay deadlines in both figures are 0.0333 seconds. These figures
show that the proposed application-centric routing algorithm significantly outperforms the other three network-centric routing approaches. This is because the application-centric routing approach is a quality-driven method, aiming at maximizing the received video quality in determining the optimal paths and video coding parameters of video units; while the network-centric routing approaches do not consider the effects of source coding and error concealment in routing. In other words, network-centric routing approaches are not quality-driven, i.e., minimizing the average end-to-end packet delay, the average end-to-end PLR or end-to-end hop count does not always lead to minimized video distortion.

Figure 3.9 shows the received video quality at each frame of the received “Mobile” sequence under a random topology in a 100-node network. The packet delay deadline is 0.0333 seconds. We observe that the proposed application-centric routing approach has achieved a significant improvement in received video quality over the other three network-centric routing approaches. Note that, in all of the above figures, the received quality by the delay-based routing approach is very close to what can be achieved by the PLR-based approach.
Table 3.3: Average video quality achieved by different routing approaches under the two different packet delay deadlines in a 100-node network (To conserve space, the video sequence names are abbreviated as “Ca”: “Carphone”, “F”: “Foreman”, “Co”: “Coastguard”, “M”: “Mobile”).

<table>
<thead>
<tr>
<th>Approach</th>
<th>$T_{\text{budget}} = 0.0333$ seconds</th>
<th>$T_{\text{budget}} = 0.0667$ seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“Ca”</td>
<td>“F”</td>
</tr>
<tr>
<td>Application-centric</td>
<td>34.57</td>
<td>33.39</td>
</tr>
<tr>
<td>Delay-based</td>
<td>31.23</td>
<td>29.31</td>
</tr>
<tr>
<td>PLR-based</td>
<td>30.04</td>
<td>29.04</td>
</tr>
<tr>
<td>Hop Count-based</td>
<td>27.96</td>
<td>25.26</td>
</tr>
</tbody>
</table>

method. This is because the retransmission mechanism is used at each hop and the packet loss rate is dominated by the packet drop rate caused by excessive queuing delay.

In our analysis on packet delay performance in Section 3.4.1, the packet delay deadline is a crucial performance parameter, directly affecting the estimation of the average end-to-end packet delay, average end-to-end packet loss rate and expected distortion. Therefore, the packet delay deadline significantly influences both the calculation of the routing path and the optimization of video coding parameters. Table 3.3 shows the comparison of average received quality under different routing approaches with different frame coding deadlines in the 100-node network. It can be concluded that, for two different packet delay deadlines, the proposed application-centric routing approach always has the best received video quality. More importantly, for any given video sequence, the proposed algorithm provides higher quality improvement than any other network-centric routing approach when the packet delay deadline increases from 0.033 to 0.0667 seconds. This is because the proposed joint optimization of routing and video encoding has the inherent advantage of enhancing the utilization of available network resources.

To evaluate the impact of source coding optimization in the proposed application-centric routing approach, we observe the slice QP selection in a transmission of the first 120 frames of the “Coastguard” sequence under an arbitrary 100-node network topology and a packet
Figure 3.10: The slice QP selection in a transmission of the “Coastguard” sequence in a 100-node network.

Table 3.4: Comparison of the proposed application-centric routing with the existing end-to-end path selection algorithm.

<table>
<thead>
<tr>
<th>Approach</th>
<th>$T_{\text{budget}} = 0.0333\text{seconds}$</th>
<th>$T_{\text{budget}} = 0.0667\text{seconds}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“Ca”</td>
<td>“F”</td>
</tr>
<tr>
<td>Proposed</td>
<td>34.57</td>
<td>33.39</td>
</tr>
<tr>
<td>Existing</td>
<td>31.43</td>
<td>30.28</td>
</tr>
</tbody>
</table>

delay deadline of 0.033 seconds. Figure 3.10 shows the QP adaptation over all the slices of “Coastguard”. We can observe that various QPs are chosen for different slices, indicating that the proposed application-centric routing approach has the intrinsic ability to adapt source coding to the underlying path selection in routing under the given lower-layer network conditions. This verifies the advantage of integrating source coding optimization with the proposed quality-driven routing.

To illustrate the perceptual video quality delivered by different approaches, we show the 99th frame of the original video of “Coastguard” in Figure 3.11(a) and compare it with
Figure 3.11: Comparison of the received video quality generated by different routing approaches.

the reconstructed video frames obtained by using the proposed application-centric routing approach (Figure 3.11(b)), the delay-based routing (Figure 3.11(c)), the PLR-based routing (Figure 3.11(d)) and the hop count-based routing (Figure 3.11(e)), respectively. Here, the network size is 100 nodes, and the frame decoding deadline is 0.0333 seconds. It can be observed that the frame under the proposed quality-driven approach has a visual quality very close to the original frame, while the frames under the other three approaches are considerably blurry.

Finally, we compare the proposed application-centric routing with a state-of-the-art routing algorithm - “end-to-end path selection” [77] in the 100-node network in Table 3.4. In “end-to-end path selection”, we select fixed source coding parameters for the current video frame such that as much data rate as in the proposed application-centric routing is generated. Moreover, we statistically determine the end-to-end path from a predeter-
mined path set such that the expected video distortion is minimized. The same packet
delay deadline concept and delay analysis method as in our application-centric routing
framework are both applied to “end-to-end path selection” with the same network settings.
The simulation results show that the proposed application-centric routing approach signifi-
cantly outperforms the ‘end-to-end path selection” algorithm, as it provides the capability
to jointly optimize video coding and routing.

3.7 Summary

Based on the analysis of the network-centric routing approaches, an application-centric
routing approach was proposed for real-time video communications in wireless multi-hop
networks. The proposed routing approach enables us to calculate an optimal routing path
to minimize the expected end-to-end video distortion within a given video packet delay
deadline. Within the proposed quality-driven framework, video source coding has been
integrated into the path routing to enhance the feasibility of multi-hop routing and the
utilization of network resources in a cross-layer manner. Experiments were conducted with
the H.264 codec and different sizes of multi-hop wireless networks. The results demonstrate
that the proposed quality-driven application-centric routing approach provides superior
end-to-end video quality over existing network-centric routing approaches.
Chapter 4

Cross-Layer Optimized Video Summary Transmission over Cooperative Wireless Networks

4.1 Introduction

Cooperative communication [99–101] has recently attracted significant attention as an effective transmission strategy, which takes advantage of the broadcast nature of wireless networks. The basic idea is to let nodes in a wireless network share information and transmit cooperatively as a virtual antenna array, which provides spatial diversity that significantly improves system performance. For example, as shown in Figure 4.1, node A would like to send information to node D. If the channel link between nodes A and D is blocked or in deep fading, then it would be difficult for node A to communicate with node D in terms of point-to-point wireless communications. However, if the nearby two nodes B and C can help node A by forwarding the information to node D, then the communications between nodes A and D is possible. In such a cooperative way, it is inherently more reliable for the destination to receive the transmitted information.

The emerging cooperative communication concept has fundamental impacts on MAC and higher network layer design of wireless networks as cooperative communications can improve network node connectivity, increase link throughput, save network power consumption, and even change network topology that allows shorter routing. Therefore, in the research community of cooperative communications, a considerable amount of work has been done for various networks and communication standards, such as cellular, WiFi, ad hoc/sensor networks, ultra-wideband (UWB) and IEEE 802.16j [102, 103].

Video summarization generates a short summary of the content of a huge volume of video data. Therefore, the receiver can get concise content information while the essential
Figure 4.1: An illustration of cooperative communications with source node A, relay nodes B and C, and destination node D.

(important) information of the original is well preserved (as illustrated in Figure 2.1 in Chapter 2). For example, for video surveillance in remote areas, network nodes are powered by either batteries or solar-energy-harvesting devices, meaning that the power is not always sufficient to transmit all the recorded video data. Moreover, most of recorded video data in video surveillance has significant redundancy. The redundant data needs not to be transmitted to save the power resource unless the moment some trigger events take place or activities are detected. To solve this problem, video summarization is a good choice in that it can significantly reduce the data amount to be transmitted with the consideration of video content coverage.

Integrating video summarization with cooperative communications has significant benefits for some emerging resource-limited wireless video applications such as video surveillance in homeland security, crime prevention, and battlefield monitoring. In these applications, power consumption, video delivery timeliness, and video quality and content coverage are fundamental issues. For power consumption, as mentioned before, video summarization can be used to reduce the data amount to be transmitted to conserve power resource. For the timeliness of video delivery, if the video delivery delay is too large, the received video data will be useless. Cooperation among network nodes can significantly decrease the expiration rate of video packets. To summarize, exploring video summary transmission in cooperative wireless networks can provide an effective solution for these power- and delay-constrained
video applications.

Currently, there are many techniques proposed in the video summarization field. However, there is very limited study on the video summary transmission over cooperative wireless networks due to the complexity of mixing together scene understanding, video coding, and wireless communications. In [15,62], video summary transmission over wireless networks was examined. However, neither the packet loss factor nor the source coding was considered, which might directly impact the perceptual quality of the reconstructed video at the receiver. In addition, these algorithms do not guarantee a good content coverage based on the selected frames because potential packet loss penalty heavily biases the frame selection process. In Chapter 2, we proposed a cross-layer optimization framework for video summary transmission that considers both source coding and content coverage [42]. However, the algorithm only aims at generic wireless networks without considering the advantages of cooperative communication between network nodes.

In this chapter, we propose to enhance the processing capability of relay nodes in cooperative wireless networks, so that these nodes are not limited to the forwarding function. Instead, they process the received video packets and generate error-resilient information for the error concealment at the destination side. Specifically, a summary of summary (SoS) video processing model is developed at the relay nodes to extract the most important information and to deliver them to the destination to improve the received video quality. In this way the proposed framework takes advantage of the inherent cooperative nature of wireless networks, as well as considers the characteristics of video summary applications. The novelty of this work is two-fold: it is the first cross-layer optimization framework for video summary transmission over cooperative wireless networks; the proposed cooperative relay scheme as well as the SoS processing model plays a critical role in improving the system performance of video summary transmission.

The remainder of this chapter is organized as follows. In Section 4.2, we briefly introduce the system model, including the video summary distortion model and the proposed
cooperative video summary transmission model. In Section 4.3, we describe the proposed cooperative video summary transmission scheme and formulate a cross-layer optimization problem for video summary transmission in cooperative wireless networks. We present our DP-based optimal solution in Section 4.4 and experimental results in Section 4.5. Section 4.6 concludes the chapter.

4.2 System Model

4.2.1 Video Summary Distortion Model

Let $J$ denote the number of frames of a video clip $\{f_1, f_2, \ldots, f_J\}$, and $I$ the number of frames of its video summary sequence $\{g_1, g_2, \ldots, g_I\}$. In this work, how to generate the video summary at the source node is not within the scope of our framework. Let $l_i$ be the index of the summary frame $g_i$ in the video clip. At the receiver side, the video clip is reconstructed by substituting missing frames with the corresponding summary frames. Let $\tilde{f}_j$ denote the displayed $j$th frame from the received summary at the receiver side. Let $\tilde{g}_i$ be the reconstructed the $i$th summary frame. Note that any summary frame may get lost during transmission. However, in order to simplify the problem formulation, we assume the first summary frame is reliably received. In this work, we also consider the content coverage issue of the received summary: we introduce a parameter $z$ such that there will be no $z$ consecutive summary frames being lost. This assumption can be satisfied by link adaptation techniques in wireless networks [42]. Note that $z$ is a programmable constant, and the introduction of the parameter does not narrow down the original problem. Then, according to the video reconstruction process as shown in Figure 2.2, the expected distortion of the video clip at the receiver side can be calculated by

$$E[D] = \sum_{j=1}^{J} E\left[d[f_j, \tilde{f}_j]\right]$$

$$= \sum_{i=1}^{I} \sum_{j=l_i}^{l_{i+1}-1} \sum_{b=0}^{\min(i-1, z-1)} \left\{(1-p_{i-b})d[f_j, \tilde{g}_{i-b}] \prod_{a=0}^{b-1} p_{i-a}\right\}.$$
where $p_i$ is the video summary frame loss rate in transmission, and function $d[\alpha, \beta]$ is the distortion measurement with the mean squared error (MSE) metric between frames $\alpha$ and $\beta$.

4.2.2 Cooperative Video Summary Transmission Model

To illustrate the basic idea of the proposed video summary cooperative transmission, without losing generality, a simple topology with a source node $s$, a relay node $r$, and a destination node $d$ is adopted as shown in Figure 4.2. Cooperative transmission is conducted in two phases. In Phase 1, $s$ broadcasts a message to both $d$ and $r$ using transmission power $P_s$. In Phase 2, $r$ sends information to $d$ (at different time slots or different orthogonal channels) using transmission power $P_r$, and $d$ combines and detects information from both $s$ and $r$. To facilitate the discussion, we define $P_0$ as the total available transmission power which can be shared by $s$ and $r$.

In this work, a Rayleigh fading channel model is used to model all the links in the network. For example, for the link $s \rightarrow d$, the received signal at the destination in Phase 1 can be written as $y^{s,d} = \sqrt{P_s G^{s,d}x} + \eta^{s,d}$, where $G^{s,d}$ is the channel gain from $s$ to $d$, $x$ is the transmitted information symbol, and $\eta^{s,d}$ is additive noise with noise variance $\sigma^2$. A closed-form analysis on the symbol-error-rate performance of different transmission/modulations schemes in Rayleigh fading channels can be found in [100].
In what follows we give a brief introduction of conventional transmission schemes for the 3-node topology as shown in Figure 4.2 to highlight the differences between these schemes and our proposed cooperative transmission scheme.

- **Direct transmission (DT):** Without the help from $r$, the information is transmitted only by the direct link $s \rightarrow d$.

- **Amplify-and-forward (AF):** In Phase 2, $r$ amplifies the received noisy signal from $s$ and forwards it to $d$. $d$ combines the waveforms sent from $s$ and $r$, and makes a final decision on the transmitted information.

- **Decode-and-forward (DF):** $r$ decodes the source information in Phase 1 and retransmits it to $d$ in Phase 2. $d$ combines the direct transmission information and relayed information together.

- **Coded cooperation (CC):** Channel coding is integrated into this scheme by letting $s$ and $r$ send different portions of a source codeword [99]. In Phase 1, $s$ sends part of the codeword to $r$ and $d$. $r$ decodes the source information and reconstructs the codeword with the same channel coding method as used at $s$. In Phase 2, $r$ punctures the codeword and transmits the resulting incremental redundancy to $d$. $d$ performs joint channel decoding on the received information from $s$ and $r$.

- **Multipath transmission (MT):** $r$ decodes the information sent from $s$ during Phase 1, and it forwards the decoded signal to $d$ during Phase 2. Due to the possible channel difference between the path $s \rightarrow d$ and the path $s \rightarrow r \rightarrow d$, the bit error at $d$ happens only if both bit decoding for the two paths fail.

Note that AF, DF, and CC are three typical cooperative transmission schemes [99]. It is also worth noting that all the aforementioned transmission schemes lack the ability to perform video rate and quality adaptation at the relay $r$. Consequently, the resource allocation for both video source and channel along the path $s \rightarrow r \rightarrow d$ is not well explored.
4.3 The Proposed Scheme

Due to the inherent characteristics of video data, motion activity, texture complexity, and perspective sensitivity of video quality make the importance of each video bit unequal. In this work, we propose a decode-process-forward (DPF) scheme where video packets are further processed at the relay and the useful information is extracted and delivered to the destination.

4.3.1 The Proposed DPF Scheme

In the DPF scheme, as shown in Figure 4.2, $s$ encodes the video summary frames $g_i (i = 1, 2, \ldots, I)$ into packets and broadcasts them to $r$ and $d$. Once $r$ receives a video summary frame $g_i$, it processes the frame to extract a concise version $g'_i$, called summary of summary (SoS), and then forwards it to $d$. The SoS can be obtained after a few processing steps with various levels of complexities. Depending on the system settings and network conditions, possible video processing methods are:

- Down-sampling the image;
- Filtering the high-frequency component of the image;
- Encoding or transcoding the video frame with a lower bit budget;
- Extracting the region of interest (ROI) information [56];
- Dropping the current video summary frame.

The destination $d$ combines the packets received from $s$ and $r$ via the paths $s \rightarrow d$ and $s \rightarrow r \rightarrow d$, respectively, and recovers the video frame using a pre-defined error concealment strategy. It is important to emphasize that the process and error concealment strategies used in $r$ and $d$ are known to the system controller which resides in $s$, controls and optimizes the parameter settings of all modules based on specific application requirements, channel conditions, and computational complexity levels.
4.3.2 Problem Formulation

Let $Q_{i}^{s}$ denote the source coding parameters at $s$ for the $i$th summary frame $g_{i}$; $Q_{i}^{r}$ the processing parameters at $r$ to extract $g_{i}^{r}$, respectively; Let $\tilde{g}_{i}$ be the reconstructed $i$th summary frame with the packets directly from $s$, and $\tilde{g}_{i}^{r}$ the reconstructed $i$th summary frame with the packets from $r$. Here we also consider the video content coverage issue. Then the expected distortion of the whole video clip at the destination side can be expressed as

$$E[D] = \sum_{j=1}^{J} E\left[d(f_{j}, \tilde{f}_{j})\right]$$

$$= \sum_{i=1}^{l} \sum_{j=1}^{l_{i}} \sum_{b=0}^{b_{i}+1} \min(i-1,z-1) \left\{ (1 - p_{i-1-b}^{s,d})d[f_{j}, \tilde{g}_{i-b}] + p_{i-1-b}^{s,d} \cdot (1 - p_{i-1-b}^{s,r})(1 - p_{i-1-b}^{r,d})d[f_{j}, \tilde{g}_{i-b}] \right\}$$

$$\cdot \prod_{a=0}^{b-1} \left[ p_{i-a}^{s,d} \cdot \left( 1 - (1 - p_{i-a}^{s,r})(1 - p_{i-a}^{r,d}) \right) \right]$$

(4.2)

where $p_{i}^{s,d}$, $p_{i}^{s,r}$ and $p_{i}^{r,d}$ are the loss probabilities for the video summary frame over the links $s \rightarrow d$, $s \rightarrow r$ and $r \rightarrow d$, respectively. The relationship of the video frame loss probability and the packet loss probability on each link depends on the specific packet encapsulation or packet fragmentation scheme. Without loss of generality, we assume that each video summary frame is compressed into one packet. Thus, the video frame loss probability is equivalent to the packet loss probability on each link. Equation (4.2) indicates that when reconstructing $f_{j}$, the closest reconstructed summary frame $\tilde{g}_{i}$ to $f_{j}$ is preferred; if $\tilde{g}_{i}$ is not available, $\tilde{g}_{i}^{r}$ will be used; if neither of $\tilde{g}_{i}$ and $\tilde{g}_{i}^{r}$ is available, the second closest reconstructed summary frame $\tilde{g}_{i-1}$ to $f_{j}$ will be considered. The process continues until the closest correctly received summary frame to $f_{j}$ from either $s$ or $r$ is available.

Let $T_{i}^{s}$ and $L_{i}^{s}$ denote the transmission delay and the consumed bit number of the $i$th summary frame $g_{i}$ at path $s$, respectively; $T_{i}^{r}$ and $L_{i}^{r}$ the transmission delay and the consumed bit number of the $i$th SoS frame $g_{i}^{r}$ at $r$, respectively. Given the transmission
rate $R^{s,d}$ at $s$ and $R^{r,d}$ at $r$, $T^s_i$ and $T^r_i$ can be expressed as

$$T^s_i = \frac{L^s_i(Q^s_i)}{R^{s,d}},$$

$$T^r_i = \frac{L^r_i(Q^r_i, Q^s_i)}{R^{r,d}}.$$  \hfill (4.3)

Then the total expected delay for the whole summary is

$$E[T] = \sum_{i=1}^{I} E[T_i]$$

$$= \sum_{i=1}^{I} \left\{ (1 - p^{s,d}_i)T^s_i + p^{s,d}_i (1 - p^{s,r}_i)(1 - p^{r,d}_i)T^r_i ight. \
\left. + p^{s,d}_i [1 - (1 - p^{s,r}_i)(1 - p^{r,d}_i)] T_c \right\}$$ \hfill (4.4)

where $T_c$ is a constant, which means if neither $g_i$ nor $g^r_i$ is correctly received by $d$ within $T_c$, they both are assumed lost.

Therefore, the problem can be formulated as to minimize the end-to-end expected distortion $E[D]$ by choosing the power allocation scheme $\{P^r, P^s\}$, the source-coding parameters $Q^s_i$, and the relay video-processing parameters $Q^r_i$ under both the power and the delay constraints, i.e.,

$$\min_{\{P^r, P^s, Q^s_i, Q^r_i\}} E[D],$$ \hfill (4.5)

$$\text{s.t.} \quad E[T] \leq T_{\text{budget}}, \quad P^r + P^s \leq P_0.$$

where $T_{\text{budget}}$ is the maximum allowable delay time for transmitting the video summary. In real-time video communications, $T_{\text{budget}}$ can be approximated as $T_{\text{budget}} = 1/f_{\text{fps}}$ where $f_{\text{fps}}$ is the video frame rate (frames per second, fps) \cite{85}. It is also worth mentioning that various transcoding/processing schemes can be employed at the relay without affecting the problem formulation.
4.4 Solution Procedure

4.4.1 Optimal Solution

For simplicity, we define the power allocation factor \( k \) between \( s \) and \( r \), i.e.,

\[
P_s = k P_0, \quad P_r = (1 - k) P_0,
\]

where \( k \) is a variable which needs to be optimized. Let \( V \) be the set of all possible decision vectors \( v_i \) for the \( i \)th summary frame (i.e., \( g_i \) and \( g_r^i \), \( i = 1, 2, \ldots, I \)), where \( v_i = \{k, Q_s^i, Q_r^i\} \).

Let \( v = \{v_1, v_2, \ldots, v_I\} \) denote the parameter vector for the whole video summary. We use the Lagrange multiplier method to relax the delay constraint. Then the Lagrangian cost function with the Lagrange multiplier \( \lambda \) is defined as

\[
J_\lambda := E[D] + \lambda E[T].
\]

(4.7)

It has been shown [72] that if there is a \( \lambda^* \) such that

\[
v^* = \arg \min_v J_{\lambda^*}(v)
\]

leads to \( T(v^*) = T_{\text{budget}} \), then \( v^* \) is also an optimal solution to (4.5). Therefore, the task of solving (4.5) is to find the pair \( (\lambda^*, v^*) \).

Next, we first explain how to find \( v \) for a given \( \lambda \) such that \( J_\lambda \) is minimized, and then we present the algorithm of searching for \( \lambda^* \). The video clip reconstruction introduces dependencies between frames. For the assumption that no \( z \) consecutive summary frames are lost on the path \( s \to d \), the reconstruction will cause the current video frame to depend on its \( z \) previous summary frames. To find \( v^* \) such that \( v^* = \arg \min_v J_{\lambda^*}(v) \), we define a cost function \( H_n(v_{n-z}, \ldots, v_n) \) which represents the minimum total distortion and delay up to and including the \( n \)th summary frame, given that \( v_{n-z}, \ldots, v_n \) are decision vectors for the \((n - z)\)th to \( n \)th summary frames. Therefore, \( H_I(v_{I-z}, \ldots, v_I) \) represents the minimum delay and distortion for all the summary frames and thus

\[
\min_v J_\lambda(v) = \min_{v_{I-z}, \ldots, v_I} H_I(v_{I-z}, \ldots, v_I).
\]

(4.9)
The key observation for deriving an efficient algorithm is the fact that given $z+1$ decision vectors $v_{n-z-1}, \ldots, v_{n-1}$ for the $(n - z - 1)$th to $(n - 1)$th summary frames, and the cost function $H_{n-1}(v_{n-z-1}, \ldots, v_{n-1})$, the selection of the next decision vector $v_n$ is independent of the selection of the previous decision vectors $v_1, v_2, \ldots, v_{n-z-2}$. This is true since the cost function can be expressed recursively as

$$
H_n(v_{n-z}, \ldots, v_n) = \min_{v_{n-z}, \ldots, v_n} \left\{ H_{n-1}(v_{n-z}, \ldots, v_{n-1}) \right. \\
+ \sum_{j=1}^{b_{n+1}-1} \min_{n-1,z-1} \left\{ \sum_{b=0}^{l_n} \left[ (1 - p_{n-b}^s d) f_j, \tilde{g}_{n-b} \right] \right. \\
+ p_{n-b}^s d \cdot (1 - p_{n-r-b}^s d) (1 - \tilde{g}_{n-b}^r) \left[ f_j, \tilde{g}_{n-b}^r \right] \\
\left. \cdot \prod_{a=0}^{b-1} \left[ (1 - p_{n-a}^s d)(1 - \tilde{g}_{n-a}^r) \right] \right\} + \lambda E(T_n) \right\}. \quad (4.10)
$$

The recursive representation of the cost function makes the future step of the optimization process independent from its past step, which is the foundation of dynamic programming. Thus, the problem $\min \mathcal{J}_\lambda(v)$ can be converted into a graph theory problem of finding the shortest path in a directed acyclic graph (DAG) [90].

As an example of the shortest path algorithm, the path map for the case of $z = 2$ is shown in Figure 4.3. There are total $I$ stages, each stage corresponding to each summary frame. At each stage, there are $|V|$ nodes where $|V|$ is the cardinality of $V$ corresponding
to the whole decision variable space, i.e., $|V| = |k||Q_n^s||Q_n^r|$. The weight $h_{n,y}^{x,y}$ on each edge from node $v_{n-1}^x$ to $v_n^y$ corresponds to the incremental cost value when the $n$th summary frame $g_n$ and its concise version $g_n^r$ are processed by the $y$th option of $v_n$ given that $g_{n-1}$ and $g_{n-1}^r$ is processed by the $x$th option of $v_{n-1}$. $h_{n,y}^{x,y}$ is equal to the sum of the second and the third terms of the right hand side of (4.10). $H_n(v_n^y)$ refers to the minimum cost value for up to the $n$th summary frames when the $n$th summary frame is processed by the $y$th option of $v_n$. With the path map, the shortest path algorithm is to find a path which has the minimum total cost value from the point “start” to “end”.

For the search of $\lambda^*$, it is well known that when $\lambda$ sweeps from zero to infinity, the solution to problem $v^* = \arg \min_V J_{\lambda^*}(v)$ traces out the convex hull of the distortion delay curve, which is a non-increasing function. Hence $\lambda^*$ can be obtained via a convex recursion in $\lambda$ using a bisection algorithm [42].

4.4.2 Implementation Considerations

The computational complexity of the above algorithm is $O(I \times |V|^{z+1})$ ($|V|$ is the cardinality of $V$), which depends on the values of $z$ and $|V|$. For most cases in video summarization applications, $z$ is a small number, so the algorithm is much more efficient than an exhaustive search algorithm which has exponential computational complexity. On the other hand, the complexity can also be decreased by reducing the cardinality of $V$. Thus, the practical solution would be an engineering decision and tradeoff between the computational capability and the optimality of a solution. It is also worth mentioning the frequency to perform the algorithm can be only once or very slow if node mobility is low. For convergence in searching $\lambda^*$, the convergence speed depends on the required accuracy of $\lambda^*$. To speed up the convergence, the fast convex search present in [90] can be used to find $\lambda^*$. 
### Table 4.1: Simulation parameter values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Optional values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power allocation factor: $k$</td>
<td>0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1</td>
</tr>
<tr>
<td>QP at the source:</td>
<td>10, 13, 16, 19, 22, 25, 28, 31, 34</td>
</tr>
<tr>
<td>QP at the relay:</td>
<td>10, 13, 16, 19, 22, 25, 28, 31, 34</td>
</tr>
<tr>
<td>Scaling rate at the relay:</td>
<td>176x144, 96x80, 64x48</td>
</tr>
<tr>
<td>Truncating rate at the relay:</td>
<td>1, 4, 8, 16</td>
</tr>
<tr>
<td>Noise power level: $\sigma^2$</td>
<td>$2 \times 10^{-10}$ Watt</td>
</tr>
<tr>
<td>Bandwidth: $W$</td>
<td>400kHz</td>
</tr>
<tr>
<td>Total power limit: $P_0$</td>
<td>1 Watt</td>
</tr>
<tr>
<td>Video frame rates (fps):</td>
<td>10, 15, 20, 30, 40, 50</td>
</tr>
<tr>
<td>Total delay limit: $T_{\text{budget}}$ (seconds):</td>
<td>7.5, 5, 3.75, 2.5, 1.875, 1.5</td>
</tr>
<tr>
<td>Source node location:</td>
<td>$(0, 0)$</td>
</tr>
<tr>
<td>Relay node location:</td>
<td>$(x, y)$</td>
</tr>
<tr>
<td>Destination node location:</td>
<td>$(50m, 0)$</td>
</tr>
<tr>
<td>Pass-loss model constant: $K$</td>
<td>-31.54 dB</td>
</tr>
<tr>
<td>Pass-loss exponent: $\gamma$</td>
<td>4</td>
</tr>
</tbody>
</table>

#### 4.5 Experimental Results

Experiments were carried out to evaluate the performance of the proposed DPF scheme by using H.264/AVC JM 12.2. The video sequence “Glasgow” (QCIF) was adopted as the test clip. The first 750 frames of the “Glasgow” sequence were summarized into 75 frames. The video summary was intra-coded for each summary frame due to the less correlation between its neighboring frames. We considered the quantization step size (QP) as the tunable video coding parameter at the source node, and QP and the scaling rate as well as the truncating rate of the DCT coefficients as the adjustable video processing parameters at the relay node. Table 4.1 shows the simulation setups.

The locations of $s$, $r$, and $d$ are denoted by their planimetric rectangular coordinates. To simulate channel coefficients such as $G^{s,d}$, we used a simplified path loss model $G = K(d)^{-\gamma}$ where $K$ is a unitless constant that depends on the antenna characteristics and the average channel attenuation, $d$ is the distance between transmitter and receiver and $\gamma$ is the pass-loss exponent. In our experiments, the 75-frame video summary were transmitted with different video frame rates [10, 15, 20, 30, 40, 50] fps. Therefore, the delay budget $T_{\text{budget}}$ are
Figure 4.4 : Performance of the proposed DPF scheme for video summary delivery over cooperative wireless networks.

[7.5, 5, 3.75, 2.5, 1.875, 1.5] seconds. We compared the performance of the proposed DPF scheme with those of other three conventional transmission schemes: DT, DF, and MT. For fair comparisons, the video coding parameters $Q^s$ at the source in these three schemes were also optimized.

Figure 4.4(a) shows the comparison of the received video quality measured by peak signal-to-noise ratio (PSNR), using different transmission schemes. The location of $r$ is (25m, 25m). It is observed that the proposed scheme always has significant performance gains over the other three schemes for different values of $T_{budget}$. Specifically, our proposed scheme obtains a 2~3 dB of performance gain over DF, and a 6~7 dB of performance gain over DT. This indicates that the proposed scheme not only exploits the path diversity of fading channels in cooperative communications, but also employs the resource allocation adaptation achieved by the flexible video processing capabilities of the relay.

Figure 4.4(b) shows the performance of the proposed scheme with different power allocations between $s$ and $r$. The relay location is (25m, 25m) and the total transmission power is 1 watt. It shows that the optimal performance under different $T_{budget}$ is achieved when on average 60% ($k = 0.6$) of the total available power is allocated to the source node. This
means that excessive power consumption by the source or relay will not bring a significant distortion performance improvement. Therefore, the optimal power allocation between $s$ and $r$ needs to be performed to achieve a significant performance gain with the proposed DPF scheme.

To illustrate the perceptual video quality delivered by different transmission schemes, we plot a sample video frame from the original video of “Glasgow” in Figure 4.5(a) and compare it with the reconstructed video frames obtained by using the proposed DPF scheme (Figure 4.5(b)), the DF scheme (Figure 4.5(c)), the MT scheme (Figure 4.5(d)), and the DT scheme (Figure 4.5(e)), respectively. Here, $T_{\text{budget}}$ is 3.75 seconds. It can be observed that the frame under the proposed DPF transmission scheme has a visual quality very close to the original frame, while the frames under the other three schemes are considerably blurry.
4.6 Summary

In this chapter, a novel video cooperative communication scheme has been proposed with considerations of the unique feature of videos. A summary of summary video processing model is used in the relay node to enhance the video transmission efficiency. A cross-layer optimization framework for video summary transmission over cooperative wireless networks has been proposed. Theoretical analysis indicates that the proposed framework takes advantage of the inherent cooperative nature of wireless networks, as well as considers the characteristics of video summarization. A DP-based approach was developed to obtain the optimal solution of the cross-layer optimization problem. It is the first cross-layer optimization framework for video summary transmission over cooperative wireless networks. The experimental results show that compared with both existing cooperative and non-cooperative transmission scheme in the literature, the proposed scheme can achieve significant improvements regarding the received video quality.
Chapter 5

Interaction Measure and Sensitivity Analysis in Cross-Layer Design

5.1 Introduction

In recent years, cross-layer design has been thought as one of the most effective and efficient ways to provide quality of service (QoS) over various communication networks [13, 104, 105], especially over wireless multimedia networks, in which the physical nature of the transmission medium poses a series of design challenges on wireless multimedia system design such as limited bandwidth, fading, and interference. In general, the basic idea of cross-layer design is to fully utilize the interactions among design variables residing in different network functional entities (such as layers in the network protocol stack) to achieve the optimal system performance under complex dynamic environment. So far, numerous cross-layer design schemes with different design objectives have been developed by using various design optimization methods [6, 21, 23, 27, 28, 42, 49, 77, 106–116]. For a concise and consistent presentation, in this chapter, we will adopt “design variables” and “design objective” to have the same meanings as “system parameters” and “system performance”, respectively.

However, it has been shown that most current research on cross-layer design has been carried out in various piecemeal approaches [117], in which cross-layer solutions simply assemble several network layers together. Piecemeal approaches may cause so-called design paradoxes such as the Ellsberg Paradox [30], in which each individual design variable residing at certain network layer makes the “best” decision to maximize the design objective at the local scope, but the overall system performance may be worse than that of not doing any optimization. Therefore, the phenomenon of a design paradox such as the Ellsberg Paradox
tells us that breaking a big problem into multiple small problems can only increase the solvability of the original problem but cannot guarantee the optimality of solving the original problem due to the tightly-coupled interactions among design variables. Furthermore, the appearance of a design paradox also indicates that the traditional additive measure methods such as probability measure may no longer hold in the context of cross-layer design due to the existence of various uncertainties and the lack of enough statistics during short-term observation intervals. Moreover, existing cross-layer solutions assume that the more design variables are considered, the system performance will be better. However, more does not necessarily mean better. Sometimes cross-layer design may lead to “spaghetti” design [118], damaging the modularity and the generality of the original system. Therefore, we conclude that all aforementioned problems in the area of cross-layer design are caused by lack of a methodological foundation to gain in-depth understanding of complex cross-layer behaviors such as temporal-spatial behavior and multi-scale behavior, which will be discussed in Section 5.2.1.

To gain fundamental understanding of cross-layer behaviors, a quantitative measure of various interactions among design variables on the design objective is very crucial. So far, there has been research done on interaction modeling in the context of cross-layer design, but most current research is qualitative and piecemeal rather than quantitative and systematic. Barrett et al. [32, 33] adopted experimental design and statistical analysis to characterize the interaction between routing and MAC protocols in ad-hoc networks. [34] proposed to model protocol interactions as optimization problems. The protocol interactions of ad hoc networks were studied by using statistical design of experiments [35] and response surface methodology [36]. [31] gave a comprehensive review on the formal methods in cross-layer modeling and optimization of wireless networks. [37] proposed a quantitative study of cross-layer performance optimization for Voice over WiFi communications based on a formal framework outlined in [38]. In [39], a metamodeling approach was introduced to study cross-layer scheduling in wireless local area networks. Although in literature cross-
layer architecture and cross-layer interaction modeling were studied, research is still needed to provide a theoretical formulation and quantitative analysis on cross-layer design issues, which will be crucial to avoid design pitfalls in existing design and provide design insights for future design.

In this work, we define a new concept, namely the interaction measure, based on the non-additive measure theory. Then, we propose a theoretical framework based on the nonlinear integral and non-additive measure to quantify various interactions among design variables of cross-layer design. The main contributions of this chapter are: 1) the concept of the interaction measure is proposed to quantitatively measure interactions among design variables towards to the design objective; 2) a nonlinear multivariate regression model for the interaction measure based on network management data is developed; 3) a sensitivity analysis algorithm based on quantitative interaction measure for both commensurable and incommensurable datasets is proposed; and 4) the proposed theoretical framework is validated through a case study on cross-layer optimized wireless multimedia communications.

The rest of the chapter is organized as follows. In Section 5.2, we give the definition of the interaction measure and introduce the quantitative model of the interaction measure in cross-layer design. Section 5.3 describes different methods of sensitivity analysis based on quantitative interaction measure. In Section 5.4, a case study on cross-layer optimized wireless multimedia communications is conducted to illustrate the major cross-layer design tradeoffs and to validate the proposed framework. Finally, we conclude the chapter in Section 5.5.
5.2 The Proposed Theoretical Framework for Quantitative Interaction Measure

5.2.1 Problem Description

Without loss of generality, let \( X = \{x_1, x_2, \ldots, x_N\} \) be a set of design variables residing in different network layers. Let \( Y \) be the performance metric of interest (the design objective function) of cross-layer design, which is a nonlinear function of \( \{x_1, x_2, \ldots, x_N\} \). In contrast with the layering paradigm where design variables of a layer can only interact with the design variables of neighboring layers, cross-layer design allows coordination, interaction and joint optimization of design variables crossing different layers to maximize the design objective function. A general cross-layer optimization problem can be formulated as

\[
\begin{align*}
\text{maximize} & \quad Y(x_1, x_2, \ldots, x_N) \\
\text{subject to} : & \quad h_m \geq 0, \text{ where } m \in \{1, 2, \ldots, M\}. 
\end{align*}
\] (5.1)

Here, \( h_1 \) to \( h_M \) are \( M \) design constraints, which may be posed by network resource limitations, QoS requirements, and upper and lower bounds of design variables. Design constraints set a bound on the best achievable design performance.

Temporal-Spatial Behaviors of Cross-Layer Design

Communication networks are dynamic complex systems due to the existence of uncertainty and nonlinearity. Essentially, cross-layer optimization can be modeled as a continuous multivariate variational problem, in which both states and constraints are functionals. Solving a variational problem might be difficult to derive closed-form solutions, especially to high-dimensional variational problems, due to excessive computation overhead of partial differential equation (PDE) and performance sensitivity on boundary conditions. In fact, there is no theory to guarantee the existence of closed-form results on nonlinear multidimensional variational problems under various boundary conditions. In cross-layer design,
boundary conditions are the value range of each design variable. Cross-layer design can be further complicated by time-varying optimization constraints. As a result, in practice, only a set of discrete operating points of each design variable is considered. Therefore, when approximating continuous multidimensional variational problems by using discrete models under dynamic environment, we need to fully understand the temporal-spatial behaviors in cross-layer design.

In general, temporal-spatial behaviors refer to various design tradeoffs in the selection of the operating points of design variables in terms of temporal and spatial scales. In dynamic environment, temporal and spatial scales are time-varying, leading to a multiscale system. For example, network events occur at multiple temporal scales - packets are transmitted in microseconds, medium access occurs in milliseconds, routing tables are updated in seconds or minutes, and network topology changes in days or months. The temporal scaling feature makes it very challenging to choose the optimal adaptation interval in cross-layer optimization. Similarly, spatial scaling issues can be found in cross-layer optimization, since many control variables in cross-layer optimization are continuous, such as power and signal-to-interference-noise ratio (SINR), which have to be discretized into a set of operating points (preselected values) to reduce the possible excessive computational and storage overhead.

To illustrate the temporal-spatial behaviors of cross-layer design, we develop a dynamic programming structure, as shown in Figure 5.1, since dynamic programming is the only exact method for global optimization over time with nonlinearities and random disturbances [119]. In Figure 5.1, assume that a variational problem is optimized over time period $T$, which can be approximated by using $M$-stage dynamic programming [120] where $t_i = 0, \tau, 2\tau, 3\tau, \ldots, k\tau, \ldots, M\tau$ and $i = 1, 2, \ldots$. Here, a fundamental question of this discretization process is how to choose the temporal scale \(\tau\) to achieve the optimal design tradeoff between computational complexity and approximation accuracy. Moreover, the selection of \(T\) is important to the performance of cross-layer design, which is closely related to the signaling mechanism in terms of the feedback delay caused by the underlying network.
Figure 5.1: The illustration of temporal-spatial behaviors of cross-layer design by using dynamic programming. The horizontal axis shows the temporal adaptation in cross-layer design, where $T$ is the adaptation interval and $\tau$ is the temporal resolution. The vertical axis shows the spatial adaptation, where the continuous values of the design variables are coarsely or finely discretized, resulting in different sets of operating points with different discretization levels.

As we mentioned earlier, in dynamic systems, many design variables are continuous. To achieve a feasible implementation, each of these continuous variables has to be discretized into a set of preselected operating points, such as quantization parameters built in H.264 video codec [1], multiple modulation and coding schemes specified by the IEEE 802.16 physical layer [10], and various choices of packet length supported in multiple network layers. Therefore, a continuous state space will turn into a discrete state space. Furthermore, as shown in Figure 5.1, to achieve the best system performance, design variables are dynamically adjusted within the preselected set of discretized operating points in different temporal and spatial scales.

To build an accurate cross-layer design model and to achieve the optimal performance of cross-layer design, we need to understand major interactions among design variables. Therefore, quantitative interaction measure and sensitivity analysis become the most important
issues in cross-layer design.

**Challenges in Sensitivity Analysis of Cross-Layer Design**

To gain an in-depth understanding of the aforementioned temporal-spatial behaviors and to achieve the optimal system performance of cross-layer design, we need to evaluate the contribution made by design variables $X = \{x_1, x_2, \ldots, x_N\}$ and their interactions on the design objective function $Y$. However, to that end, we have to overcome several challenges. First, the design objective function is a nonlinear function of design variables, meaning that it is not easy to derive a close-form expression for nonlinear optimization. Second, various interactions among design variables may significantly affect the performance of cross-layer design. Third, it is very difficult to characterize the interactions among design variables due to uncertainty and randomness existing in cross-layer design. To gain an in-depth understanding of cross-layer design, it is highly desirable to quantitatively measure the contribution of each subset of design variables in $X$.

**5.2.2 Interaction Measure**

In this work, we give the definition of the interaction measure on $X$ to quantitatively evaluate interactions among the design variables in each subset of $X$, based on non-additive measure theory [121].

*Definition 1*

The interaction measure on $X$ is defined as a set function $\mu : \mathcal{P}(X) \to \mathcal{R}$ with a constraint $\mu(\emptyset) = 0$, where $\mathcal{P}(X)$ is the power set of $X$, and $\mathcal{R}$ is the real domain.

Note that we have relaxed the following two traditional restrictions in non-additive measure theory on the interaction measure: (i) the co-domain of the set function $\mu$ is $\mathcal{R}$ instead of $\mathcal{R}^+$; (ii) the monotonicity, $A \subset B \subseteq X$ implies $\mu(A) \leq \mu(B)$, is unnecessary. These relaxations are based on the observations that design variables may negatively affect the design objective in a given cross-layer design. The unique feature of the interaction
measure defined above is that it can express interactions among design variables being aggregated in a more flexible and accurate manner. Recall that \( N \) is the number of design variables in cross-layer design. Then, given a positive integer \( k < 2^N \), let \( k_Nk_{N-1} \cdots k_1 \) represent the binary representation of \( k \), and \( k_n \) represent the number in the \( n \)th bit. Denote \( X'_k \) as a subset of variables in \( X \) such that \( X'_k = \bigcup_{k_n=1} \{x_n\} \). Hence, \( X'_k \in \mathcal{P}(X) \), where \( \mathcal{P}(X) \) is the power set of \( X \). For simplicity, we use \( \mu_k \) to denote \( \mu(X'_k) \).

From the viewpoint of cross-layer design, the interaction measure \( \mu(X'_k) \) can also be regarded as the significance measure \([41]\) of each design variable set \( X'_k \) on the cross-layer design objective. The sign of \( \mu(X'_k) \) indicates the positive or negative effect of increasing the values of the design variables in \( X'_k \) on the design objective \( Y \). In particular, the positive sign of \( \mu(X'_k) \) indicates that increasing the values of the variables in \( X'_k \) will increase \( Y \), and vice versa. Likewise, the negative sign of \( \mu(X'_k) \) indicates that increasing the values of the variables in \( X'_k \) will reduce \( Y \), and vice versa. The absolute value of \( \mu(X'_k) \) indicates the significance of set \( (X'_k) \). A large \( |\mu(X'_k)| \) suggests that the design objective \( Y \) is more sensitive to the change of the variables in \( X'_k \).

The concept of the interaction measure can be explained by Figure 5.2, which shows that in a given cross-layer design with three design variables, \( X = \{x_1, x_2, x_3\} \), different subsets of design variables belonging to \( X \) may have different contributions to the design objective. For instance, as shown in Figure 5.2, \( \mu_7 \) refers to the interaction among parameters \( \{x_3, x_2, x_1\} \), while \( \mu_1 \) refers to the significance of parameter \( \{x_1\} \) to the design objective.

In non-additive measure theory, a measure is defined as additive if \( \mu(A \cup B) = \mu(A) + \mu(B) \), wherever \( A \subset X \), \( B \subset X \), and \( A \cap B = \emptyset \). For the method of the interaction measure defined above, due to the complicated interdependency among different design variables with various uncertainties in communication networks, the interaction measure for most of cross-layer systems is non-additive. The non-additive interaction measure can be observed from Figure 5.2, where \( \mu_1(\{x_1\}) + \mu_4(\{x_3\}) \neq \mu_5(\{x_3, x_1\}) \), meaning that the joint contribution of design variables \( \{x_3, x_1\} \) to the design objective is not equal to the
sum of the individual contributions made by \( \{x_3\} \) and \( \{x_1\} \).

**Definition 2**

The interaction between variable sets \( A \) and \( B \) is called positive interaction if \( |\mu(A \cup B)| > \max\{|\mu(A)|, |\mu(B)|\} \) where \( A \subset X \), \( B \subset X \), and \( A \cap B = \emptyset \). Similarly, if \( |\mu(A \cup B)| < \max\{|\mu(A)|, |\mu(B)|\} \) it is called negative interaction.

The positive interaction among the design variables in \( X'_k \) implies that the joint optimization of these design variables brings more significant effect on the design objective than the optimization of each individual design variable. Similarly, the negative interaction among the design variables in \( X'_k \) implies that the joint optimization of these design variables brings less significant effect on the design objective than the optimization of each individual design variable. The goal of cross-layer design is to determine and take advantage of the positive interaction \( \mu(X'_k) \). If the design objective \( Y \) needs to be maximized as in (5.1), we should increase the values of the variables in \( X'_k \) when \( \mu(X'_k) \) has a positive sign,
or decrease the values of the variables in $X'_k$ when $\mu(X'_k)$ has a negative sign. On the other hand, if the design objective $Y$ needs to be minimized, we should decrease the values of the variables in $X'_k$ when $\mu(X'_k)$ has a positive sign, or increase the values of the variables in $X'_k$ when $\mu(X'_k)$ has a negative sign.

5.2.3 The Proposed Non-Linear Regression Model for the Interaction Measure

Based on the definition of the interaction measure, we propose a non-linear regression model based on the non-additive measure theory for interaction measure. The unique feature of this model is that the impact of interactions among design variables on the design objective can be quantitatively measured through a nonlinear integral, namely the Choquet integral.

Choquet Integral

Before we go further, a brief introduction of the Choquet integral is necessary. The Choquet integral [122] is a generalization of the Lebesgue integral, defined over a set of non-additive measures (a.k.a. fuzzy measures). Let $A = \{a_1, a_2, \ldots, a_N\}$ be a set of attributes, $f(a)$ be the observed or partially evaluated value on each attribute $a \in A$, $f$ be a tuple of observed or partially evaluated values on $A$, and $z$ be an objective. The linear/additive multivariate regression model is traditionally represented as a weighted sum $z = \sum_{a \in A} w_a f(a)$, where the weight $w_a$ is also regarded as a Lebesgue measure $w$ on a singleton $\{a\}$, since the linear model is equivalent to a Lebesgue integral $z = (L) \int_{A} f dw$. The Choquet integral model breaks the restriction that the combined contribution of $\{a_i, a_j\}$ towards to the objective $z$ is the weighted sum of their respective contributions. Instead, it uses a non-additive measure $v$, which is defined over the power set of $A$, and the Choquet integral, $z = (C) \int_{A} f dv$. Therefore, it is more powerful than the Lebesgue integral model since the non-additive measure $v$ considers the interaction among attributes towards to the objective. In such a setting, $v(\{a_i, a_j\})$ may not be a linear sum of $v(\{a_i\})$ and $v(\{a_j\})$; the Lebesgue integral model thus becomes a special case of the Choquet integral model where the linear sum
equation holds.

The Nonlinear Multivariate Regression Model

To calculate the interaction measure $\mu_k$ as defined in Section 5.2.2, we introduce a new nonlinear multivariate statistical regression model based on the aforementioned Choquet integral. For a given cross-layer design problem (5.1), the design variables $X$ can be considered as the attributes $A$ of the Choquet integral, while the design objective $Y$ is corresponding to the objective $z$ in the Choquet integral. The proposed nonlinear model of the interaction measure is described as follows.

Assume that the observed data under cross-layer design constraints in (5.1) consists of $Q$ observations of $\{x_1, x_2, \ldots, x_N\}$ and $Y$ as

\[
\begin{array}{ccccccc}
  x_1 & x_2 & \cdots & x_N & Y \\
  f_{11} & f_{12} & \cdots & f_{1N} & y_1 \\
  f_{21} & f_{22} & \cdots & f_{2N} & y_2 \\
  \vdots & \vdots & \cdots & \vdots & \vdots \\
  f_{q1} & f_{q2} & \cdots & f_{qN} & y_q \\
  \vdots & \vdots & \cdots & \vdots & \vdots \\
  f_{Q1} & f_{Q2} & \cdots & f_{QN} & y_Q
\end{array}
\]  

(5.2)

where each row is an observation of design variables $\{x_1, x_2, \ldots, x_N\}$ and the design objective $Y$. The observation of $\{x_1, x_2, \ldots, x_N\}$ can be regarded as a nonlinear function $f : X \rightarrow (-\infty, +\infty)$; hence the $q$th observation of $\{x_1, x_2, \ldots, x_N\}$ is denoted by $f_q$ and $f_{qn} = f_q(x_n)$, where $1 \leq n \leq N$ and $1 \leq q \leq Q$.

As defined in Section 5.2.2, interactions among the design variables $X$ towards to the design objective $Y$ is described by a set function $\mu$ defined on the power set of $X$ satisfying the condition of vanishing at the empty set, i.e., $\mu : P(X) \rightarrow (-\infty, +\infty)$ with $\mu(\emptyset) = 0$, where the set function $\mu$ is a non-additive measure. Based on the Choquet integral, the relationship between the design objective $Y$ and the interaction measure $\mu$ can be described by a new nonlinear multivariate regression model [124]:

\[
Y = e + \int_{(c)} f d\mu + N(0, \delta^2),
\]  

(5.5)
Algorithm 1:

1. Construct the $Q \times (2^N + 1)$ augmented matrix $Z = [z_{qk}]$ as follows.

$$
\begin{align*}
    z_{q0} &= 1 \\
    z_{qk} &= \begin{cases} 
        \min_{n=1} \max_{n=0} (f_{qn}) - \max_{n=0} (f_{qn}) & : \text{if it is }>0 \\
        0 & : \text{otherwise}
    \end{cases} \\
    z_{q2^N} &= y_q
\end{align*}
$$

(5.3)

where $k = 1, 2, \ldots, 2^N - 1$ and $q = 1, 2, \ldots, Q$.

2. Find the least-square solution to the linear equations having the above augmented matrix for unknown variables $e, \mu_1, \mu_2, \cdots, \mu_{2^N - 1}$. The standard approach can be found from textbooks covering the least-square and augment matrix [123].

3. The regression residual error $\hat{\delta}^2$ can be calculated by:

$$
\hat{\delta}^2 = \frac{1}{Q} \sum_{q=1}^{Q} (y_q - e - \sum_{k=1}^{2^N - 1} z_{qk}\mu_k)^2.
$$

(5.4)

Figure 5.3 : The algorithm to calculate the interaction measure $\mu_k$ in cross-layer design.

where $e$ is a regression constant, $\int_{(c)}$ is the Choquet integral, $f$ is an observation of ${x_1, x_2, \ldots, x_N}$, $N(0, \delta^2)$ is a normally distributed random perturbation with expectation 0 and variance $\delta^2$, and $\delta^2$ is the regression residual error.

Given the observation data, the optimal regression coefficients $\mu$ can be determined by using the least square method in order to make $\delta^2$ minimal. Recall that we use $\mu_k$ to denote $\mu(X_k')$ where $X_k'$ is a subset of $X$, i.e. $X_k' \in \mathcal{P}(X)$. The method to determine $\mu_k(k = 1, 2, \cdots, 2^N - 1)$ has been described by Algorithm 1 as shown in Figure 5.3 [40].

Once $\mu_k$ has been determined, we can identify which subset of design variables has the most significant impact on the design objective. So we can further fine-tune those design variables to improve the system performance under the current system and network conditions.
5.3 The Proposed Theoretical Framework for Sensitivity Analysis with Incommensurable Observation Data

So far, we have introduced a theoretical framework on the quantitative interaction measure in cross-layer design, where the significance of each subset of design variables is calculated in terms of their contributions on the design objective function. However, it is also important to quantify how sensitive the design objective function will be when design variables change. Figure 5.4 shows the performance sensitivity of the design objective $Y$ with different design variables $x_1$ and $x_2$. As shown in Figure 5.4(a), the design objective $Y$ is assumed to be a binary function of two design variables $x_1$ and $x_2$. If we observe the function in the coordinate systems ($Y$ vs. $x_1$) in Figure 5.4(b) and ($Y$ vs. $x_2$) in Figure 5.4(c), respectively, we find that the performance of $Y$ is more sensitive to $x_2$ than to $x_1$, i.e., $\frac{\Delta Y}{\Delta x_1} \ll \frac{\Delta Y}{\Delta x_2}$.

In cross-layer design, it is very important to capture such sensitivity features, not only for selecting a proper operating point of a design variable, but also for determining which subset of design variables should be adjusted to obtain the optimal design performance under a given set of network conditions.

Furthermore, for commensurable datasets, the theoretical framework on the quantitative interaction measure introduced in Section 5.2.3 can directly give the significance of each design variable on the design objective. However, in many cases, design variables are residing in different network layers and are incommensurable. In other words, these design variables are of different units, making it extremely difficult to directly apply the proposed framework of the interaction measure for sensitivity analysis.

In order to overcome the incommensurability problem, a new data preprocessing procedure was introduced in [41], where all observations of design variables were first normalized by their minimum values, and then data fitting was applied to find the best-possible nonlinear relations between the normalized observations of both the design objective and design variables. However, normalizing the multidimensional observation data does not consider
the possible difference on the dynamic range of design variables. For example, assume that a cross-layer scheme is designed to enhance the link adaptation of wireless networks by jointly adjusting the Modulation and Coding scheme (MCS) [64] of the physical layer and the packet size of the data link layer. Also, assume that the collected operating points of the MCS are Bit Error Rate (BER), say, \(10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\), and the collected operating points of the packet size are \((300, 600, 900, 1200, 1500)\) bytes. Then, if the normalization method proposed in [41] is applied to the above observations, the normalized values will be \((10^4, 10^3, 10^2, 10, 1)\) and \((1, 2, 3, 4, 5)\). Thus, there still exists a big difference in the order of magnitude between the two normalized value sets, which makes it infeasible
for an accurate calculation of the interaction between MCS and packet size.

In this work, we propose a derivative-based method, in which the derivatives of the design objective with respect to design variables are used to analyze interactions among and perform sensitivity analysis on design variables. The derivatives are calculated by data fitting observations and by carrying out functional analysis. Additionally, we also propose a pseudo-derivative-based method for those scenarios where the derivatives of the design objective with respect to design variables are not easy to be acquired directly through data fitting.

### 5.3.1 Derivative-Based Sensitivity Analysis

From the standpoint of system analysis, the sensitivity of system response can be captured by using partial derivatives of the system output (design objective function) $Y$ with respect to the system parameters (design variables) $x_n$, i.e., $\frac{\partial Y(X)}{\partial x_n}$. However, in many cases, it is not feasible to derive an analytical expression of the design objective function $Y(X)$ between the system output $Y$ and the system input $X$. Therefore, in this work, we will estimate the relation between $Y$ and each of $\{x_1, x_2, \cdots, x_N\}$ by using the observations of $Y$ and the system parameters $\{x_1, x_2, \cdots, x_N\}$.

Let $x_n$ and $y$ be the observation vectors of $Q$ data observations of design variable $x_n$ and design objective $Y$, respectively, i.e., $x_n := \{x_{1n}, x_{2n}, \cdots, x_{Qn}\}$ and $y := \{y_1, y_2, \cdots, y_Q\}$. Figure 5.5 depicts the derivative calculation of design objective $Y$ with respect to design variable $x_n$, given the observation vectors $x_n$ and $y$. In particular, $Y_n$ can be constructed to describe the relation between $x_n$ and $Y$ by data fitting the corresponding observations $x_n$ and $y$.

Here, one question arising from data fitting is how to choose the right type of fitting function (e.g., polynomial, exponential, logarithmic, or other functions). In cross-layer design, one observed value of design variable $x_n$ may correspond to multiple different observed values of design objective $Y$ due to interactions with other design variables $\{x_1, x_2, \cdots,$
Figure 5.5: Illustration of the derivative calculation of the design objective $Y$ with respect to the design variable $x_n$ by using the observation vectors $x_n$ and $y$.

To describe the one-to-multiple mapping between design variable $x_n$ and design objective $Y$, data fitting is performed on the data-set $x_n$ and $y$ by using the rational function $\xi(y)$ in the form of

$$
x_n = \xi(y) = \frac{c_u y^u + c_{u-1} y^{u-1} + \cdots + c_1 y + c_0}{d_v y^v + d_{v-1} y^{v-1} + \cdots + d_1 y + d_0}.
$$

The selection of order $u$ and $v$ in Equation (5.6) depends on the desired accuracy of approximation and acceptable computational complexity for a given cross-layer design. Meanwhile, the selection of $u$ and $v$ also determines the degree $\epsilon$ of function $\xi(y)$ as $\epsilon = \max\{u, v\}$.

The coefficients $\{c_u, c_{u-1}, \cdots, c_0\}$ and $\{d_v, d_{v-1}, \cdots, d_0\}$ can be determined by solving a least-square problem. Note that in this chapter without losing generality, we adopt rational functions. Similar results can be derived when other fitting functions such as B-splines [125] and P-spline [126] are adopted.

With the derived polynomial coefficients $\{c_u, c_{u-1}, \cdots, c_0\}$ and $\{d_v, d_{v-1}, \cdots, d_0\}$, Equation (5.6) can be solved in terms of the analytical expression of $y$, which provides the inverse function of polynomial $\xi(y)$. However, if the degree of $\xi(y)$ is $\epsilon$, Equation (5.6) may have $\epsilon$ different roots in $y$. Let $G(x_n)$ denote the set of $\epsilon$ roots as $G(x_n) := \{g_s(x_n) | 1 \leq s \leq \epsilon \text{ and } s \in$
Algorithm 2:

1. For design variable \( x_n \), perform data fitting based on the observations \( x_n \) and \( y \), and determine the optimal fitting function \( Y_n \).

2. Calculate the derivative \( \frac{\partial Y_n}{\partial x_n} \bigg|_{x=x_{qn}} \).

3. Apply Algorithm 1 to the derivatives by substituting \( \frac{\partial Y_n}{\partial x_n} \bigg|_{x=x_{qn}} = f_{qn} \) for \( f_{qn} \) to calculate \( e, \mu_1, \mu_2, \cdots, \mu_{2N-1} \).

Figure 5.6: The algorithm for sensitivity analysis by using the derivative-based method.

Define \( g_{sn} \) as the output vector of function \( g_s(x_n) \) on the input value \( x_n \). In order to derive the optimal data fitting function based on data points \( \{(x_{1n}, y_1), (x_{2n}, y_2), \cdots, (x_{Qn}, y_Q)\} \), the Euclidean norm of the vector difference \( (g_{sn} - y) \) is calculated. Thus, the function \( g^*_s(x_n) \) can be determined as

\[
Y_n := g^*_s(x_n) = \arg \min_{G(x_n)} \| g_{sn} - y \|_2. \tag{5.7}
\]

The optimal fitting function \( Y_n \) characterizes the relationship between design variable \( x_n \) and design objective \( Y \).

After deriving all fitting functions \( \{Y_1, Y_2, \cdots, Y_N\} \) of design variables \( \{x_1, x_2, \cdots, x_N\} \), the derivatives of these functions with respect to \( x_n \) can be calculated, and the following data structure can be developed:

\[
\begin{array}{ccccccc}
  & x_1 & x_2 & \cdots & x_N & Y \\
\hline
\frac{\partial Y_1}{\partial x_1} \bigg|_{x=x_{11}} & \frac{\partial Y_2}{\partial x_1} \bigg|_{x=x_{12}} & \cdots & \frac{\partial Y_N}{\partial x_1} \bigg|_{x=x_{1N}} & y_1 \\
\frac{\partial Y_1}{\partial x_2} \bigg|_{x=x_{21}} & \frac{\partial Y_2}{\partial x_2} \bigg|_{x=x_{22}} & \cdots & \frac{\partial Y_N}{\partial x_2} \bigg|_{x=x_{2N}} & y_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial Y_1}{\partial x_{q1}} \bigg|_{x=x_{q1}} & \frac{\partial Y_2}{\partial x_{q1}} \bigg|_{x=x_{q2}} & \cdots & \frac{\partial Y_N}{\partial x_{q1}} \bigg|_{x=x_{QN}} & y_q \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial Y_1}{\partial x_{Q1}} \bigg|_{x=x_{Q1}} & \frac{\partial Y_2}{\partial x_{Q1}} \bigg|_{x=x_{Q2}} & \cdots & \frac{\partial Y_N}{\partial x_{Q1}} \bigg|_{x=x_{QN}} & y_Q \\
\end{array} \tag{5.8}
\]

With these derivatives, \( e, \mu_1, \mu_2, \cdots, \mu_{2N-1} \) can be determined by using Algorithm 1. The complete derivative-based method is summarized as Algorithm 2 in Figure 5.6.
Note that by using the derivatives of design objective over design variables, the calculated $\mu_1, \mu_2, \cdots, \mu_{2^N-1}$ by Algorithm 2 can not only explain the interactions among different sets of design variables, but also can analyze the sensitivity of the design objective with respect to the changes of design variables.

### 5.3.2 Pseudo-Derivative-based Sensitivity Analysis

The proposed derivative-based sensitivity analysis is effective when explicit or well-approximated functional relations between design objective $Y$ and design variables $X$ by data fitting are available. However, for some cross-layer design, data fitting may not be a feasible solution due to swarm patterns in the observed data. To overcome this problem, two pseudo-derivative-based methods are proposed for swarm observation patterns.

#### Dense Swarms of Observations

In some scenarios of cross-layer design, for some observations of design variable $x_n$, the variation of design objective $Y$ may be relatively small due to interactions with other design variables. Such observations are defined as dense swarms of observations as illustrated in Figure 5.7. Let $\overline{y}_{q'n}$ and $\overline{x}_{q'n}$ be the mean values of the observations of $Y$ and $x_n$, corresponding to the $q'$th data swarm. The method to obtain the interaction measure among design variables can be summarized as Algorithm 3 in Figure 5.8.

Note that various statistical models can be considered in data fitting, such as linear (e.g., $a + bx$), quadratic (e.g., $a + bx + cx^2$), logarithmic (e.g., $a + b \log(x + c)$), and combined (e.g., $\exp^{a+bx} + c$). The data fitting accuracy can be calculated by comparing the resulting mean square deviations, and a smaller mean square deviation indicates a more accurate data fitting.
Algorithm 3:

1. For each data swarm \( q' (q' = 1, 2, \ldots, Q') \), calculate the mean values \( \overline{y}_{q'n} \) and \( \overline{x}_{q'n} \) of the observations of both design objective \( Y \) and design variable \( x_n \) in the data swarm, respectively.

2. Perform data fitting on the data sets \( \{ \overline{y}_{1n}, \overline{y}_{2n}, \ldots, \overline{y}_{q'n}, \ldots, \overline{y}_{Q'n} \} \) and \( \{ \overline{x}_{1n}, \overline{x}_{2n}, \ldots, \overline{x}_{q'n}, \ldots, \overline{x}_{Q'n} \} \) to obtain the functional relationship \( Y = f(x_n) \).

3. With the function \( Y = f(x_n) \), calculate the derivative \( \frac{\partial Y}{\partial x_n} |_{x_n=x_{q'n}} \).

4. Apply Algorithm 1 to the derivatives by substituting \( \frac{\partial Y}{\partial x_n} |_{x_n=x_{q'n}} \), \( n = 1, 2, \ldots, N \) for \( f_{qn} \) to calculate \( e, \mu_1, \mu_2, \ldots, \mu_{2^N-1} \).

Figure 5.8: The algorithm for sensitivity analysis in cross-layer design with dense swarms of observations of design objective \( Y \) and design variables \( x_1, x_2, \ldots, x_N \).

Sparse Swarms of Observations

Compared with the dense swarms of observations, for some observations of design variable \( x_n \), the variation of design objective \( Y \) may be relatively large due to interactions with
Figure 5.9: The sparse swarms of observations of design objective $Y$ and design variable $x_n$.

other design variables. Such observations are defined as sparse swarms of observations as illustrated in Figure 5.9.

To ease the discussion, let $\overline{x}_{q'}$ be the mean value of the observations of $x_n$ in the $q'$th data swarm. In Figure 5.9, we observe that for the second data swarm adjusting design variable $x_n$ in the neighboring area of $\overline{x}_{2n}$ can only bring a small variation of the resulting design objective. In other words, design objective $Y$ is not sensitive to $x_n$ in the vicinity of $\overline{x}_{2n}$. On the other hand, adjusting $x_n$ in the neighboring area of $\overline{x}_{Q'n}$ can generate a significant variation of design objective $Y$, meaning that design objective $Y$ is very sensitive to $x_n$ in the vicinity of $\overline{x}_{Q'n}$. To capture the degree of deviation of $Y$ at some value of $x_n$, the variance of $Y$ and the mean of $x_n$ for each data swarm are calculated. The method to derive the interactions among design variables can be summarized by Algorithm 4 in Figure 5.10.

The interaction measure $\{\mu_1, \mu_2, \cdots, \mu_{2N-1}\}$ calculated by any of Algorithms 1 to 4 gives the sensitivity of the design objective responding to the design variables. Moreover, with
Algorithm 4:

1. Calculate the variance $\hat{y}_{q'n}$ of the observations of $Y$ in each data swarm $q'$ as $\hat{y}_{q'n} = \frac{1}{N q'} \sum_{p=1}^{N_{q'}} (y_p - \overline{y}_{q'n})$; calculate the mean value $\overline{x}_{q'n}$ of the observations of $x_n$ in each swarm $q'$.

2. Apply Algorithm 1 to variances $\hat{y}_{q'n}$ and means $\overline{x}_{q'n}$ to calculate $e$, $\mu_1$, $\mu_2$, $\ldots$, $\mu_{2N-1}$.

Figure 5.10: The algorithm for sensitivity analysis in cross-layer design with sparse swarms of observations of design objective $Y$ and design variables $x_1$, $x_2$, $\ldots$, $x_N$.

the interaction measure $\mu_i$, we can identify the most important subset of design variables to improve the design objective effectively.

5.4 A Case Study: Cross-Layer Optimized Multimedia Delivery over Wireless Networks

In this section, we adopt cross-layer optimized wireless multimedia communications as a case study to illustrate the major cross-layer design tradeoffs and validate the proposed framework of interaction measure and sensitivity analysis. The study mainly consists of two parts: 1) we will illustrate the design challenge of modeling and analyzing cross-layer design tradeoffs under uncertainties such as temporal and spatial behaviors; 2) we will present analytical and experimental results to validate the proposed sensitivity analysis framework.

5.4.1 Experiment Environment

In the protocol stack of wireless multimedia, each layer has one or multiple key design variables which significantly affect the overall design objective. For instance, at the application layer, prediction mode and quantization parameter (QP) in video encoding are two critical design variables [127]. At the data link layer, automatic repeat request (ARQ), media access control protocols, and packetization are often used to maintain a low packet loss rate. At
the physical layer, modulation and coding schemes (MCS) have been adopted to achieve a good tradeoff between transmission rate and transmission reliability.

In this case study, we investigate the real-time transmission of an individual video bitstream across a multi-hop 802.11a/e wireless network, in which contention-free access to the medium provided by the HCF controlled channel access protocol (HCCA) [82] is assumed. The system model for the cross-layer optimized multimedia transmission is shown in Figure 5.11, where the expected received video quality is used as the design objective, and the design variables include QP at the application layer and MCS at the physical layer. It is assumed that the controller is able to acquire the corresponding system information, such as the expected video distortion from the encoder and the network conditions from lower layers by interacting with each layer. Besides the cross-layer optimization capability, the controller is also assumed to be able to derive interactions among design variables and perform sensitivity analysis based on the collected network data to improve the cross-layer design.
In the latest H.264 standard, the allowed QP values are \{0, 1, 2, \ldots, 51\}. To achieve a good tradeoff between performance and computational complexity, without loss of generality, in this work we choose \{5, 7, 9, \ldots, 45\} as the operating points of the design variable QP. For another design variable MCS, without losing generality, the schemes in Table 5.1 specified in 802.11a networks are adopted as its operating points.

Let II = \{π₁, π₂, \ldots, πₐ\} be the set of I packets that compose the current video frame to be transmitted. Let \(E[D_{i}]\) be the expected distortion of packet \(π_{i}\). To provide a smooth video display experience to end users, each frame is associated with a frame decoding deadline \(T_{\text{budget}}\) [26, 85]. \(T_{\text{budget}}\) imposes a delay constraint on the transmission of each packet composing the current frame as

\[
\sum_{i=1}^{I} T_{i} \leq T_{\text{budget}},
\]

where \(T_{i}\) is the end-to-end delay of packet \(π_{i}\) transmitted from the sender to the receiver. More details on how to calculate \(T_{i}\) can be found in [44].

Let \(s_{l}\) denote the QP for the \(l\)th coding unit of the current frame and \(a_{i}\) the MCS for transmitting packet \(π_{i}\). Here, the coding unit could be a video frame, a slice, or a macroblock, depending on different adaptation time intervals, which will be explained in more details in Section 5.4.2. Denote \(S\) and \(A\) as the sets of all operating points of \(s_{l}\) and

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Modulation</th>
<th>Coding Rate</th>
<th>Spectral Efficiency (bits/sym.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BPSK</td>
<td>1/2</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>BPSK</td>
<td>3/4</td>
<td>0.75</td>
</tr>
<tr>
<td>3</td>
<td>QPSK</td>
<td>1/2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>QPSK</td>
<td>3/4</td>
<td>1.5</td>
</tr>
<tr>
<td>5</td>
<td>16-QAM</td>
<td>1/2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>16-QAM</td>
<td>3/4</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>64-QAM</td>
<td>2/3</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>64-QAM</td>
<td>3/4</td>
<td>4.5</td>
</tr>
</tbody>
</table>
respectively. We assume that $|S| = S$ and $|A| = A$, where $|S|$ is the cardinality of the set $S$. Thus, the goal of cross-layer design is to find the optimal operating point vector \{\(s_i, a_i\)\} such that the received video distortion is minimized under the constraint of packet delay deadline, i.e.,

$$
\min_{\{s_i \in S, a_i \in A\}} \sum_{i=1}^{I} E[D_i] \\
\text{s.t.} : \sum_{i=1}^{I} T_i \leq T_{\text{budget}}.
$$

The experiments are designed using H.264/AVC JM 12.2 [98]. We encode the Y-component of the first 150 frames of the QCIF video sequence “Foreman” at different frame rates. More information on the experiment setting and the solution procedure can be found in [43] and is omitted here to conserve space.

5.4.2 Illustration of Major Cross-Layer Design Tradeoffs

Impact of Temporal Behavior

In the case study, to understand the temporal behavior of cross-layer design, we will observe how the selection of the temporal resolution $\tau$ in cross-layer optimization can affect the received video quality by jointly optimizing QP and MCS at the selected temporal resolution. We consider the following three temporal resolutions of adaptation: frame, slice, and macroblock. The temporal relation among frame, slice, and macroblock are depicted in Figure 5.12, where $\alpha$ represents the maximum number of slices of a frame and $\beta$ is the maximum number of macroblocks of a slice. As shown in Figure 5.12, the frame-level temporal resolution is the coarsest temporal resolution while the macroblock-level temporal resolution is the finest temporal resolution.

Figure 5.13 shows the comparison of the received video quality with three temporal resolutions under different channel conditions, where the video frame rate is 30 frames per second. It can be observed that finer temporal resolutions bring better video quality, due to
the fact that cross-layer design using finer temporal resolutions leads to finer quantization of the manifold of the design objective function and reduces the sampling error in the discretization process. However, cross-layer design using finer temporal resolutions will introduce much higher computational complexity than that using coarser ones. Therefore, proper selection of temporal resolution in cross-layer design is necessary to achieve a good tradeoff between the performance gain and the computational complexity.
Impact of Feedback Delay

We have evaluated the impact of feedback delay on the performance of cross-layer design under different temporal resolutions. We have considered two types of wireless channels: slow-fading channel and fast-fading channel. In the slow-fading channel, the channel SINR is assumed to remain invariant when transmitting a group of video frames. In the fast-fading channel, the channel SINR varies frame by frame but remains time-invariant when transmitting a single video frame. A finite-state Markov channel model with different fading speed [128] is used to simulate both the slow-fading and fast-fading channels.

Figure 5.14 shows the received video quality of cross-layer design with different video frame rates and under different channel conditions. We observe that PSNR decreases with the increase of video frame rate $f_r$, especially for the cross-layer design using the frame-level temporal resolution. This is because a larger video frame rate $f_r$ leads to a tighter delay constraint to make video packets available at the receiver in time for playback. We also observe that the cross-layer design using the frame-level temporal resolution has a big
PSNR performance difference over different channels, while the one using the slice-level temporal resolution produces almost the same PSNR performance. This is because using the frame-level temporal resolution will overlook the fast-fading channel, and the SINR feedback adopted in cross-layer design cannot reflect the true SINR values of the underlying fast-fading channels due to the feedback delay. In contrast, when using the slice-level temporal resolution, cross-layer design uses the true SINR values of the underlying fast-fading channels, which brings better PSNR performance. However, it should be noted that this better PSNR performance is achieved at the cost of a higher SINR feedback frequency. Therefore, there is a design tradeoff between the performance gain and the signaling cost in cross-layer design.

**Impact of Spatial Behavior**

To illustrate the spatial behavior of cross-layer design, two operating point sets of the design variable QP with different discretization levels are considered. One set has finely-discretized QP values, denoted by $\mathcal{S}_1 = \{5, 7, 9, \ldots, 45\}$ and $|\mathcal{S}_1| = 21$, and another set has coarsely-
discretized QP values, denoted by $S_2 = \{5, 10, 15, \ldots, 45\}$ and $|S_2| = 9$. Figure 5.15 shows the comparison of the received video quality of the cross-layer design using the slice-level temporal resolution. We find that the cross-layer design using $S_1$ brings better video quality than that using $S_2$. This is because finely-discretized QP set $S_1$ provides more flexibility and adaptation to video coding than coarsely-discretized QP set $S_2$. However, the cross-layer design using $S_1$ introduces much higher computational complexity than that using $S_2$. Therefore, proper selection of operating sets of design variables in cross-layer design is necessary to achieve a good tradeoff between the performance gain and the computational complexity.

### 5.4.3 Validation of the Proposed Theoretical Framework

In the following, we will validate the proposed sensitivity analysis framework for cross-layer design. The observations of the design variables QP and MCS are their operating points, respectively. The observations of the design objective are the values of the distortion corresponding to all of the possible operating point combinations of the two design variables. Both good (SINR=30 dB) and bad (SINR=15 dB) channels are considered. Since there are two design variables QP and MCS (denoted by $s$ and $a$, respectively), we have three interaction measures that are $\mu(\{s\})$, $\mu(\{a\})$, $\mu(\{s, a\})$. Based on the proposed framework of sensitivity analysis in Section 5.3, the calculated interaction measures are shown in Table 5.2. The sign of $\mu(X_k')$ in Table 5.2 indicates whether the interaction among the design variables in $X_k'$ has a positive or negative effect on the received video distortion. In other words, the positive sign of $\mu(X_k)$ indicates that increasing the values of the design variables in $X_k'$ can increase the distortion, and vice versa. Likewise, the negative sign of $\mu(X_k)$ indicates that the distortion will be reduced when increasing the values of the design variables in $X_k'$, and vice versa.

To validate the proposed framework of interaction measure and sensitivity analysis, the relations between PSNR and the design variables QP and MCS under different channels
Table 5.2: The calculated interaction measures of the design variables QP and MCS in the case study.

<table>
<thead>
<tr>
<th></th>
<th>SINR=30dB</th>
<th>SINR=15dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu({s})$</td>
<td>4.306</td>
<td>29.621</td>
</tr>
<tr>
<td>$\mu({a})$</td>
<td>-0.311</td>
<td>-0.123</td>
</tr>
<tr>
<td>$\mu({s, a})$</td>
<td>4.665</td>
<td>27.791</td>
</tr>
</tbody>
</table>

Figure 5.16: The received video quality with different operating points of both QP and MCS under different channel conditions.

are plotted in Figures 5.16. The relations between PSNR and each design variable under different channels are plotted in Figures 5.17 and 5.18.

The interaction measures $\mu(\{s\})$, $\mu(\{a\})$, and $\mu(\{s, a\})$ in Table 5.2 well match the results as shown in Figures 5.16 to 5.18. In Table 5.2, under the good channel, $|\mu(\{s\})|$ is larger than $|\mu(\{a\})|$, meaning that the received video quality is more sensitive to QP than to MCS. As shown in Figure 5.17(a), all curves are sharply dropped, which means that adjusting QP can significantly affect the design performance. Compared with the relation between QP and PSNR as shown in Figure 5.17(a), the relation between MCS and PSNR as shown in Figure 5.17(b) does not change much, meaning that MCS is not sensitive or not important to the design performance under the given network environment.
Figure 5.17: The received video quality with different operating points of both QP and MCS under the good channel.

Under the scenario of bad channel, the design performance is more sensitive to QP than to MCS. The relation between QP and PSNR as shown in Figure 5.18(a) contains several convex curves, while the relation between MCS and PSNR as shown in Figure 5.18(b) contains heavy tailed and even flattened curves, meaning that the performance gain of adjusting QP is much larger than adjusting MCS.

From the experimental results significant positive interactions among design variables can also be observed. For instance, under good channels, we can observe $\mu(\{s, a\}) > \mu(\{s\}) + \mu(\{a\})$, meaning that under good channel conditions, the distortion performance is more sensitive to the joint optimization of QP and MCS than to the individual optimization of either variable. Furthermore, under either good or bad channels, we can observe $|\mu(\{s, a\})| > |\mu(\{a\})|$, which shows that the distortion performance is more sensitive to the joint optimization of QP and MCS than to the individual optimization of MCS under all channel conditions.

To illustrate the quality of the reconstructed videos, we show a sample video frame from the original video in Figure 5.19(a) and compare it with the reconstructed video frames by
jointly adjusting QP and MCS (Figure 5.19(b)), by only adjusting QP (Figure 5.19(c)) and by only adjusting MCS (Figure 5.19(d)). When only adjusting QP, all the operating points of QP are considered, and only an operating point of MCS with a median value in terms of the spectral efficiency in Table 5.1 is adopted. Similarly, when only adjusting MCS, all the operating points of MCS are considered, and only an operating point of QP with a median value in [5, 45] is used. When jointly adjusting QP and MCS, all the operating points of QP and MCS are considered. We can observe that the frame derived by the joint optimization of QP and MCS has a visual quality very close to the original frame and much better than the frames in Figures 5.19(c) to 5.19(d) derived by the individual optimization of one single design variable, which verifies the inequality $\mu(\{s, a\}) > \mu(\{s\}) + \mu(\{a\})$ under good channels in Table 5.2. Moreover, the frame derived by the individual optimization of QP has a better visual quality than the one derived by the individual optimization of MCS, which verifies the inequality $|\mu(\{s\})| > |\mu(\{a\})|$ in Table 5.2.
5.5 Summary

One of the major challenges in cross-layer design is the lack of a theoretical foundation for the interaction measure and sensitivity analysis to gain in-depth understanding of temporal and spatial behaviors in cross-layer design. In this chapter, we have proposed a modeling and analysis framework to quantitatively characterize various tradeoffs in cross-layer design. A new concept called the interaction measure has been proposed to measure interaction among design variables and their impacts on the design objective function. We have also introduced a nonlinear multivariate regression model based on the non-additive measure theory for quantitative interaction measure. Furthermore, based on the interaction measure with incommensurable data sets, sensitivity analysis in cross-layer design has been studied. The proposed theoretical framework has been illustrated through a case study based on cross-layer optimized wireless multimedia communications. Extensive experiments have
been carried out to demonstrate the major design tradeoffs of cross-layer design and validate the proposed quantitative framework.
Chapter 6

Summary and Future Work

6.1 Summary of Research Contributions

In this dissertation, we investigated the problem of quality-driven cross-layer optimized wireless multimedia networking. The key contributions are as follows.

In chapter 2, we studied for the first time cross-layer optimized video summary transmission over single-hop wireless networks. Within a rate-distortion theoretical framework, the source coding, allowable retransmission, and adaptive modulation and coding have been jointly optimized, which reflects the joint selection of parameters at the physical, data link and application layers. The problem is formulated to achieve the best video quality and content coverage of the received summary frames and to meet the delay constraint. The problem is solved by using Lagrangian relaxation and dynamic programming. To evaluate the performance of the proposed framework and validate the benefit of our cross-layer design, we conducted extensive experiments by using H.264/AVC JM 10.2 and MATLAB. Our experimental results indicate the effectiveness and efficiency of the proposed optimization framework, especially when the delay budget imposed by the upper layer applications is small, where more than 10% distortion gain can be achieved. For the implementation of our framework, we proposed a system controller, whose responsibility is to communicate with each layer and dynamically determine the corresponding parameters that guarantee the best output video quality, and then drive the application system to perform efficiently. In addition, the proposed framework is quite generic and flexible for devices with various storage and computational capabilities.

Compared to the existing systems, the novelty of this work is in twofold: first, to our knowledge, this is the first cross-layer optimization framework proposed for the coding and
transmission of video summary data; second, AMC and ARQ are jointly considered in the cross-layer design, which acts a link adaptation scheme and gives the controller more flexibility in delivering the summary frames.

In Chapter 3, we extended our research for cross-layer optimized video communications over multi-hop wireless networks. We proposed an application-centric routing framework for real-time video transmission in multi-hop wireless networks. The multi-hop routing problem is formulated as the minimization of the expected end-to-end video distortion constrained by a predefined video packet delay deadline imposed by the video playback streaming application. To solve the problem, we developed a routing algorithm by utilizing the user-received video quality as the routing metric. The received video quality is evaluated as the expected end-to-end distortion. The expected distortion is accurately calculated in real-time at the source node by taking all related parameters into account, such as source codec parameters (e.g., quantization, packetization, and error concealment) and network parameters (e.g., throughput and delay). Queuing analysis was performed to ensure that the end-to-end delay constraint is satisfied. Both theoretical and experimental results demonstrate that the proposed quality-driven application-centric routing approach can achieve a superior performance over existing network-centric routing approaches. In consideration of the implementation of the proposed routing framework, we designed a cross-layer controller at the source node to optimize the performance of the entire system. We have also evaluated the computational complexity of the proposed framework, and pointed out that the framework could be flexibly adjusted to satisfy practical computational constraints.

The major contributions of this work are: 1) compared with the majority of video streaming research with pre-encoded videos, our work integrates online video coding with dynamic network routing path selection to achieve the best perceived video quality, and 2) the proposed application-centric routing metric, i.e., the expected end-to-end video distortion, is calculated on-the-fly in the process of routing in multi-hop wireless networks.

In chapter 4, we addressed cross-layer optimized video summary delivery over coopera-
tive wireless networks. The motivation is to fully exploit the advantages from cooperative communication and from video summarization. In this work, a novel decode-process-and-forward (DPF) scheme was proposed for video summary transmission, where a relay node with video processing capability is involved to generate a concise version of the summary frame, called summary of summary (SoS). The SoS information is effectively consumed by the destination side to enhance its error concealment capability, leading to an improved video reconstruction quality. We proposed a generic cross-layer optimization framework for cooperative video summary transmission, which jointly considers the source coding, relay processing parameters, power allocation between the source and relay nodes, and error concealment strategy to achieve the best video quality. The problem is solved by using Lagrangian relaxation and dynamic programming (DP). Experimental results show that the proposed scheme significantly outperforms the direct transmission scheme, the multipath scheme, and the decode-and-forward transmission scheme by up to 25%.

The novelty of this work is two-fold: it is the first cross-layer optimization framework for video summary transmission over cooperative wireless networks; the proposed cooperative relay scheme as well as the SoS processing model plays a critical role in improving the system performance of video summary transmission.

In chapter 5, we conducted a theoretical study of the methodology of cross-layer design and optimization. We first defined the concept of the interaction measure to quantitatively measure interactions among design variables towards to the design objective. Then, we adopted a nonlinear multivariate regression model to calculate the interaction measure based on network management data. To quantify how sensitive the design objective function will be when design variables change, a derivative-based sensitivity analysis algorithm based on quantitative interaction measure for both commensurable and incommensurable datasets was proposed. Finally, we conducted a case study on cross-layer optimized wireless multimedia communications to illustrate the major cross-layer design tradeoffs and validate the proposed theoretical framework. Both analytical and experimental results show the
correctness and effectiveness of the proposed framework. The proposed framework can significantly enhance our capability for cross-layer behavior characterization and provide insights for future design.

6.2 Future Work

6.2.1 Imperfect CSI and Channel Feedback Delay

In this dissertation, we have focused on cross-layer optimized wireless multimedia networking with the assumption that perfect CSI is available at the receiver and the feedback channel is error and latency free. The next step is to study the performance of the proposed cross-layer optimization frameworks under imperfect CSI and channel feedback delay, especially for mobile multimedia networks with fast-moving users.

6.2.2 Multi-Users and Multiple Antennas

In this dissertation, we have considered the scenario of single-user links with single transmit and single receive antenna. One possible direction is to generalize our work to situations with multi-users and multiple transmit and receive antennas. For example, video coding, transmission, and antenna selection could be jointly considered to perform optimal resource allocation among multi-users.

6.2.3 Content-Aware Video Communications

In this dissertation, we have proposed cross-layer optimization frameworks for wireless transmission of video summary to provide video content coverage and enhance user interaction. It would be very interesting to be able to perform adaptive source coding and transmission on different video contents (e.g., foreground and background), and apply this content-aware video communication strategy to resource-constrained wireless video applications, such as wireless video monitoring and surveillance. We are currently working on investigating a
content-aware video coding and transmission scheme for video surveillance over wireless sensor and actuator networks using pan-tilt-zoom cameras [56,129].

6.2.4 Power Consumption on Video Consumption and Transmission

In Chapter 4, we considered power consumption on video transmission. Mobile multimedia systems are receiving more research attention, where most existing mobile computing devices are powered by battery with limited energy resource. Therefore, the key to reduce the power consumption of such devices and improve the battery operating time is how to achieve the best tradeoff between the power consumption on computation and the power consumption on communication. We are currently working on both the analysis of power consumption of a H.264 video codec and the optimal power management of mobile computing devices in a cross-layer design fashion [130,131].

6.2.5 Derive the Analytical Expression of the Design Objective Function

In chapter 5, we derived an analytical expression of the design objective function \( Y(X) \) between the system output \( Y \) and the system input \( X \) by performing data fitting on data observations with the right type of fitting function (e.g., polynomial, exponential, logarithmic, or other functions). An interesting future direction is to derive the design objective function by using multivariate function basis and function approximation theory, where the design objective function is represented as a linear combination of basis functions.
Bibliography


[41] ———, “Significance measure with nonlinear and incommensurable observations,” in Proc. The IEEE Global Communications Conference (GLOBECOM), Nov. 2007.


[77] Y. Andreopoulos, N. Mastronade, and M. van der Schaar, “Cross-layer optimized


