

การจัดเรียงแม่โครบล็อคที่ยึดหยุ่นแบบปรับตัวได้ตามภาพสำหรับการเข้ารหัสและส่งวีดิทัศน์ H.264

ทนทานต่อความผิดพลาด



นาย เตียน หิว หงู

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2553

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

ADAPTIVE FRAME BASED FLEXIBLE MACROBLOCK ORDERING FOR
ERROR RESILIENT H.264 VIDEO CODING AND TRANSMISSION



Mr. Tien Huu Vu

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy Program in Electrical Engineering

Department of Electrical Engineering

Faculty of Engineering


Chulalongkorn University

Academic Year 2010

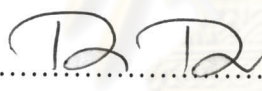
Copyright of Chulalongkorn University


Thesis Title ADAPTIVE FRAME BASED FLEXIBLE MACROBLOCK
ORDERING FOR ERROR RESILIENT H.264 VIDEO
CODING AND TRANSMISSION
By Mr. Tien Huu Vu
Field of study Electrical Engineering
Thesis Advisor Assistant Professor Supavadee Aramvith, Ph.D.
Thesis Co-Advisor Professor Yoshikazu Miyanaga, Ph.D.

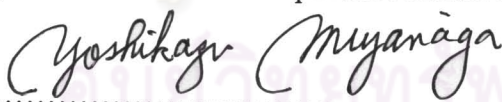
Accepted by the Faculty of Engineering, Chulalongkorn University in Partial
Fulfillment of the Requirements for the Doctoral Degree



..... Dean of the Faculty of Engineering
(Associate Professor Boonsom Lerdhirunwong, Dr.Ing.)

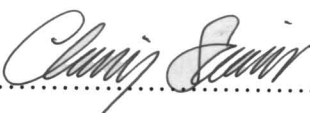
THESIS COMMITTEE


..... Chairman
(Professor Prasit Prapinmongkarn, Ph.D.)


..... Advisor
(Assistant Professor Supavadee Aramvith, Ph.D.)


..... Co-Advisor
(Professor Yoshikazu Miyanaga, Ph.D.)


..... Examiner
(Assistant Professor Suree Pumrin, Ph.D.)


..... Examiner
(Assistant Professor Chaichet Saivichit, Ph.D.)


..... External Examiner
(Associate Professor Kosin Chamnongthai, Ph.D.)

เตียน หิว หวู : การจัดเรียงแมโครบล็อกที่ยืดหยุ่นแบบปรับตัวได้ตามภาพสำหรับการเข้ารหัสและส่งวิดีโอ
 ทัศน์ H.264 ทนทานต่อความผิดพลาด (ADAPTIVE FRAME BASED FLEXIBLE MACROBLOCK
 ORDERING FOR ERROR RESILIENT H.264 VIDEO CODING AND TRANSMISSION)

อ.ที่ปรึกษาวิทยานิพนธ์: ผศ.ดร. สุภาวดี อารัมวิทย์, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม: ศ. ดร. โยชิคาซุ มิยามา
 กะ, 92 หน้า.

ในทุกวันนี้การส่งวิดีโอผ่านช่องสัญญาณไร้สายกำลังจะกลายเป็นสิ่งที่แพร่หลาย เมื่อมีการใช้
 ความก้าวหน้าของการเทคโนโลยีสื่อสารและข้อมูลสื่อประสมเข้าสู่การทำตลาด สิ่งนี้จะช่วยให้เกิดการ
 ประยุกต์การใช้งานที่ความต่อเนื่อง เช่น กลุ่มวิดีโอเคลื่อนที่ และ กระแสการส่งวิดีโอ ไร้สาย อย่างไรก็ตาม
 คุณภาพวิดีโอที่รับได้ดั้นนั้นเป็นประเด็นที่ท้าทายสำหรับการประยุกต์การส่งวิดีโอที่ใช้งานในทางปฏิบัติ
 เนื่องจากลักษณะเฉพาะความผิดพลาดสูงของช่องสัญญาณไร้สาย ที่เกิดจากผลกระทบหลายเส้นทาง และ เฟดลิก
 ดังนั้นจึงมีแนวความคิดเชิงวิจัยที่เผชิญความผิดพลาดบนช่องสัญญาณสำหรับทำให้ได้คุณภาพของวิดีโอที่ดีขึ้น
 โดยเฉพาะอย่างยิ่งการเข้ารหัสวิดีโอที่ทนต่อความผิดพลาดจึงเป็นหนึ่งในหัวข้อที่สำคัญในปัจจุบัน

ในวิทยานิพนธ์ฉบับนี้ ได้นำเสนอขั้นตอนการยืดหยุ่นความผิดพลาดสองวิธีการในการเข้ารหัสวิดีโอ
 ไร้สายด้วยการจัดเรียงมาโครบล็อกแบบยืดหยุ่นได้สำหรับงานประยุกต์ที่มีการรับส่งวิดีโอ ไร้สาย วิธีการที่หนึ่ง
 อาศัยผลกระทบของการกระจายของความผิดพลาดระหว่างเฟรมที่ถูกใช้เป็นตัวบ่งชี้ที่ใช้ในการสร้างแผนที่มาโคร
 บล็อกแบบยืดหยุ่นได้แบบภาพต่อภาพ เทคนิคหนึ่งที่ใช้ในการสร้างแผนที่มาโครบล็อกแบบยืดหยุ่นได้สามารถ
 ทำได้ด้วยการทำนายความผิดพลาดที่รุนแรงของช่องสัญญาณ นอกจากนี้การเลือกอัตราคืนสภาพภายในที่
 เหมาะสมยังช่วยลดผลกระทบที่เกิดจากการกระจายตัวของความผิดพลาด ดังนั้นจึงทำให้จำนวนมาโครบล็อกที่
 สำคัญและที่ไม่สามารถถอดรหัสได้มีจำนวนลดน้อยลง สำหรับวิธีการที่สอง ได้ทำการทดสอบ แนวความคิด
 ระหว่างชั้นคือ ชั้นการประยุกต์กับชั้นการควบคุมการเข้าถึงสื่อจากแนวความคิดพื้นฐานที่ว่ากลุ่มข้อมูลวิดีโอจะ
 ถูกจำแนกและถูกจัดลำดับความสำคัญที่ไม่เท่ากันที่ชั้นการควบคุมการเข้าถึงสื่อ จึงเป็นสาเหตุที่ทำให้เกิดกลุ่ม
 ข้อมูลที่ยกเลิกอย่างไม่จำเป็นเมื่อมีเข้าลำดับที่มีนัยสำคัญต่ำ โดยที่แนวความคิดที่นำเสนอเป็นการสร้างแผนที่กา
 จัดเรียงมาโครบล็อกแบบยืดหยุ่นในแต่ละลำดับภาพ จากการปรับตัวได้ของการจัดเรียงมาโครบล็อกแบบยืดหยุ่น
 และอัตราเข้าลำดับกระแสต้น ตัวเข้ารหัสแผนที่การจัดเรียงมาโครบล็อกแบบยืดหยุ่นในกลุ่มลักษณะที่อัตราการ
 เข้ามาถึงของกลุ่มข้อมูลซึ่งเป็นลำดับที่เต็มจะถูกทำให้ลดลง และอัตราการเข้ามาถึงของกลุ่มข้อมูลซึ่งเป็นลำดับที่
 ว่างจะถูกทำให้เพิ่มขึ้นบนพื้นฐานของการประมาณการเข้าลำดับกระแสต้นที่ชั้นการควบคุมการเข้าถึงสื่อ ดังนั้น
 จำนวนของกลุ่มข้อมูลที่ถูกลบที่ชั้นการควบคุมการเข้าถึงสื่อจึงลดลง

ผลจากการทดลองได้แสดงให้เห็นถึงวิธีการที่นำเสนอนั้นได้ปรับปรุงในเรื่องของอัตราส่วน PSNR
 จำนวนของมาโครบล็อกที่ไม่สามารถถอดรหัสได้ และ จำนวนของกลุ่มข้อมูลที่ถูกลบเมื่อเทียบกับวิธีการ
 เสนอก่อนหน้านี้ วิทยานิพนธ์นี้จะเป็นประโยชน์กับกลุ่มการสื่อสารวิดีโอในการนำมาใช้ในอนาคตของ
 เทคโนโลยีในการประยุกต์การส่งผ่านวิดีโอแบบไร้สาย


ภาควิชา..... วิศวกรรมไฟฟ้า.....

สาขาวิชา..... วิศวกรรมไฟฟ้า.....

ปีการศึกษา..... 2553.....

ลายมือชื่ออนิสิต..... 

ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก..... 

ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม..... 

507 18553 21: MAJOR ELECTRICAL ENGINEERING

KEYWORDS: H.264/ FMO/ ERROR PROPAGATION/ ERROR RESILIENT

TIEN HUU VU: ADAPTIVE FRAME BASED FLEXIBLE MACROBLOCK ORDERING FOR ERROR RESILIENT H.264 VIDEO CODING AND TRANSMISSION. THESIS ADVISOR: ASST. PROF. SUPAVADEE ARAMVITH, Ph.D., THESIS CO-ADVISOR: PROF. YOSHIKAZU MIYANAGA, Ph.D., 92 pp.

The transmission of video signal over wireless environments is becoming more common today as advances of telecommunication and multimedia convergence market make it possible. This enables the seamless mobile multimedia applications such as mobile video class and wireless video streaming. However, to achieve good video quality from such transmissions is still a challenging issue for practical video applications. This is due to the high error prone characteristic of wireless channels due to multipath effects and deep fades. Thus, research approaches to combat channel errors for better video quality especially error resilience video coding are one of the key research topics nowadays. In this dissertation, we propose two methods for error resilient in wireless video coding based on adaptive flexible macroblock ordering (FMO) for wireless video streaming applications. In the first method, effect of error propagation between frames is used as an indicator to generate FMO map frame-by-frame. A technique for generating FMO map is proposed to adapt with predicted error bursts of channel. In addition, a suitable intra refresh rate is selected to reduce the effect of error propagation. Therefore, the number of undecodable important macroblocks (MBs) is decreased. In the second method, we investigate the cross-layer approach between Application and Media Access Control (MAC) layers. In traditional approaches, video packets are classified and mapped into queues with different priorities at MAC layer. However, these approaches cause the unnecessary dropping packet at low priority queues. In the proposed approach, FMO map for each frame is generated by adaptive FMO and queuing overflow rate. Based on the estimated queuing overflow state information at the MAC layer, encoder maps MBs into slice groups in such a way that the arrival rate of packets to the full queue is reduced and arrival rate of packets to empty queue is increased. Hence, the number of dropped packet at Mac layer is minimized.

The results in experiments showed that the proposed methods gain an improvement in terms of peak signal-to-noise rate (PSNR), number of undecodable MBs and number of dropped packets as compared to the previously proposed methods. The contributions gained from this dissertation would benefit the video communication community in future adoption of technology in practical wireless video transmission applications.

Department: Electrical Engineering

Student's Signature

Field of Study: . Electrical Engineering

Advisor's Signature

Academic Year: 2010.....

Co-Advisor's Signature

ACKNOWLEDGMENTS

I would like to take this opportunity to express my deep gratitude to everyone who has made it possible for me to successfully complete this dissertation. First of all, I am deeply indebted to my advisor, Assistant Professor Supavadee Aramvith, Ph.D., for the great deal of effort she expended upon supervising me during my study in Chulalongkorn University.

My thanks also go to committee members, Professor Prasit Prapinmongkolkarn, Ph.D., Professor Yoshikazu Miyanaga, Ph.D., Associate Professor Kosin Chamnongthai, Ph.D., Assistant Professor Suree Pumrin, Ph.D., and Assistant Professor Chaiyachet Saivichit, Ph.D., for their contributing time, technical suggestions in the completion of this work.

I would like also to send my thanks to Assistant Professor Nguyen Huu Thanh, Ph.D and Dr. Ngo Quynh Thu, my Master's degree advisors, for enlightening me to research works, for directing me on my study plan at Chulalongkorn University, and for their concern of my studying in Thailand.

My sincere appreciation to friends in Digital Signal Processing Research Lab, Chulalongkorn University and in Laboratory of Information Communication Networks, Hokkaido University for their enthusiasm and all what they have done for me during my study.

A word of thanks is not enough for members in my family. They gave me motivation and patience while waiting for me to complete this dissertation. I would like to dedicate this work and express my heartfelt appreciation to them

Finally, I would like to send my gratefulness to the staffs of ISE and AUN/SEED-Net, for offering me the chance of pursue my Ph.D. degree. Without their financial assistance, care, and help, my study at Chulalongkorn University could not be processed.

This research has been supported in part by the Collaborative Research Project entitled Wireless Video Transmission, JICA Project for AUN/SEED-Net, Japan.

CONTENTS

	Page
Abstract (Thai)	iv
Abstract (English)	v
Acknowledgments	vi
Contents	vii
 CHAPTER	
I INTRODUCTION	1
1.1 Motivation and Problem Statement	1
1.2 Research Contributions	4
1.3 Thesis Organization	5
II BACKGROUND	7
2.1 Video Coding	7
2.1.1 Basic of video coding and standards	7
2.1.2 H.264 video coding standard	11
2.1.3 Error resilience tools in H.264	13
2.1.4 Flexible Macroblock Ordering	14
2.1.5 Network Abstract Layer	18
2.2 Wireless Channel	20
2.2.1 Wireless Channel Characteristics	20
2.2.2 Channel Model	21
2.2.3 Wireless Channel Simulator	24
2.3 Cross-Layer Optimization	26
2.3.1 Layers in Video Coding	26
2.3.2 Approaches in Cross-Layer	27

CHAPTER	Page
2.3.3 Previous works using cross-layer approach	29
III Adaptive FMO Based on Error Propagation for Error Resilient H.264 Video Coding	31
3.1 Introduction	31
3.2 Locating Error Burst Positions	34
3.3 Adaptive Explicit FMO Map Generation	39
3.3.1 The Estimation of MB Importance	39
3.3.2 Adaptive Explicit FMO Map Generation	42
3.4 Simulation Results and Discussions	44
3.4.1 Experimental Setup	44
3.4.2 Simulation Results	45
3.5 Summary	48
IV Adaptive FMO Using Cross-Layer Application-MAC Layer Scheme	55
4.1 Introduction	56
4.2 Overflow rate	59
4.3 Cross-layer MAC-Application-layer using Adaptive FMO and Queuing Overflow Rate	60
4.4 Experiment Results	62
4.4.1 Simulation setup	62
4.4.2 Result analysis	64
V Conclusions and Future Works	73
REFERENCES	75
 APPENDICES	
APPENDIX A	81
APPENDIX B	84
BIOGRAPHY	92

LIST OF TABLES

Table	Page
2.1 Comparison of H.264 to previous video codecs	11
2.2 Bitcount of MBs in frame 6 th frame, Akiyo sequence	17
2.3 Arranging MBs into slice groups	17
2.4 Explicit FMO map of 6 th frame, Akiyo sequence	18
2.5 NALU Packet Structure Field Description	19
2.6 Wireless channel and air interface parameters used in this study	25
3.1 Percentage difference (P_d) between estimated and actual locations of burst and guard sections	39
3.2 Comparison of Average PSNR (dB)	45
3.3 Comparison of Average PSNR (dB)	47
4.1 Access Category at the MAC layer	63
4.2 The number of video packets and packet size of each method	63
4.3 Average PSNR comparison of three methods at 64 kbps	68
4.4 Average PSNR comparison of three methods at 128 kbps	68
4.5 Average PSNR comparison of three methods at 384 kbps	68
4.6 Drop rate comparison at AC2 queue of three methods with 64 kbps . . .	69
4.7 Drop rate comparison at AC1 queue of three methods with 64 kbps . . .	69
4.8 Drop rate comparison at AC2 queue of three methods with 128 kbps . .	69
4.9 Drop rate comparison at AC1 queue of three methods with 128 kbps . .	69
4.10 Drop rate comparison at AC2 queue of three methods with 384 kbps . .	70
4.11 Drop rate comparison at AC1 queue of three methods with 384 kbps . .	70
4.12 Number of undecodable MBs of three methods with 64 kbps	71
4.13 Number of undecodable MBs of three methods with 128 kbps	71
4.14 Number of undecodable MBs of three methods with 384 kbps	71

LIST OF FIGURES

Figure	Page
2.1 Encoder structure of video codec.	8
2.2 Intra prediction samples for 4x4 blocks	12
2.3 Six types of FMO maps in H.264	15
2.4 NALU packet structure	19
2.5 Average duration of fade and level crossing rate	21
2.6 Packet sequence for an error channel	22
2.7 Three-state Markov Model.	23
2.8 Block diagram of wireless channel simulator	25
2.9 Error pattern with Doppler frequency of 1Hz	25
2.10 Error pattern with Doppler frequency of 40Hz	26
2.11 H.264 video transmission layers	27
3.1 Simple codec structure of H.264	32
3.2 Average PSNR of the proposed method for video sequences with dif- ference feedback values.	35
3.3 Effect of error propagation with two frames for feedback delay.	36
3.4 Locating burst and guard section for the current frame.	36
3.5 Error patterns of wireless channel simulator in slow fading at different bit rates.	37
3.6 Error patterns of wireless channel simulator in fast fading at different bit rates.	37
3.7 Comparison actual and estimated channel state for "Akiyo" sequence in slow fading, 32 kbps.	38
3.8 Comparison actual and estimated channel state for "Akiyo" sequence in fast fading, 32 kbps.	38
3.9 Error propagation from the past frame to the next frame.	40

Figure	Page
3.10 Flow chart of the proposed method.	49
3.11 EEP, Bitcount and ID values of MBs in frame 29th before sorting.	50
3.12 EEP, Bitcount and ID values of MBs in frame 29th after sorting.	50
3.13 Estimation of error burst and guard locations in wireless channel.	51
3.14 Explicit FMO map.	51
3.15 PSNR comparison of "Carphone" in slow fading.	52
3.16 PSNR comparison of "Carphone" in fast fading.	52
3.17 PSNR comparison of "Carphone" in slow fading.	53
3.18 PSNR comparison of "Carphone" in fast fading.	53
3.19 Visual comparison of frame 49 th of "Carphone" sequence between methods in slow and fast fading	54
4.1 Cross layer QoS architecture	57
4.2 Importance of MBs in frame 10 of "akiyo" sequence	60
4.3 Explicit FMO map of frame 10 of "akiyo" sequence	61
4.4 The importance of MBs after rearranging in frame 10 of "akiyo" sequence	61
4.5 Cross-layer architecture of the proposed method	62
4.6 Queue length of the method using Data Partition with "coastguard" se- quence at 64kbps	64
4.7 Queue length of the method using None-Adaptive FMO with "coast- guard" sequence at 64 kbps	65
4.8 Queue length of the proposed method using adaptive FMO with "coast- guard" sequence at 64 kbps	65
4.9 Subjective evaluation for the "Akiyo" sequence frame 50	72
4.10 Subjective evaluation for the "Coastguard" sequence frame 60	72
A.1 Comparison actual and estimated channel state for "Akiyo" sequence in slow fading, 64 kbps.	81
A.2 Comparison actual and estimated channel state for "Akiyo" sequence in fast fading, 64 kbps.	81

Figure	Page
A.3 Comparison actual and estimated channel state for "Akiyo" sequence in slow fading, 128 kbps.	82
A.4 Comparison actual and estimated channel state for "Akiyo" sequence in fast fading, 128 kbps.	82
A.5 Comparison actual and estimated channel state for "Akiyo" sequence in slow fading, 256 kbps.	83
A.6 Comparison actual and estimated channel state for "Akiyo" sequence in fast fading, 256 kbps.	83
B.1 PSNR comparison of "Akiyo" in slow fading at different bit rates.	84
B.2 PSNR comparison of "Akiyo" in fast fading at different bit rates.	84
B.3 PSNR comparison of "Akiyo" in slow fading at different bit rates.	85
B.4 PSNR comparison of "Akiyo" in slow fading at different bit rates.	85
B.5 PSNR comparison of "Foreman" in slow fading at different bit rates.	86
B.6 PSNR comparison of "Foreman" in slow fading at different bit rates.	86
B.7 PSNR comparison of "Foreman" in slow fading at different bit rates.	87
B.8 PSNR comparison of "Foreman" in slow fading at different bit rates.	87
B.9 PSNR comparison of "Claire" in slow fading at different bit rates.	88
B.10 PSNR comparison of "Claire" in slow fading at different bit rates.	88
B.11 PSNR comparison of "Claire" in slow fading at different bit rates.	89
B.12 PSNR comparison of "Claire" in slow fading at different bit rates.	89
B.13 PSNR comparison of "Carphone" in slow fading at different bit rates.	90
B.14 PSNR comparison of "Carphone" in slow fading at different bit rates.	90
B.15 PSNR comparison of "Carphone" in slow fading at different bit rates.	91
B.16 PSNR comparison of "Carphone" in slow fading at different bit rates.	91

CHAPTER I

INTRODUCTION

1.1 Motivation and Problem Statement

Video is one of the most popular types of media in several applications. However, with a large amount of data in raw video, the transmission of uncompressed video through networks such as the current internet is impossible. In addition, storage devices such as Compact Disk (CD), Digital Versatile Disk (DVD) can only store a few seconds of raw video at television-quality resolution and frame rate. Therefore, a good compression algorithm for video coding is as important as ever.

Nonetheless, compressed video often experiences errors when transmitting through communication channel, especially wireless channels. Hence, the decoded video quality is bound to suffer dramatically at high channel bit error ratios (BER). This quality degradation is exacerbated when no error resilient mechanism is employed to reduce the effects of error-prone environment. A single bit error that happens in a coded video stream could lead to disastrous quality deterioration for extended periods of time. Moreover, the temporal and spatial predictions used in most of the video coding standards today make the compressed video stream become more vulnerable to channel errors. This vulnerability is represented by the rapid propagation of errors in both time and space and the quick degradation of the reconstructed video quality.

To mitigate the effects of channel errors on decoded video quality, error-resilient mechanisms are added to advance video coding standards like H.264/AVC. These standards are equipped with a wide range of error-resilience tools such as flexible macroblock ordering (FMO), scalability, data partitioning, redundant slices, etc.

In this thesis, we investigate wireless video transmission quality and propose error-resilient video coding method utilizing FMO tools in H.264/AVC video coding standard to improve video quality. In video compression standards, macroblocks (MBs)

are processed and transmitted in raster-scan order, starting from the top-left corner of the image to the bottom right. In this case, if an MB is lost, this usually results in the loss of neighboring area in a single frame. In order to provide a more flexible transmission order of MBs in a frame in H.264/AVC, FMO allows MBs to be mapped into slice groups. Each slice group can be divided in several slices and a slice can also be decoded independently. An identification number for each MB is given by a macroblock allocation map (MBAmap) to specify which slice groups MB belong to. Therefore, MBs may be transmitted out of raster-scan order in a flexible and efficient way. For example, by using FMO, spatially collocated image areas can be interleaved in different slices. In case of a MB is lost, the neighboring MBs are still available in other error-free slices. This is useful for decoder in concealment when using the availability of correctly received neighbor MBs to conceal a lost MB.

To generate an FMO, initially, an importance indicator for MBs is selected. There are some researches using bitcount [1], MB PSNR [2] as indicator. After that, based on importance, MBs are mapped into slice groups in a specified way to create FMO map. In the method using FMO tool, two-pass scheme is used. In the first pass, all frames are encoded to estimate the importance value for MBs. In the second pass, the frames are encoded again with explicit FMO map generated in the first pass. Thus, this method is applied for video streaming applications such as video-on-demand, video news archive. In these applications, video sequences are encoded in advance in the first pass to extract some necessary parameters of video sequence including motion vector, bitcount, PSNR, etc. These parameters are stored in database. When an end-user requests a video sequence, encoder carries out encoding the requested raw video sequence with appropriate data from database.

In this research, distortion of error propagation is used because this is one of causes making the degradation of video quality. In video coding, to exploit the temporal redundancy between successive frames of video sequence, motion compensated prediction is used. Because of dependence between frames in this method, the error is spread from one frame to the next frame if that frame is error while transmitting over communication channel. Thus, measuring the distortion of error propagation from the

past frame to the current frame is necessary in evaluating the effect of this error type.

To stop error propagation, intra coding mode is used. In video coding, each MB may be encoded in one of two coding modes: inter-mode and intra-mode. In inter-mode, the MB is first "predicted" from the previously decoded frame via motion compensation. Then the prediction error, or residue, is transform-coded. In intra-mode, the original MB data are transform-coded directly without prediction. Although operation in inter-mode generally achieves higher compression efficiency, it is more sensitive to channel errors as the dependence between frames. If an MB is error while transmitting over error-prone channel, the MBs referring to the error MB in the next frame are also error. Thus, to stop error propagation, intra-mode should be applied to MBs referring to error MBs.

However, there are some issues when using intra coding mode. Firstly, intra-mode consumes more bit than inter-mode. Therefore, the number of intra MBs in the current frame may affect to the quality of the next frames in a same group of pictures (GOP). In H.264/AVC, before encoding a GOP, the total number of bits allocated for this GOP is estimated. If a frame with too many intra coded MBs may drain bits in total, the bits left to the next frames will be exhausted. As a result, the quality of video in the next frames may not be guaranteed. Hence, a suitable number of intra MBs in a frame is necessary to reduce the effect of error propagation while not affect to compression efficiency. Secondly, beside intra refresh rate, the distribution of intra MBs in slice groups is also considered. If intra MBs concentrate densely in a slice group, the target bit allocated to slice groups will not be equal, as the result, the quality of slice groups are not uniform. Thus, intra MBs should be dispersed in all slice groups by using FMO tool to avoid losing important MBs in the worse case of channel.

Beside the error-resilient methods, cross-layer schemes also are investigated in this study. The characteristics of real-time multimedia applications are high bandwidth consuming and very stringent in delay constraint. Unlike file transfers, real-time multimedia applications do not require a complete insulation from packet losses, but rather require the Application (APP) layer to cooperate with the lower layers to select the optimal wireless transmission strategy that maximizes the multimedia performance. Thus,

in order to address multimedia support over wireless network, the resource management, adaptation, and protection strategies available in the lower layers such as Physical layer, Medium Access Control (MAC) layer, and Network/Transport layer are optimized with considering the specific characteristics of the multimedia applications.

In the previous researches using cross-layer approach [3], [4], packets are classified at Application layer and mapped into queues with different priorities at MAC layer. In these methods, although important packets are protected from dropping, high priority queues are not used effectively. With the higher priority, these queues are usually empty while the lower priority queues have to drop packets because of fullness. The unbalance in using queues causes to the unnecessary dropping packet at the lower priority queues.

In this research, by using APP-centric approach in the proposed method, the APP layer optimizes the parameters of MAC layer based on the requirements from MAC layer. In particular, the overflow state of queues at MAC layer is monitored by video encoder. Depend on the state of queues, FMO map is changed to adapt with the fullness state of queues. Thus, the dropping packet rate at queues is decreased.

1.2 Research Contributions

The main contribution in this research is to propose an error resilience framework considering channel prediction to generate an adaptive explicit FMO map and adopting suitable intra refresh rate for H.264 video transmission. We propose a method using three-state Markov model to estimate channel state. Initially, based on feedback information from decoded frames, average burst length (ABL) and average guard length (AGL) of channel are computed. After that, error bursts in channel at transmission period of the current frame are predicted. In addition, a technique is introduced to evaluate the importance of MBs based on error propagation between frames. Finally, an explicit FMO map of the current frame is generated as the rule: MBs with low importance are arranged in slice groups which are transmitted in error burst sections and MBs with high importance are transmitted in error-free sections. To reduce the number of lost important MBs, MBs are scattered into some slice groups. Furthermore, with the esti-

mated information channel state, a suitable intra refresh rate is computed to stop error propagation.

In the second proposed method, joint Application-MAC cross-layer mechanism is used to generate explicit FMO map for H.264 video transmission over an IEEE 802.11e wireless network. In cross-layer approach, FMO map of each frame is generated at encoder by using information from queues of MAC layer. In particular, a frame of video is divided into slice groups including high important and low important slice groups. In case of light network traffic load, the low important slice groups are encoded first and mapped into low priority queue. After that, the high important slice groups are encoded and mapped into high priority queue. However, in case of heavy network traffic load, the congestion happens and results in the overflow state at queues. Because of different priority in queues, high priority queues usually are in light load state while low priority queues are in over load state. To avoid unnecessary packet dropping at low priority queues, FMO map is changed in the way that high important slice groups are encoded and mapped to high priority queue first. Accordingly, the arrival rate of packets to the low priority queue is reduced and thus dropping packet at this queue is decreased.

1.3 Thesis Organization

This thesis is organized into five chapters including this chapter. The following paragraphs provide brief descriptions of the remaining chapter of this thesis.

Chapter 2 we discuss some basic video compression concepts, some background about the H.264 video codec, some error resilient tools in H.264/AVC including FMO especially. Some characteristics of wireless channel are also mentioned in this chapter. Finally, a short summary of 802.11 wireless LAN standard is introduced.

Chapter 3 presents the first method using adaptive explicit FMO map to reduce effect of error propagation in H.264 video coding. Results of the proposed method are compared to some previous methods and are shown in this chapter.

Chapter 4 presents the second method using cross-layer mechanism to generate FMO map with target at reducing packet dropping at queues of MAC layer. The

proposed method is applied to video transmission over 802.11e WLAN. Results and analysis are shown of the method is shown in this chapter.

Chapter 5 includes conclusions and future works of the research.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

CHAPTER II

BACKGROUND

In this chapter, some video compression concepts and the latest H.264/AVC video coding standard are introduced. Related to the main topic of this research, error resilient tools and flexible macroblock ordering tool especially are discussed in the next section. The characteristic of the wireless channel is presented to provide some basic understanding of the nature of error in fast and slow fading channel environment. Finally, cross layer optimization is introduced.

2.1 Video Coding

2.1.1 Basic of video coding and standards

Compression is the process of compacting data into a smaller number of bits. Video compression (video coding) is the process of compacting or condensing a digital video sequence into a smaller number of bits. 'Raw' or uncompressed digital video typically requires a large bitrate and compression is necessary for practical storage and transmission of digital video.

Compression involves a complementary pair of systems, a compressor (encoder) and a decompressor (decoder). The encoder converts the source data into a compressed form prior to transmission or storage and the decoder converts the compressed form back into a representation of the original video data. The encoder/decoder pair is often described as a CODEC.

Video codec consists of some basic functional units such as prediction, motion compensation, transformation, quantization, and entropy coding. A functional block diagram of an encoder is shown in Fig. 2.1. The encoder consists of two dataflow paths. The forward path deals with encoding a frame or field. The reconstruction path deals with reconstructing a frame to form references for future predictions.

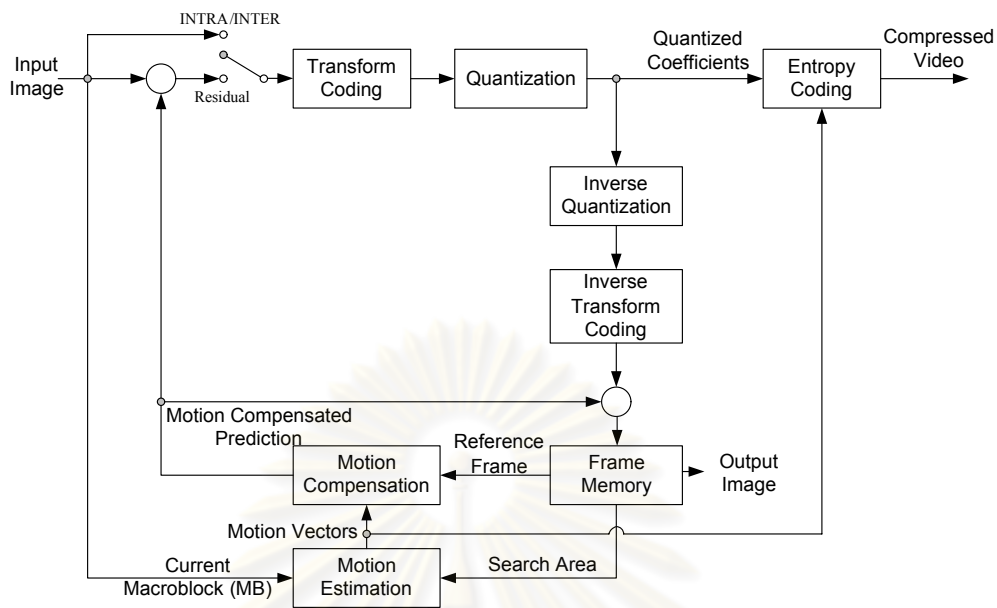


Figure 2.1: Encoder structure of video codec.

Each picture in a progressive or interlaced video signal is represented by a frame of pixels and in an interlaced video signal as fields. A frame consists of two fields, a Top field and a Bottom field. A Top field captures the odd lines of a frame and a Bottom field captures the even lines of a frame. Each frame/field is encoded by a video encoder and finally transmitted as a bitstream to a receiver. Each frame is an encoded video bitstream is identified by a frame number and similarly each coded field has an associated picture order count.

An input frame is processed in units of macroblocks (MB). An MB is intra or inter encoded. Typically all MBs in the first frame of a video sequence are intra coded. In intra mode, the prediction signal for the current MB is formed from spatially neighboring samples that have been previously encoded in the current slice. The prediction signal is formed based on the chosen prediction mode which yields the least prediction error or residual. Prediction residual is computed by subtracting the prediction signal from the original samples. MBs of subsequent frames of a video sequence are inter coded or intra coded which ever yields least prediction error between the original signal and the predicted signal. Often, MBs of remaining frames are inter coded. In inter coding a prediction signal is formed from a reference frame by motion compensation and motion

estimation. Motion estimation will determine the motion vector which minimizes the prediction error.

Motion compensation will apply the estimated motion vector displacement to the reference picture to form the prediction signal, which is subtracted from the original samples to yield a prediction error/residual. Motion vectors are encoded and transmitted as side information. But encoding a motion vector of each MB partition or subpartition will cost a significant number of bits. In order to save bits, motion vector prediction will predict the motion vector of the current partition or sub partition based on the motion vector information of neighboring partitions or sub partitions. Motion vector difference between the current vector and the predicted vector is encoded and transmitted.

The residual signal is first transformed using DCT transform, and then scaled. The scaled transformed coefficients are quantized and then entropy coded. Sophisticated entropy coding schemes such as Context Adaptive Binary Arithmetic Coding (CABAC) and Context Adaptive Variable Length Coding (CAVLC) are employed. The coded data is transmitted as a bitstream.

The reconstruction path of the encoder is a mini decoder which deals with inverse quantization and inverse transformation of quantized DCT coefficients. Inverse transformed DCT coefficients will form the reconstructed residual signal which is then added to the prediction signal already computed in the forward path of the encoder to form the reconstructed MB. The reconstructed MB is filtered using a deblocking filter to reduce blocking distortion, by smoothing the block edges thereby improving the appearance of a reconstructed frame. These frames are used as references for further prediction.

Decoder is very similar to the reconstruction path of Encoder. The Entropy decoder decodes the received bitstream to generate quantized DCT coefficients and forms a prediction signal based on the received prediction modes and motion vector information. The rest of the procedure is the same as the reconstruction path of the encoder.

The most important developments in video coding standards have been due to two international standards bodies: the International Telecommunications Union (ITU) and the International Standards Organization (ISO). The ITU has concentrated on standards to support real-time, two-way video communications. The group responsible for

developing these standards is known as VCEG (Video Coding Experts Group) and has issued:

- H.261 (1990): Video telephony over constant bit-rate channels, primarily aimed at ISDN channels of *64kbps*.
- H.263 (1995): Video telephony over circuit and packet switched networks, supporting a range of channels from low bit rates (20-30kbps) to high bit rates (several Mbps).
- H.263+ (1998), H.263++ (2001): Extensions to H.263 to support a wider range of transmission scenarios and improved compression performance.
- H.26L: The project of the VCEG and the ISO/IEC Motion Picture Experts Group (MPEG). The main goals of H.28L are a simple and straight forward video coding design to achieve enhanced compression performance for “conversation” (video telephony) and “non-conversational” (storage, broadcast, or streaming) applications.

In parallel with the ITU’s activities, the ISO has issued standards to support storage and distribution applications. The two relevant groups are JPEG (Joint Photographic Experts Group) and MPEG (Moving Picture Experts Group) and they have been responsible for:

- JPEG(1992): Compression of still images for storage purposes.
- MPEG-1 (1993): Compression of video and audio for storage and real-time playback on CD-ROM (at a bit rate of 1.4Mbps).
- MPEG-2 (1995): Compression and transmission of video and audio programmes for storage and broadcast applications.
- MPEG-4 (1998): Video and audio compression and transport for multimedia terminals.

Table 2.1: Comparison of H.264 to previous video codecs

Algorithm Characteristic	MPEG2/H.263	H.264
Intra Prediction	None	Multi-direction, Multi-pattern
Coded Image Types	I, B, P	I, B, P, SP
Transform	8x8 DCT	4x4 DCT
Motion Estimation Blocks	16x16	16x16, 8x8, 4x4
Frame Distance for Prediction	+/- 1	Unlimited forward/backward
Fractional Motion Estimation	$\frac{1}{2}$ Pixel	$\frac{1}{4}$ Pixel
Supported error resilient tools	Just one method using additional pictures	Providing many tools including: Flexible slice structured coding, Flexible MB ordering, Arbitrary slice ordering, Redundant pictures, Data partitioning, SP/SI synchronization/switching pictures

- JPEG-2000 (2000): Compression of still images (featuring better compression performance than the original JPEG standard).

2.1.2 H.264 video coding standard

H.264/AVC [5] is the latest video compression standard jointly developed by MPEG and VCEG. H.264 is otherwise referred to MPEG-4 part 10. This new standard is far superior to earlier video coding standards such as H.263 and MPEG-2, in terms of compression efficiency. The efficiency is achieved by improving several function units or blocks of the standards such as motion compression, inter prediction, intra prediction, transformation, quantization and more importantly improved context adaptive entropy coding techniques. Some advantages of H.264 compared to previous video codec such as MPEG2, H.263 etc. are shown in Table 2.1.

Intraframe prediction:

H.264 exploits spatial redundancies better than MPEG-2 by allowing intraframe prediction. An MB is coded as an intra MB when temporal prediction is impossible (for the first frame of video) or inefficient (at scene changes). The prediction for the

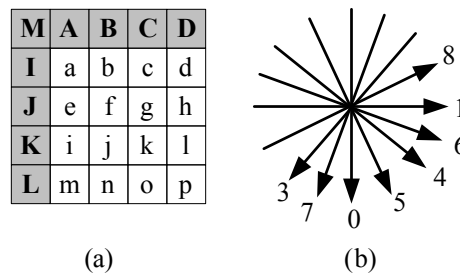


Figure 2.2: Intra prediction samples for 4x4 blocks

intra MB is determined using the neighboring pixels in the same frame because they are likely similar. An MB can be coded as one 16x16, four 8x8, or sixteen 4x4 blocks. The predictions for these blocks are determined using the weighted average of the pixels to the left of and above the current block.

Figure 2.2(a) shows the neighboring pixels used to predict a 4x4 block. The current block's pixels (lowercase alphabet in the figure) are predicted using a weighted average of the neighboring pixels (uppercase alphabet in the figure). The arrows in Figure 2.2(b) indicate the direction of prediction in each mode. For mode 3-8, the predicted samples are formed from a weighted average of the prediction samples A-M. For example, if mode 4 is selected, the top right sample of MB (labeled "d" in Figure 2.2(a)) is predicted by $\text{round}(B/4 + C/2 + D/4)$.

There are nine prediction modes for 4x4 and 8x8 block sizes and four prediction modes for the 16x16 block. These intra prediction modes perform well when a picture contains directional structures. The difference between the current and predicted blocks is then encoded. With the intraframe prediction, the intra MBs can be encoded more efficiently compared to MPEG-2, which doesn't support intraframe prediction.

Interframe prediction:

The interframe prediction is the traditional motion compensated prediction supported by earlier MPEG video coding standards including MPEG-2. The H.264 extends this several ways:

- Variable block sizes for motion compensation.
- Multiframe references for prediction.

- Generalized B frame prediction.
- Use of B frames as references.
- Wighted prediction.
- Fractional pixel accuracy for motion vectors.

These extensions improve the coding performance and increase the complexity substantially. The motion compensation of MBs in H.264 uses variable block sizes and motion vectors with quarter-pixel resolution. An MB can be coded as one 16x16, two 16x8, two 8x16, or for 8x8 blocks. Each 8x8 block can be in turn be coded as one 8x8, two 4x8, two 8x4, and four 4x4 sub blocks. An I frame can use as many as 16 difference reference frames, and the actual number of reference frames is only constrained by the buffer size in a particular profile and level.

2.1.3 Error resilience tools in H.264

When the compressed video bitstream is transmitted over a communication channel, it is subjected to channel error. Generally, forward error correction (FEC) code is used for protecting data. However, the FEC is only effective for random errors, but inadequate in the case of burst errors. Thus, H.264/AVC provides some error resiliency schemes in Video Coding Layer (VCL). These schemes includes:

Flexible Macroblock Ordering (FMO): In this tool, picture can be partitioned into regions (slice). Each slice can be independently decoded. The purpose of this tool is stopping the propagation of errors between slices.

Arbitrary Slice Ordering (ASO): Since each slice is independently decodable, slices can be sent and received out of order. This can improve end-to-end delay time on certain networks.

Data Partitioning: In H.264/AVC data partitioning mode, each slice can be segmented into header and motion information, intra information, and inter texture information by simply distributing the syntax elements to individual data units. These information are mapped into three partitions A, B and C. Partition A contains header

and motion information. Partition B contains intra information. The inter information is mapped into partition C. If the partition A is lost, it is likely to be difficult or impossible to reconstruct the slice. Partition B and C can be made to be independently decodable and so a decoder may decode A and B only, or A and C only, lending flexibility in an error-prone environment.

Redundant Pictures: A picture marked as “redundant” contains a redundant representation of part or all of a coded picture. In normal operation, the decoder reconstructs the frame from “primary” (nonredundant) pictures and discards any redundant pictures. However, if a primary coded picture is damaged, the decoder may replace the damaged area with decoded data from a redundant picture if available.

Switching Pictures: A new feature in H.264/AVC consisting of picture types that allow exact synchronization of the decoding process of some decoders with an ongoing video stream produced by other decoders without penalizing all decoder with the loss of efficiency resulting from sending an Intra-coded picture. This can be enable switching a decoder between representation of the video content that used different data rates, recovery from data losses or errors as well as enabling fast-forward and fast-reverse playback functionality.

2.1.4 Flexible Macroblock Ordering

FMO [6] is one of the new error-resilience tools available of the H.264/AVC standard designed to improve transmission of video streams over error-prone networks. In H.264/AVC, an image can be divided into regions called slice groups. Each slice group can be further divided in several slices which containing a sequence of MBs. These MBs are processed in a scan order (left to right and top to bottom) and a slice can be decoded independently to the other slices. By using FMO, each MB can be assigned freely to a slice group using an MBAMap. The MBAMap consists of an identification number for each MB of the image that specifies which slice group the MB belongs to. The number of slice groups is limited to 8 for each image to prevent complex allocation schemes. If FMO is not used, the images will be composed of a single slice with the MBs in a scan order. The use of FMO is totally compatible with any type of inter-frame

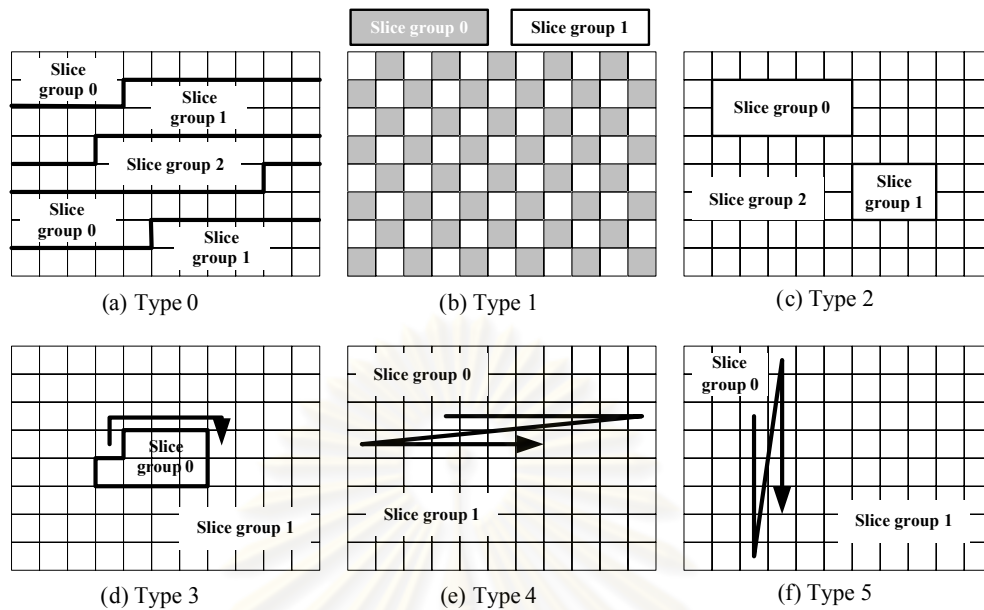


Figure 2.3: Six types of FMO maps in H.264

prediction.

With this technique, errors can be corrected easily by exploiting the spatial redundancy of the images. It's a good idea to choose the slice groups in a way that no MB and its neighbors belong to the same group. Therefore, if a slice is lost during transmission, it's easy to reconstruct the lost blocks with the information of the neighboring blocks. However, the disadvantage of using FMO is the slight reduction in coding efficiency, because motion compensation will be limited to MBs with the same slice groups. Another drawback is the number of overhead bits incurred by the slice header for each slice group will be increased. This can affect to low bit-rate applications because the texture bit for MBs will be decreased and thus quality of video can be reduced.

H.264/AVC provides 7 types of FMO, labeled Type 0 to Type 6 as shown in Fig. 2.3.

- FMO type 0 uses runlengths which are repeated to fill the frame. Therefore those runlengths have to be known to rebuild the MBMap on the decoder.
- FMO type 1, also known as scattered slices, uses a function, which is known to both the encoder and decoder, to spread the MBs. The more slice group used, the

more each MB will be surrounded by MBs from different slice groups.

- FMO type 2 is used to mark rectangular areas, so called regions of interest (ROI), inside a frame. MBAmaps can be stored using the top left and bottom right coordinates of those rectangles.
- FMO types 3 to 5 are dynamic ones and let the slice groups grow and shrink over the different pictures in a cyclic way. Only the grow rate, the direction and the position in the cycle have to be known.
- FMO type 6 is called the explicit FMO. This type allows the full flexibility of assigning MBs to any slice, as long as the mapping is specified in the MBAmap. Extensive studies have been done by different researches to investigate the use of explicit FMO as an error resilient tool for H.264. The general procedure on how to use explicit FMO to design a specific MB-to-slice group mapping is as follows:
 - Parameter Specification. Find a parameter to quantify the importance of a MB.
 - MB Classification. Classify the MBs to slice groups using the chosen parameter.
 - MBAmap design. The result of the classification process determines the MB-to-slice group map.

To design the slice group map, previous approaches use an indicator to define the important of MB. Those previously proposed indicators are bit count [1], spatial-temporal indicator [7], distortion-from-error propagation [8], MB PSNR parameter [2], and MB importance factor [9].

In [1], bit count indicator was used to measure the importance of MB. The basic idea of this approach is that the MB using the higher number of bits is the more important MB due to the properties of motion-compensated prediction. Therefore, we try to interleave consecutive the more-bit-count MB to be in different slice.

Table 2.2: Bitcount of MBs in frame 6th frame, Akiyo sequence

0	0	0	0	0	4	0	0	0	0	0
0	0	0	0	397	582	318	0	0	0	0
0	0	0	0	888	1512	432	0	0	0	0
0	0	0	5	2830	2334	1274	235	0	0	0
0	0	0	0	504	2064	770	80	0	0	0
0	0	69	332	695	1470	485	240	332	5	0
0	0	0	0	125	575	414	0	173	108	0
0	138	0	290	0	294	318	0	0	522	0
0	0	0	0	0	5	193	0	0	0	0

Table 2.3: Arranging MBs into slice groups

Slice group	MB bitcount values
0	2830 695 397 235 5 0 0 0 0 0 0 0
1	2334 582 332 193 5 0 0 0 0 0 0 0
2	2064 575 332 173 5 0 0 0 0 0 0 0
3	1512 522 318 138 4 0 0 0 0 0 0 0
4	1470 504 318 125 0 0 0 0 0 0 0 0
5	1274 485 294 108 0 0 0 0 0 0 0 0
6	888 432 290 80 0 0 0 0 0 0 0 0
7	770 414 240 69 0 0 0 0 0 0 0 0

Table 2.2 shows an example of bitcount for MBs in frame 6th of Akiyo sequence. We can see that MBs having high bitcount concentrate in the center of frame. If they are not dispersed into some slice groups, many important MBs are affected when an MB is error. Thus, [1] proposes a method to dispersed important MBs to 8 slice groups by sorting MBs in descending order of bitcount and mapped to 8 slices consecutively as shown in Table 2.3. According to this arrangement, neighborhood important MBs in Table 2.2 are separated into 8 slice groups. Therefore, if an MB is error, the other MBs are not affected.

Finally, we have an explicit FMO map for 6th frame as shown in Table 2.4.

The results from [1] showed that, with the proposed method, the number of undecodable MBs is decreased by up to 70% . However, bitcount indicator may not be accurate in evaluating the importance of MB because bitcount includes the information

Table 2.4: Explicit FMO map of 6th frame, Akiyo sequence

0	1	2	3	4	7	5	6	7	0	1
2	3	4	5	0	1	4	6	7	0	1
2	3	4	5	6	3	6	6	7	0	1
2	3	4	0	0	1	5	0	5	6	7
0	1	2	3	4	2	7	6	4	5	6
7	0	7	1	0	4	5	7	2	5	1
2	3	4	5	4	2	7	6	2	5	7
0	3	1	6	2	5	3	3	4	3	5
6	7	0	1	2	1	1	3	4	5	6

of the number of bits of motion vector difference (MVD) as well as the number of bits of residual. MVD is the difference between the predicted and actual motion vector, and the predicted motion vector can be changed following which slice group the MB belongs to. Hence, bitcount information of an MB is vary and may not truly represent the importance of that MB.

In [7], we can overcome the limitations of bit count method by using distortion and considering the importance of MB within one slice group. The method is based on the idea that if we have an initial MB-to-slice group mapping, derived from distortion measure for example, we generate another slice group map that will have lower distortion from concealment error. As an extension, in [8], in addition to the error propagation within a slice group, the error propagation between frames is considered to estimate the importance of MBs. In [2]- [9], PSNR and error concealment distortion are used in a heuristic method to find out important MBs. After that, an efficient FMO mapping technique is proposed accompany with an UEP technique to protect important MBs.

2.1.5 Network Abstract Layer

Unlike earlier video coding standards where compressed video is just transmitted as a bit stream, H.264 uses an interface layer to interact with lower layers such as Real-time Protocol (RTP). This Network Abstract Layer (NAL) is part of the standard and it interfaces the VCL from other lower layer. NAL abstracts VCL data from the network related parameters. Hence the VCL data can be transported in a variety of networks

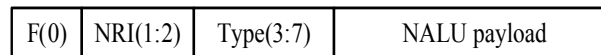


Figure 2.4: NALU packet structure

Table 2.5: NALU Packet Structure Field Description

Field	Description	Values
F	Forbidden bit	0: no errors 1: syntax error
NRI	NALU Reference Indicator	00: not used for reference 01: low priority 10: high priority 11: highest priority
Type	NALU defined types	28 is used in this research
NALU Payload	Variable length NALU payload	

such as IP networks, circuit switched networks, etc. This network friendliness nature of H.264 is exploited here to transport the NAL packets through RTP.

The NAL is used to transmit both VCL data and Non VCL data. VCL data represents the compressed video bit stream, such as encoded motion vectors, encoded quantized DCT coefficients etc. Non VCL data are Sequence Parameter Sets (SPS) and Picture Parameter Sets (PPS). These parameter sets carry very critical information. Without knowing these parameter sets, the decoder does not know how to decode a bit stream. The SPS contains all the information related to a video sequence defined between any two Intra Decoder Refresh (IDR) frames. The PPS contains all the information common to all slices in a single picture.

Whether it is VCL data or Non VCL data, it is encapsulated into the NAL Unit (NALU). The packet structure of a NALU is shown below in the Fig. 2.4 and a description of the fields is listed in the Table 2.5. For VLC data, the NALU payload carries one slice of information. For Non VCL data, the PPS and the SPS are transmitted in separate NALUs. For a detailed description of NALU packet structure, please refer to [10]. In the scope of thesis, we will use field `Nal_Ref_Idc` (NRI) to setup priority for video packets.

2.2 Wireless Channel

2.2.1 Wireless Channel Characteristics

Wireless channel characteristics depend on the model of propagation used by wireless communication system [11], [12]. There are two kinds of propagation model: large-scale and small-scale or fading model. Large-scale propagation models are used to characterize signal strength over large transmitter-receiver separation distances (several hundreds or thousands of meters). On the other hand, small-scale propagation models are used to characterize the rapid fluctuations of the received signal strength over very short travel distances (a few wavelengths) or short time durations (on the order of seconds). In this work, we consider to only small-scale propagation model. Doppler spread and coherence time are parameters which describe the time varying nature of the channel in a small-scale region. Doppler spread is a measure of the spectral broadening caused by the time rate of change of the mobile radio channel and is defined as the range of frequencies over which the received Doppler spectrum is non-zero. Coherence time is the time domain dual of Doppler spread and is used to characterize the time varying nature of the frequency dispersiveness of the channel in the time domain. The Doppler spread and coherence time are inversely proportional to one another.

Depending on how rapidly the transmitted baseband signal changes as compared to the rate of change the channel, a channel may be classified either as a fast fading or slow fading. In a fast fading channel, the coherence time of the channel is smaller than the symbol period of the transmitted signal. This causes frequency dispersion due to Doppler spreading, which leads to signal distortion. In slow fading channel, the Doppler spread of the channel is much less than the bandwidth of the baseband signal. It should be clear that the velocity of the mobile and the baseband signaling determines whether a signal undergoes fast fading or slow fading.

In order to characterize a wireless channel, and determine its quality, we use typical parameters which describe the quality of the transmitted signal along the transmission channel, such as the Level Crossing Rate (LCR) and Average Fading Duration (AFD) (Fig. 2.5). LCR is defined as the expected rate at which the Rayleigh fading

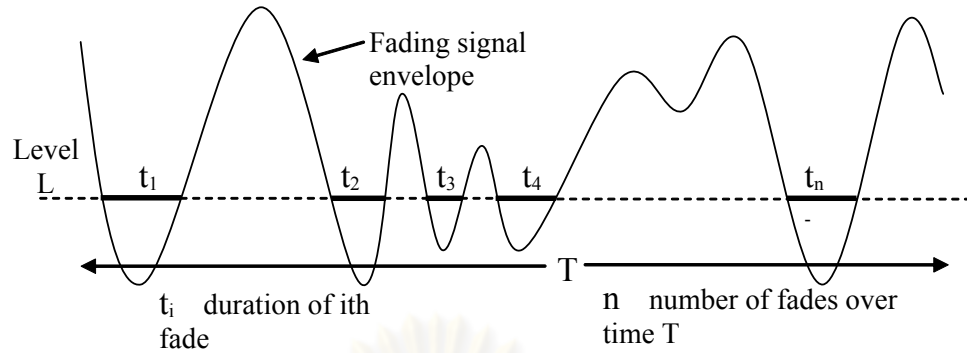


Figure 2.5: Average duration of fade and level crossing rate

envelope crosses a specified level in a positive-going direction. The number of level crossings per second is given in Eq. (2.1) [11]:

$$LCR = \sqrt{2\Pi} f_m \rho e^{-\rho^2} \quad (2.1)$$

f_m , maximum Doppler shift, is equal to the velocity divided by the wavelength $f_m = \frac{v}{\lambda}$. $\rho = \frac{R}{R_{rms}}$ is the value of the level R normalized to the root mean square values of fading envelope amplitude. AFD presents the average time period for which the received signal envelope is lower than a specific level. AFD duration is computed by:

$$ADF = \frac{e^{\rho^2} - 1}{\sqrt{2\Pi} f_m \rho} \quad (2.2)$$

As shown in Eq. (2.1) and (2.2), ADF is inversely proportional to f_m and LCR is directly proportional to f_m . So we can see that in fast fading case, the error bursts are shorter but occur more frequently and vice-versa in slow fading case.

2.2.2 Channel Model

There are many different methods used to model the wireless channel. Two-state Markov model is one of the popular models and is used in many researches for video transmission. In [13] [14], channel is modeled by two-state Markov model whose transition probabilities are functions of the channel characteristic. In [15], throughput of wireless channel is estimated by two-state Markov model at packet level. In [16], characteristics of wireless channel derived from two-state Markov model is used to optimize intra re-

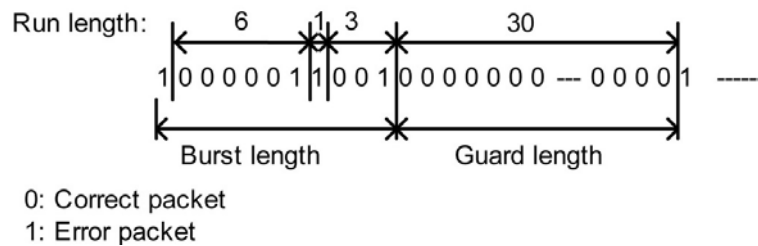


Figure 2.6: Packet sequence for an error channel

fresh rate as well as determine an optimal channel code rate to get the maximum MSE at decoder. However, two-state Markov has limitation in estimation of error location of channel. In particular, model used in [13]- [16] can only predict whether the next state of channel is good or bad. Two-state model cannot estimate the average length of guard and burst section. Thus, locations of guard and burst cannot be estimated. To deal with this limitation, a higher order of Markov models, i.e., more than two states are used. As an extension of two-state model, finite-state Markov model is analyzed in [17], [18]. Authors indicate that the higher state of Markov model is used, the more accurate in capturing the error burst nature of channel. However, the complexity level of model is also directly proportional to the number of states in Markov model. With the limitation of two-state model and the cost in computing of higher state models, three-state Markov model is selected to estimate channel state.

In this section, three-state Markov model in [19], [20] is introduced. However, instead of using model in bit level, we apply three-state Markov model in packet level to estimate the position of error bursts in wireless channel. Figure 2.6 shows an example of a packet sequence in an error channel. Similarly to [19], [20], we define the following definitions at packet level. A guard section is defined as a duration in which all packets are error-free. A burst section is defined as duration sandwiched between guard sections. From now on, the section means guard or burst section. Minimum guard length is the minimum number of error-free packets a guard section should have. In this system, the minimum guard length is chosen to be 30 packets for computing the transition probability. Thus, each guard section is longer than or equals to 30 consecutive error-free packets. The run length is defined as the length from an error packet to

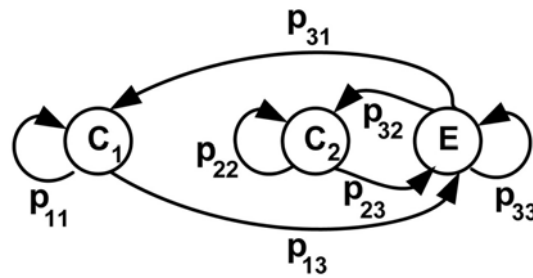


Figure 2.7: Three-state Markov Model.

the next error packet excluding the first error packet. The first return probability $P(i)$ is defined as the occurrence probability of each run length i .

Figure 2.7 shows the transition probabilities of a three-state Markov model where C_1 and C_2 show error-free states which are state 1 and state 2, and E shows the error state which is state 3. C_1 shows error-free state in a guard section while C_2 and E show error-free and error state in a burst section. p_{nm} is the probability of transition from state n to state m . p_{nn}^{i-2} is the probability of the case in which there are $(i - 2)$ consecutive transitions from state n to state n . The first return probabilities are computed by

$$\begin{aligned}
 P(1) &= p(E/E) = p_{33} \\
 P(2) &= p(C_1, E/E) + p(C_2, E/E) \\
 &= p_{31}p_{13} + p_{32}p_{23} \\
 &\dots \\
 P(i) &= p(C_1, \dots, C_1, E/E) + p(C_2, \dots, C_2, E/E) \\
 &= p_{31}p_{11}^{i-2}p_{13} + p_{32}p_{22}^{i-2}p_{23}
 \end{aligned} \tag{2.3}$$

where $p(\alpha/\beta)$ means that β is the first state and then the sequence of α occurs. The

transition probabilities are computed by

$$\begin{aligned}
p_{33} &= P(1) \\
p_{32} &= \sum_{i=2}^{L_{min}} P(i) \\
p_{23} &= \frac{p_{32}}{\sum_{i=2}^{L_{min}} (i-1)P(i)} \\
p_{31} &= 1 - p_{33} - p_{32} \\
p_{13} &= \frac{1}{\sum_{i=L_{min}+1}^{\infty} \frac{(i-1)P(i)}{p_{31}} - L_{min}} \\
p_{22} &= 1 - p_{23} \\
p_{11} &= 1 - p_{13}
\end{aligned} \tag{2.4}$$

where L_{min} is the minimum guard length. According to [19], the average guard length, L_G , and average burst length, L_B , are computed as

$$\begin{aligned}
L_G &= L_{min} + \frac{1}{\frac{p_{31}}{p_{13}} + \frac{p_{32}}{p_{13}} + 1} \\
L_B &= \frac{p_{13}}{p_{31}p_{11}^{L_{min}-1}} - \left(1 + \frac{1}{p_{13}}\right)
\end{aligned} \tag{2.5}$$

2.2.3 Wireless Channel Simulator

In this thesis, Rayleigh fading channel is simulated by using the technique described by Jakes [21]. The autocorrelation function of the simulated two-ray fading waveforms approximates the Bessel function $J_0(2\pi f_d \Delta t)$, where f_d is the maximum Doppler frequency and Δt is the time separation between two fading signals. The block diagram of wireless channel simulator [22] is shown in Fig. 2.8 and can be described as follows: Encoded video stream is converted to symbols using the Quadrature phase shift keying (QPSK) modulation. After Nyquist filter with the roll of factor is 0.25, the channel impulse response with Rayleigh fading of multi-path delay spread is calculated. At receiver side, a coherent receiver is used with an optimal symbol timing recovery and perfect carrier recovery. A maximal ratio combiner for antennal diversity combining is used. After that, signal is demodulated to get encoded video stream. In this system, the

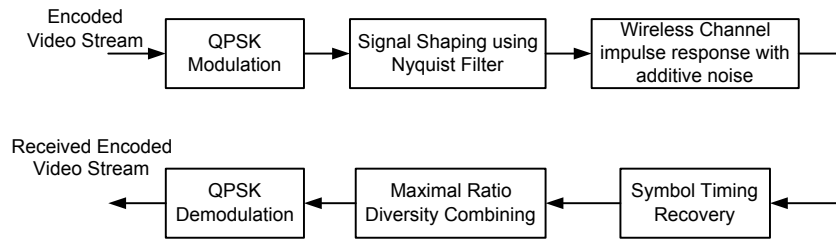


Figure 2.8: Block diagram of wireless channel simulator

Table 2.6: Wireless channel and air interface parameters used in this study

Multiple Access	TDMA
Modulation	QPSK
Channel rate	32kbps
Maximum Doppler Frequency	1Hz
Transmitted Signal Power	14dB
Time delay spread	$\frac{1}{4}$ of symbol period
Power delay profile	2-ray with equal power
Antenna diversity	1

multi-path model is an independent 2-ray fading model. The impulse response of the model is represented as $h(t) = a_1\delta(t - \tau_1) + a_2\delta(t - \tau_2)$ where a_i is the amplitude of the received impulse due to i^{th} path, and τ_i is the time delay of the i^{th} arrived impulse.

Table 2.6 shows parameters of our wireless channel simulation system. These parameters related to the channel characteristics, and the nature of the error. In other words, randomize of error depends on the setting values for parameters. Figure 2.9 and Fig. 2.10 show the error patterns in cases of slow fading and fast fading. Because Doppler frequency affects to ADF and LCR, the figures show that in the case of slow fading, the duration of fades are longer and the frequency of fades is lower. Conversely,

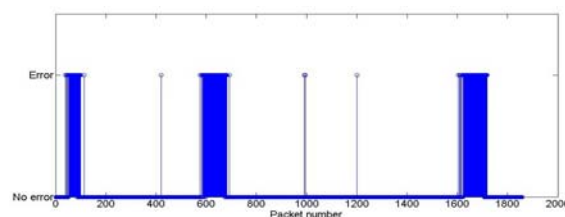


Figure 2.9: Error pattern with Doppler frequency of 1Hz

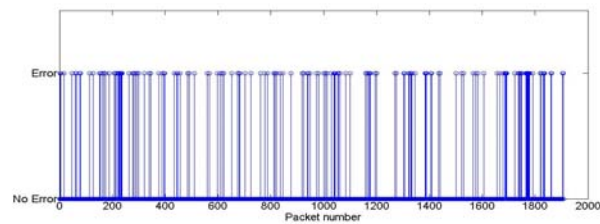


Figure 2.10: Error pattern with Doppler frequency of 40Hz

for fast fading, the error bursts are shorter, but occur more frequently.

2.3 Cross-Layer Optimization

2.3.1 Layers in Video Coding

H.264 consists of two conceptually different layers as shown in Fig. 2.11. First the VCL contains the specification of the core video compression engines that achieve basic function such as motion compensation, transform coding of coefficients, and entropy coding. This layer is transport-unaware, and its highest data structure is the video slice - a set of coded MBs in scan order. Second, the NAL is responsible for the encapsulation of the coded slices into transport entities of the network.

At network dependent protocol layer, the NALU is encapsulated into the RTP packet. RTP includes header information and payload. Header information includes CRC, sequence number, timestamp, The NALU is carried by payload in RTP. Since the NALU size can be varying and the RTP size is fixed, there is a need to fragment the NALU into several RTP packets. If the NALU size is less than the RTP payload size, then one complete NALU can be transmitted in a single RTP packet with the remaining bits padded with zero bits. If the NALU size is greater than the RTP payload size, then the NALU is fragmented into an equal length RTP payload size and the fragmented NALU is encapsulated into an RTP packet. Typically zero bits are padded into the last fragmented NALU to create a fixed length RTP packet.

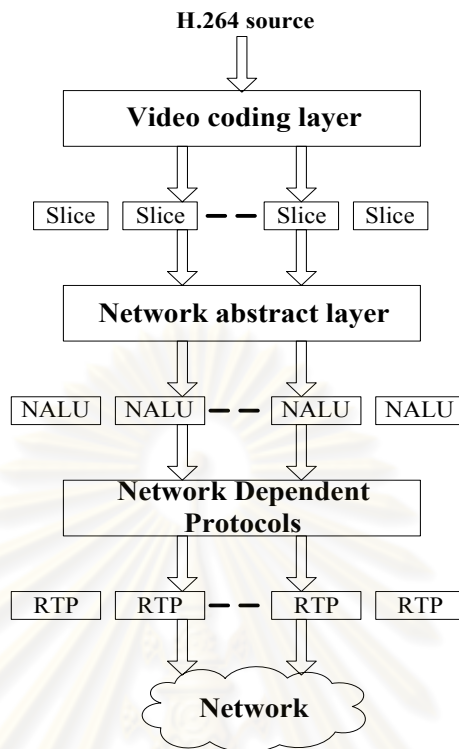


Figure 2.11: H.264 video transmission layers

2.3.2 Approaches in Cross-Layer

In recent years the research focus has been to adapt existing algorithm and protocols for multimedia compression and transmission to the rapidly varying and often scarce resources of wireless networks. However, these solutions often do not provide adequate support for multimedia applications in crowded wireless network. This is because the resource management, adaptation, and protection strategies available in the lower layers of the stack are optimized without explicitly considering the specific characteristics of multimedia applications, and conversely, multimedia compression and streaming algorithm do not consider the mechanisms provided by the lower layers. The independency in implementation of layers results in reduction in performance of system. To gain more effectiveness, some solutions are given to connect the operation between layers or cross-layer optimization in the system [23]:

- **Top-down approach:** The higher layer protocols optimize their parameters and the strategies at the next lower layer. This cross-layer solution has been deployed

in most existing systems, wherein the Application layer dictates the MAC layer parameters and strategies, while the MAC selects the optimize PHY layer module scheme.

- **Bottom-up approach:** The lower layers try to insulate the higher layers from losses and bandwidth variations. This cross-layer solution is not optimal for multimedia transmission, due to the incurred delays and unnecessary throughput reductions.
- **Application-centric approach:** The APP layer optimizes the lower layer parameters one at time in a bottom-up or top-down manner, based on its requirements. However, this approach is not always efficient, as the APP operates at slower timescales and coarser data granularity than the lower layers.
- **MAC-centric approach:** In this approach the APP layer passes its traffic information and requirements to the MAC, which decides which APP layer packets/flows should be transmitted and at what QoS level. The MAC also decides the PHY layer parameters based on the available channel information. The disadvantages of this approach resides in the the inability of the MAC layer to perform adaptive source channel coding trade-offs given the time-varying channel condition and multimedia requirements.
- **Integrated approach:** In this approach, strategies are determined jointly. However, to try all the possible strategies and requirements of all layers to find out an optimized set of parameters is impractical due to the associated complexity. A possible solution to solve this complex cross-layer optimization problem is in an intergrated manner is to use learning and classification techniques. For this, we identify content and network features that can easily be computed and are good indicators of which composite strategy is optimal.

2.3.3 Previous works using cross-layer approach

This section illustrates some methods using cross-layer design for optimized wireless multimedia transmission. With MAC-centric approach, in [24], parameters of PHY and APP layer are sent to MAC layer for optimizing the number of retransmission for each packets. Given the channel SNR and the PHY modes, the optimal packet size L^* that maximizes the goodput is computed as

$$L^* = \frac{L^{Header}}{2} + \frac{1}{2} \sqrt{(L^{Header})^2 - \frac{4b(L^{Header})^2}{\ln(1 - P_s)}} \quad (2.6)$$

where b is the number of bits per symbol and P_s is the probability of symbol error, which depends on the modulation type and link SNR.

Since in the delay-constrained wireless video transmission, the maximum number of times a packet j can be transmitted cannot actually be ∞ and is bounded by the delay deadline *Deadline*. The optimized limit number of retransmission for packet j at MAC layer is computed by:

$$m_j^{max,opt} = \left\lfloor \frac{Deadline - \sum_{k=1}^{j-1} m_k (L_k^{opt} / Rate_{PHY} + Time_o)}{L_j^{opt} / Rate_{PHY} + Time_o} \right\rfloor \quad (2.7)$$

where $\lfloor \cdot \rfloor$ is the floor operation. $Time_o$ is timing overhead for 802.11 MAC protocol.

Besides optimal MAC retry limit for each packet given the maximum available bit rate, the maximum tolerable delay, and the experienced bit error rate is calculated, in [25], a classifier is used to assigning a retry limit to each packet such that the expected overall distortion is minimized.

In above works, the problem of cross-layer optimization is considered in isolation, at each wireless station (WSTA). However, in wireless multimedia transmission systems, the cross-layer strategies adopted by the various WSTAs impact other competing stations. If a WSTA is adapting its transmission strategy, the delay and throughput of the competing stations are effected and, as a consequence, they may need to adjust

their own strategies. In [26], time fairness tries to deal with the problem by allocating each WSTA a fair share of time, which is proportional to the requirements mentioned in TSPEC, rather than guaranteeing a specific bandwidth. This proportional time allocation removes part of the unfairness resulting from the deployment of different cross-layer strategies by the various WSTAs. The time allocated to the stream i and j can be defined

$$\frac{T_i(t_1, t_2)}{T_j(t_1, t_2)} \geq \frac{\phi_i}{\phi_j}, j = 1, 2, \dots, N \quad (2.8)$$

ϕ_i is the weight of video flow i served in a time interval (t_1, t_2) .



CHAPTER III

Adaptive FMO Based on Error Propagation for Error Resilient H.264 Video Coding

In this chapter, we propose an error resilience scheme for wireless video transmission based on adaptive FMO and intra refresh. FMO explicit map is generated frame-by-frame using prior information. This information involves estimated locations of guard and burst sections in the channel and estimated effect of error propagation (EEP) from the previous frame to the current frame. In addition, the role of the current frame in propagating error to the next frame is also considered. Suitable intra refresh rate which adapts to channel state is used to reduce the dependency between frames and thus can stop the effect of error propagation. The results in experiments show that the proposed method gains some improvements in terms of peak signal to-noise rate (PSNR) as compared to some other methods that have not considered channel condition and error propagation in generating FMO map.

The rest of this chapter is organized as follows: In Section 3.1, we present the effect of error propagation and literature reviews of previous works related to error propagation. The proposed method in locating and generating explicit FMO map is introduced in Section 3.2 and 3.3, appropriately. Finally, simulation results and discussion are presented in Section 3.4.

3.1 Introduction

In H.264, MBs can be grouped into slice groups. Structure of a slice group consists of a resynchronization marker, a header field and a series of coded macroblocks in raster scan order. Each MB is coded in order: Motion estimation and compensation (inter coding), discrete cosine transform, quantization and entropy coding. In H.264, variable length code (VLC) [27] is used in entropy coding. One serious drawback of VLC is

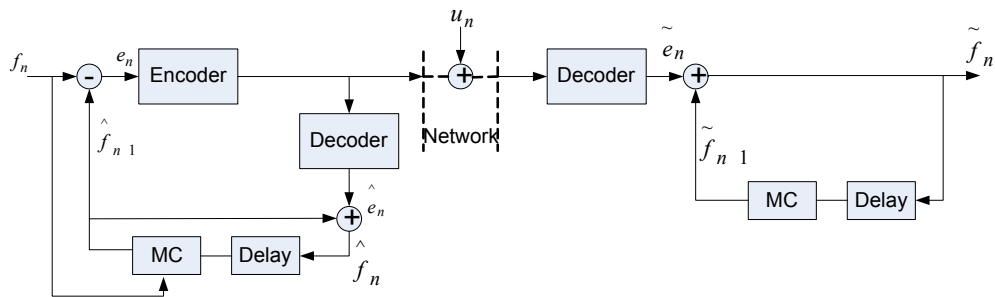


Figure 3.1: Simple codec structure of H.264

that they are sensitive to transmission errors. An error in a sequence of VLC may cause the bit stream to lose synchronization and fail to decode subsequent codes correctly, leading to error propagation in a decoded sequence. Thus, if a MB is error, the rest of MBs in the slice groups are error.

While DCT is used in the decoder to exploit spatial redundancy, inter frame prediction is used to exploit temporal redundancy to achieve a high compression ratio. However, the disadvantage of this technique is inter-frame error propagation. To understand the effect inter-frame error propagation to overall performance of hybrid video coding, consider the simplified codec structure in Fig.3.1. At the encoder, the original frame f_n is predicted by a motion compensated previous frame and the prediction error e_n is transmitted over the channel. Assume that there is error on frame n , where the signal u_n is added to the prediction error e_n . At the receiver, because of channel error, the output of decoder is different from the output of decoder at the transmitter, $\hat{e}_n \neq \tilde{e}_n$. However, due to the recursive structure of the decoder, not only the received frame \tilde{f}_n is error but also the following frames are error. As the result, the error in transmission will be propagated to the next frames in group of pictures (GOP) until resynchronization bits of the next GOP are inserted in bit stream.

In recent years, there are many researches proposing model for inter-frame error propagation. In [28], by analyzing the effects of error propagation, it provides useful indicator such that appropriate intra refresh rate is determined to minimize the effect of errors. Reyes et al. [29] presented a corruption model for error propagation by using Markov model, which takes into account the parameters that characterize spatial

resilience, temporal resilience, and the source rate. Base on this model, the bit rate is allocated optimally to minimize distortion in video. In [30], authors estimated the packet-level error propagation. This measurement is taken as the estimation of the packet loss impact value of each packet. According to the packet-level loss estimation and the feedback information from ARQ, packets are assigned an appropriate priority. Based on the packet's loss impact, the lower priority packets will be dropped to save bandwidth to retransmit the important lost packets under a retransmission delay constraint. In [31], [32], authors developed a statistical model to describe the overall behavior of the channel distortion. The method recursively computes the total decoder distortion at pixel level precision to accurately account for spatial and temporal error propagation. The estimated distortion is integrated into a rate-distortion framework for optimal switching between intra-coding and inter-coding modes per macroblocks.

There is also approach using joint source channel rate-distortion analysis. In [33], some MBs with higher channel distortion in a frame have been selected to be forcedly intra-coded. All the MBs in the current frame will be mapped to two slice groups: slice group 0 and slice group 1. The intra MBs will be assigned into slice group 0 and the inter MBs into slice group 1. Obviously, slice group 0 has a higher importance than slice group 1. And thus, slice group 0 is protected by using a stronger Reed Solomon code. Similarly, to generate FMO map, in [34], MBs first are dispersed into two slice groups (SG1 and SG2) and both slice groups are further splitted into more SGs according to the impact factor of MBs. In order to make more efficient use of FEC, MBs with similar impact factors are grouped into the same slice group. In this work, k-means clustering algorithm is used to classify MBs. The k-means clustering classified this given data set into a certain number of clusters (assume k clusters) with a distance as far away from each other as possible. Each MB is regarded as a single data point and the impact factor as its position in the coordinate axis for the algorithm. The k-means algorithm clusters the impact factors of MBs into several variable size subsets and generates a FMO table accordingly.

In this work, we will consider the effects of error propagation and propose an approach of how FMO map could be generated given considering error propagation

effects.

3.2 Locating Error Burst Positions

Assume that before encoding the current frame (n), encoder receives channel feedback information containing position of error packets of the frame ($n - 2$). It is practical to assume that the roundtrip delay in wireless networks is less than 100ms. Before encoding the current frame (n), encoder receives channel feedback information containing position of error packets of the frame ($n - 2$) after one-frame feedback delay. There are some reasons why we should choose this period for feedback delay in system. The feedback delay is set based on the assumption of the system that the roundtrip delay time is less than 100 ms. For the channel bit rate of 32kbps and frame rate of 10fps, the feedback of one frame is possible. Nevertheless, the authors also evaluated the proposed method when the feedback delay is more than one frame, i.e., feedback delay = 2, 4 frames. The average PSNR values of four sequences are shown in Fig. 3.2. In summary, we found that the longer feedback delay affects the decrease in PSNR value. Because of propagation, we should stop error as fast as possible. As shown in Fig.3.3, if the feedback delay is two frames, the frame ($n - 2$) and frame ($n - 1$) will be affected by error propagation can be stopped sooner, at frame ($n - 1$). Thus, the system with lower feedback delay would yield better performance.

Based on feedback information, encoder uses Eq. (2.5) to estimate values L_G and L_B of channel. These values are used for the whole transmission duration of current frame (n) and frame ($n - 1$). Then values of L_G and L_B are updated when encoder receives the next feedback information. To estimate the first location of burst or guard section in frame ($n - 1$), position of the last burst or guard section of frame ($n - 2$) is used. If border between frame ($n - 2$) and frame ($n - 1$) is in a guard section (see Fig. 3.4), distance from the last burst in frames ($n - 2$) to the first burst in frame ($n - 1$) is L_G packets. Otherwise, if the border is in a burst section, the distance from the last guard in frame ($n - 2$) to the first guard in frame ($n - 1$) is L_B packets. The next sections in frame ($n - 1$) and frame (n) are estimated from the position of the first section.

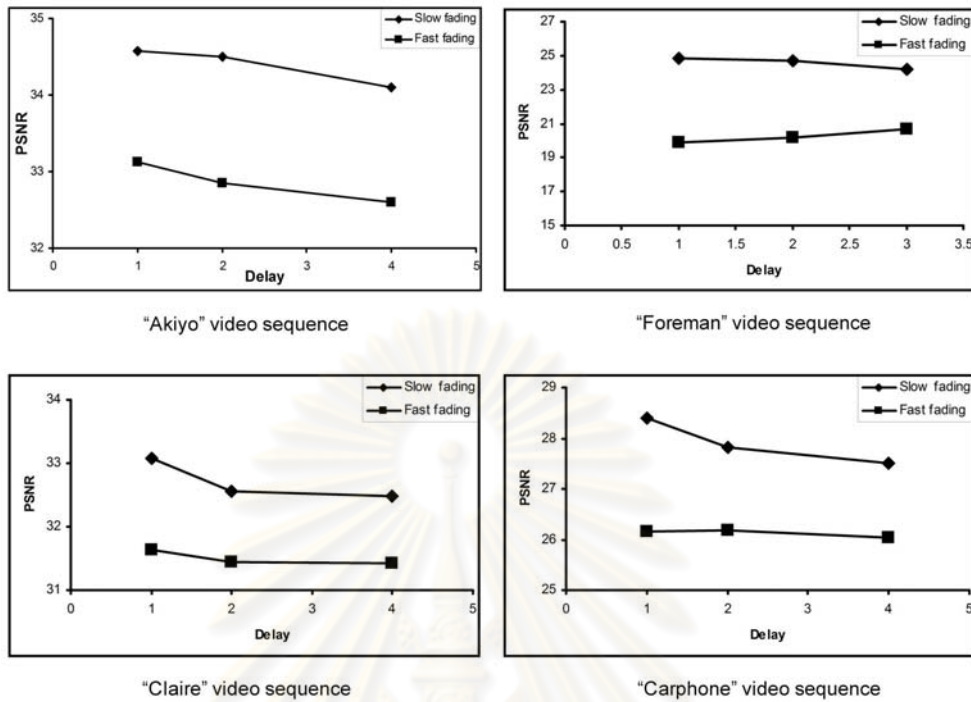


Figure 3.2: Average PSNR of the proposed method for video sequences with difference feedback values.

In this work, to validate the accuracy in locating burst sections of wireless channel, the estimated burst locations are compared with the actual burst locations of wireless channel which are generated from wireless channel simulator [22]. Figure 3.5 and 3.6 show actual burst locations of wireless channel at different bit rates. For the data rate of 32 kbps, the bit duration is about 0.03 ms. For the data rate of 64 kbps, the bit duration is 0.015 ms and equals to a half of 32 kbps. However, with the same Doppler frequency and threshold level, LCR and AFD of a Rayleigh fading signal are constant. Thus, with the same Doppler frequency and threshold, the number of error bit in 64 kbps channel will be double that of 32 kbps channel. That why the burst length and guard length (in packet) of 64 kbps is double the burst length of 32 kbps, the burst length and guard length of 128 kbps is double that of 64 kbps and so on.

However, because error patterns are unknown, system has to use three-state Markov model to predict locations of burst and guard section in channel. To validate the correctness of the proposed method, the accuracy in locating burst and guard section of

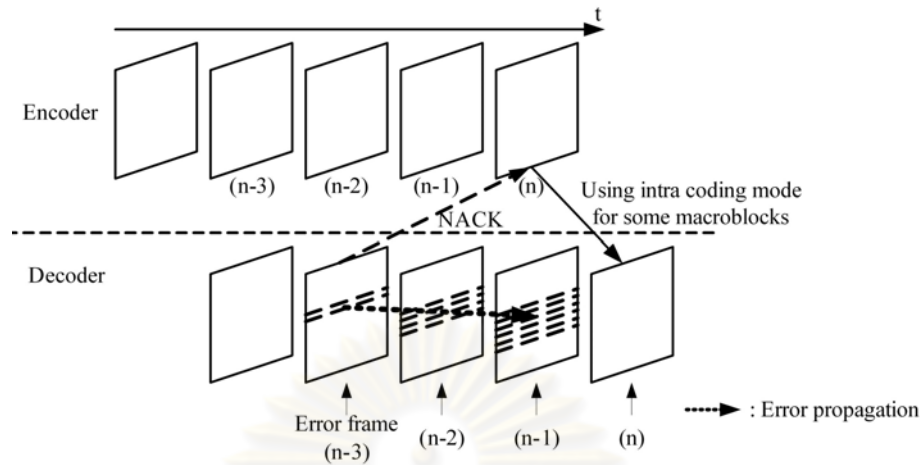


Figure 3.3: Effect of error propagation with two frames for feedback delay.

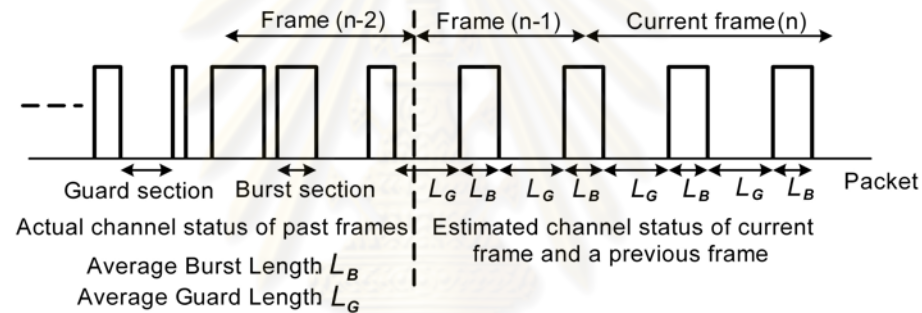


Figure 3.4: Locating burst and guard section for the current frame.

wireless channel is tested. Figure 3.7 and 3.8 illustrate the estimation of burst and guard locations for 100 frames of Akiyo sequence in fast and slow fading. Figure 3.7(a) and 3.8(a) show the estimation of the locations of burst and guard sections when transmitting Akiyo sequence over error channel. Figure 3.7(b) and 3.8(b) show the actual locations of burst and guard. They are obtained from wireless channel simulator in [22]. The differences between actual locations and estimated locations include two cases:

- A packet in the actual channel is in burst section but in the estimated channel, it is in guard section.
- A packet in the actual channel is in guard section but in the estimated channel, it is in burst section.

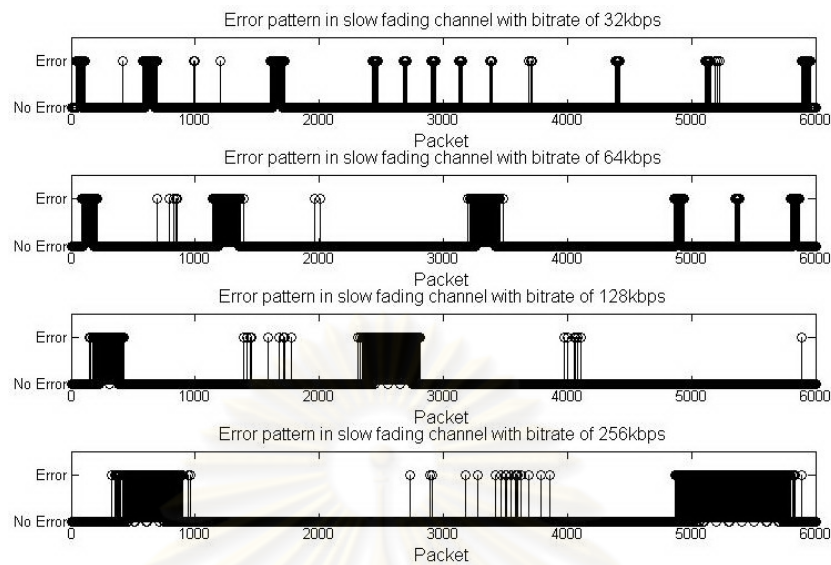


Figure 3.5: Error patterns of wireless channel simulator in slow fading at different bit rates.

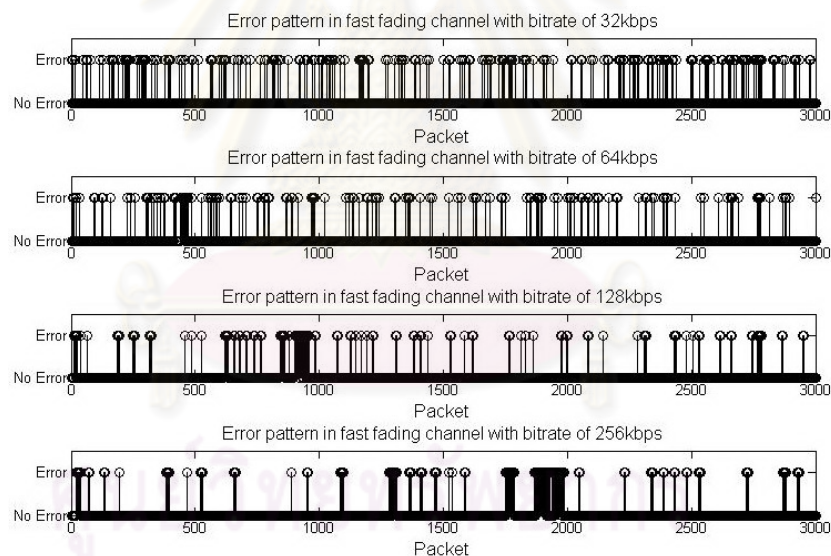


Figure 3.6: Error patterns of wireless channel simulator in fast fading at different bit rates.

The estimation mismatches in the first case where the actual channel state is good do not have much affect to the result. Thus, we consider only the second case, i.e., a channel state is bad and encoder estimates that it is good. Figure 3.7(c) and 3.8(c) shows the second case only. The other comparison between actual and estimated channel state at different bit rates are shown in Appendix A. For the data rate of 32 kbps, the

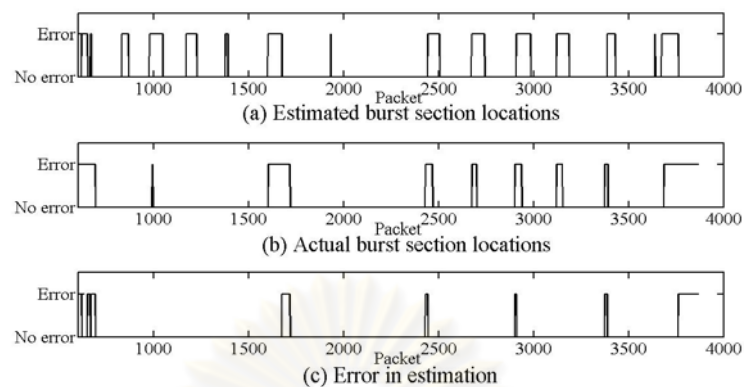


Figure 3.7: Comparison actual and estimated channel state for "Akiyo" sequence in slow fading, 32 kbps.

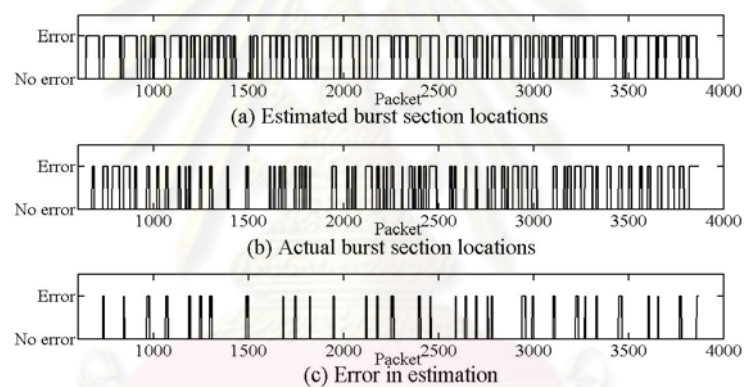


Figure 3.8: Comparison actual and estimated channel state for "Akiyo" sequence in fast fading, 32 kbps.

bit duration is about 0.03 ms. However, Fig.A.1-A.2 show that with the data rate of 64kbps, the bit duration is 0.015 ms and equals to a half of 32 kbps. Because level crossing rate (LCR) and average fade duration (AFD) of a Rayleigh fading signal are constant with the same Doppler frequency. Thus the number of error bit in 64 kbps channel will be double that of 32 kbps channel. Similarly, in Fig. A.3-A.6, the burst length (in packet) of 256 kbps is double the burst length of 128 kbps, the burst length of 128 kbps is double the burst length of 64 kbps and so on.

Table 3.1 shows the percentage difference of four video sequences "Akiyo", "Foreman", "Claire" and "Carphone" in both slow and fast fading cases. The percentage

difference is calculated by

$$P_d = \frac{e}{T} \cdot 100\% \quad (3.1)$$

where P_d is the percentage difference. e is the number of errors in estimation. T is total number of packet of a video sequence.

Table 3.1: Percentage difference (P_d) between estimated and actual locations of burst and guard sections

Bitrate	Akiyo		Foreman		Claire		Carphone	
	Slow	Fast	Slow	Fast	Slow	Fast	Slow	Fast
32 kbps	2.5	5.2	4	5.7	3.2	3.3	3.7	4.5
64 kbps	2.5	4.6	2.1	3.1	2.1	2.4	3.4	3.6
128 kbps	2.7	5.1	2.5	4.3	3.2	4.1	2.1	3.3
256 kbps	2.4	4.3	2.0	5.2	2.8	5.9	2.2	3.7

3.3 Adaptive Explicit FMO Map Generation

In this section, the importance of an MB is estimated from the distortion caused by error propagation. After that, an explicit FMO map of the current frame is generated by mapping high important MBs and low important MBs into slice groups which are transmitted in guard and burst sections appropriately.

3.3.1 The Estimation of MB Importance

To estimate the importance for an MB in the current frame, firstly, we estimate the distortion at that MB caused by error MBs in the past frame. Secondly, we measure EEP caused by that MB to the next frame. Sum of distortion and EEP is considered as the importance of that MB. This importance in company with estimated burst locations are taken into account to generate the FMO map and to decide inter/intra coding mode for that MB. Figure 3.9 describes the error propagation from the past frame ($n - 2$) to the current frame (n) and to the next frame ($n + 1$).

Compute the distortion at the frame ($n - 1$) caused by an error pixel in the frame ($n - 2$):

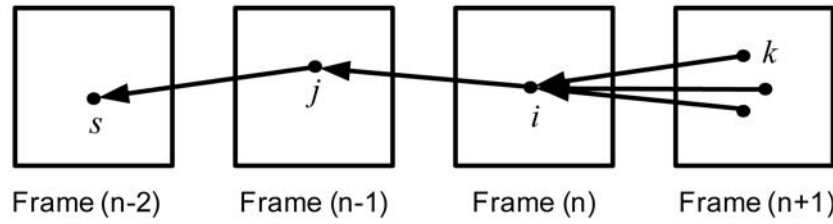


Figure 3.9: Error propagation from the past frame to the next frame.

Assume that a pixel j in the frame $(n - 1)$ refers to a pixel s in the frame $(n - 2)$.

If the pixel s in the frame $(n - 2)$ is in error, the decoder will copy the pixel s of the frame $(n - 3)$ with assuming that the non-motion compensated error concealment method is used. Therefore the distortion at the pixel j in the frame $(n - 1)$, $D(s, j, n - 1)$, is computed as

if j is an inter coded pixel:

$$D(s, j, n - 1) = \left| |f(j, n - 1) - f(s, n - 2)| - |f(j, n - 1) - f(s, n - 3)| \right| \quad (3.2)$$

if j is an intra coded pixel:

$$D(s, j, n - 1) = 0$$

where $f(x, y)$ is reconstructed value of the pixel x^{th} in the frame y^{th} .

If the pixel s in the frame $(n - 2)$ is error-free, the distortion at the pixel j in the frame $(n - 1)$ is computed in Eq. (3.3).

$$D(s, j, n - 1) = 0 \quad (3.3)$$

Step 2: Compute distortion at the frame (n) caused by an error pixel in the frame $(n - 2)$:

Assume that the pixel i in the frame (n) refers to the pixel j in the frame $(n - 1)$.

If the pixel j is inter coded, the distortion at the pixel i is computed as:

if j is error:

$$D(j, i, n) = \left| |f(i, n) - f(j, n - 1)| - |f(i, n) - f(j, n - 2)| \right| \quad (3.4)$$

if j is error-free:

$$D(j, i, n) = D(s, j, n - 1)$$

In conclusion, in the case pixel j is inter coded, the distortion at the pixel i is computed as

$$D(j, i, n) = q(j) \left| |f(i, n) - f(j, n - 1)| - |f(i, n) - f(j, n - 2)| \right| + (1 - q(j))D(s, j, n - 1) \quad (3.5)$$

where $q(j)$ is error probability pixel j . $q(j)$ depends on the error probability of packets and the length in packets of the MB containing pixel j .

If the pixel j is intra coded, the distortion at the pixel i is computed as

if j is error:

$$D(j, i, n) = \left| |f(i, n) - f(j, n - 1)| - |f(i, n) - f(j, n - 2)| \right| \quad (3.6)$$

if j is error-free:

$$D(j, i, n) = 0$$

In conclusion, in the case j is intra coded, the distortion at pixel i is computed as shown in Eq. (3.7).

$$D(j, i, n) = q(j) \left| |f(i, n) - f(j, n - 1)| - |f(i, n) - f(j, n - 2)| \right| \quad (3.7)$$

Step 3: Compute EEP at the frame $(n + 1)$ caused by pixel i in the frame (n) :

To compute EEP from the past frame to the next frame through the current frame, all MBs in the current frame are coded in inter mode in the first pass. Assume that the pixel i in the current frame is referred by a pixel k in the next frame. The distortion caused by error propagation for the pixel k is computed by

$$D(i, k, n + 1) = q(i) \left| |f(k, n + 1) - f(i, n)| - |f(k, n + 1) - f(i, n - 1)| \right| + (1 - q(i))D(j, i, n) \quad (3.8)$$

where $D(j, i, n)$ is computed as shown in Eq. (3.5) or Eq. (3.7) depending on coding mode of the pixel j . The overall distortion propagated from pixel i in the current frame (n) to the next frame $(n + 1)$ is computed as

$$I(i, n) = \sum_{k \in \{N\}} D(i, k, n + 1) \quad (3.9)$$

where N is the number of pixels in the frame ($n + 1$) which refer to the pixel i in the current frame (n).

Step 4: Estimate the importance of an MB in the current frame:

The importance of the l^{th} MB in the current frame is computed as

$$D_l(n) = \sum_{i,j=1}^M D(j, i, n) + \sum_{i=1}^M I(i, n) \quad (3.10)$$

where $D(j, i, n)$ is computed in Eq. (3.5) or Eq. (3.7). $I(i, n)$ is computed in Eq. (3.9). M is the number of pixels in an MB. In this case, $M = 256$.

3.3.2 Adaptive Explicit FMO Map Generation

Flow chart of algorithm generating an explicit FMO map for the current frame is shown in Fig. 3.10. In the first pass, all MBs in the P frame are encoded in inter mode to get motion vector and number of packets of MBs. In the second pass, after getting feedback information from decoder, L_G and L_B are computed. After that, the positions of burst and guard sections are predicted for the current frame. Based on the importance computed in Eq. (3.10), MBs are selected to fill in a section until the total number of packets of selected MBs equals to the length of section. The number of slice groups in a section, N_{slg} , is computed by

$$N_{slg} = \lceil \frac{N_{mb}}{\gamma} \rceil \quad (3.11)$$

where $\lceil z \rceil$ is the smallest integer not less than z . N_{mb} is the number of MBs selected to fill in a burst or guard section. For QCIF frames, to guarantee 99 MBs are fairly distributed in 8 slice groups, the number of MBs in a slice group, γ , is selected to 13. In the guard section, some MBs with high importance are selected for intra coding mode. The number of intra MBs in the guard section, N_{intra} , is computed as

$$N_{intra} = \lambda N_{slg} \quad (3.12)$$

To balance the compression efficiency and the effect of error propagation, λ is

empirically selected to 2. According to Eq. (3.12), the number of intra MBs in frames varies from 0 (if there is no guard section in transmission duration of frame) to 16 (if the whole frame is transmitted within a guard section). Intra coded MBs are dispersed into some slice groups in a guard section to reduce the number of lost intra coded MBs in case channel prediction is not precise.

The algorithm is described through the following example: After the first pass, motion vector and bit count of MBs are derived. Based on motion vector, bit count and feedback information, in the second pass, encoder estimates EEP values for MBs. Figure 3.11 shows an example of bit count and EEP of MBs in frame 29th in "Akiyo" sequence.

After sorting in descending order of EEP (Fig.3.12-a, we have appropriate bit-count and id of MBs shown in Fig.3.12-b and Fig.3.12-c. Figure 3.13 shows the estimated locations of error burst. The number of packets in frame 29th is 30 packets and the first packet of frame is 1480th in video sequence. The figure is extracted from Fig.3.7-a and focus on offset from packet 1480th to packet 1510th.

In the next step, MBs are arranged into guard and burst sections as the rule: MBs with high EEP are arranged in guard section until the sum of bitcount of frame fill up the length of section. Similarly, MBs with low EEP are arranged in burst sections. The detail of algorithm is described as following:

- The length of the first guard section is 800 bits (10 packets x 80 bit/packets). Thus, MB id (selected from top to bottom) 96, 54, 92, . . . , 32, 47, 83 (yellow MBs in Fig. 3.12-c) with the total bitcount is 769 bits ($57 + 0 + 44 + 9 + \dots + 17 + 84 = 769 < 800$, Fig. 3.12- b) are put into the first section.
- The length of next burst section is 560 bits (7 packets). Thus, MB id (selected from bottom to top) 79, 78, 68, . . . , 99, 51 (red MBs in Fig. 3.12 - c) with the total bitcount is 524 ($0 + 0 + 0 + \dots + 46 + 85 = 524 < 560$, Fig. 3.12- b) are put into this section.
- The length of the next guard section is 800 bits (10 packets). MB id 37, 50, 72, . . . , 62, 22, 31 (green MBs in Fig. 3.12 - c) with the total bitcount is 790 bits ($63 +$

$61 + 182 + \dots + 29 + 17 + 0 = 790 < 800$, Fig. 3.12- b) are put into this section.

- The rest of MB ids (magenta MBs in Fig. 3.12 - c) are put into the last guard section.
- In a slice group, if an MB is error, the rest of MBs in the slice group also is error. Therefore, to reduce the number of error MBs in a slice group, MBs in each section are dispersed into two smaller slice groups.

Finally we have a completed explicit FMO map with 8 slice groups shown in Fig. 3.14 (MBs in a slice group are sorted in increasing order of ID). Note that all above processing are computed before encoding the frame. After generating explicit FMO map, macroblocks are encoded and transmitted following ID order as shown in FMO map.

With the above arrangement, the MBs with high EEP value (yellow and magenta) are transmitted in guard section. MBs with low EEP value (green and red) are transmitted in burst section. Therefore, the number of lost important MBs is reduced.

3.4 Simulation Results and Discussions

3.4.1 Experimental Setup

In the study, wireless channels are simulated for video transmission using 2-ray Rayleigh Fading channel [9]. The block diagram of the wireless channel simulator and the system parameters are shown in Fig. 2.8 and Table 2.6. One important wireless channel simulator's parameter is Doppler frequency. In this research, we are interested to test our technique in both slow fading channel, i.e., Doppler frequency = 1Hz, and fast fading channel, i.e., Doppler frequency = 40Hz. We used H.264 reference software JM 9.2 [35] with baseline profile in simulation. Video sequences in QCIF format (176x144 pixels/frame) are coded at 32kbps with a frame rate of 10fps. The following video sequences are used in the experiment: Akiyo, Foreman, Claire, and Carphone. At the decoder, the non-motion compensated error concealment is used.

Table 3.2: Comparison of Average PSNR (dB)

PSNR (dB)	Akiyo		Foreman		Claire		Carphone	
	Slow	Fast	Slow	Fast	Slow	Fast	Slow	Fast
No FMO	30.22	26.59	17.67	15.54	28.37	24.65	22.41	19.75
Bitcount	34.62	32.14	21.28	18.32	32.14	28.87	25.34	22.95
STI-FMO	33.75	30.7	21.56	18.37	32.20	28.94	25.55	22.45
Proposed method (no IR + no CP)	34.29	31.29	21.8	19.5	33.19	28.8	25.98	24.41
Proposed method (no IR + CP)	35.02	32.71	23.82	18.4	33.9	30.45	27.58	24.61

3.4.2 Simulation Results

In this experiment, to evaluate the efficiency of using EEP as an indicator for FMO as well as the effectiveness of channel prediction method, the proposed method is compared to some other methods using different indicators, including bitcount [1] and spatial-temporal indicator (STI) [7]. Moreover, to validate the method of selecting MBs for intra coding mode and computing intra refresh rate, the proposed method is compared to methods using fix intra refresh rate (FIR) [28] and random intra refresh rate (RIR) [35]. In comparisons, we use both subjective and objective measures. For objective measure, PSNR is used as the performance metric in quantifying the effectiveness of methods.

Table 3.2 shows the average PSNR of video sequences in the scenario of slow and fast fading channels. The simulation results show that the proposed method without intra refresh and channel prediction (no IR + no CP) gains higher average PSNR than the conventional methods. Especially, if compared to the method of not using FMO, the improvement of average PSNR is up to 5 dB. However, in some cases, PSNR of the new method is lower. This is because the quality of measurements of video in terms of PSNR depends solely on the locations of bit errors as well as the error concealment method applied. In this experiment, simple non-motion compensated error concealment is used, therefore we expect that the higher PSNR improvement can be achieved if more sophisticated technique of error concealment is used in further study. Results

show that the average PSNR is improved when channel prediction is applied to the proposed method. This is because the number of lost important MBs is reduced when the locations of burst and guard sections are estimated. However, in the fast fading case, the improvement does not significantly increase in comparison with the case without channel prediction. The reason is that channel prediction algorithm is more precise in slow fading case. In fast fading case, there are more errors in locating burst and guard sections. Therefore, the number of lost important MBs in fast fading case is higher than that in slow fading case. Figure 3.15 and 3.16 show PSNR curve of Carphone test sequence in the slow and fast fading case, respectively. From the curves, it can be observed that the average PSNR of the proposed method is higher than the others. This improvement is achieved by using an accurate method in stopping the effect of error propagation. Furthermore, by estimating locations of burst and guard sections, important MBs are put in error-free sections, thus the number of lost important MBs is reduced. Consequently, the PSNR of the new method is increased.

Table 3.3 shows average PSNR of video sequences when comparing proposed method using adaptive intra refresh rate with other methods using fix and random intra refresh rate. In the first scene, the proposed method uses a fix intra refresh rate without channel prediction (FIR + no CP). The fix intra refresh rate is 11 MBs per frame. The results show that with considering effect of error propagation from the current frame to the next frame, the proposed method has higher average PSNR than the method used in [28]. In [28], only errors propagated from the previous frame to the current frame are taken into account. Consequently, some MBs in the current frame are skipped in evaluating importance because these MBs are not much affected by error propagation from the past frame. However, these MBs may cause a high distortion for the next frame. Therefore, it is necessary to consider both effects of error propagation from the previous frame to the current frame and from the current frame to the next frame. In the second scene, the proposed method uses an adaptive intra refresh rate with consideration of channel prediction (AIR + CP). By locating the burst and guard sections, intra coded MBs are more guaranteed than the other methods. Thus, average PSNR of proposed method is higher.

Table 3.3: Comparison of Average PSNR (dB)

PSNR (dB)	Akiyo		Foreman		Claire		Carphone	
	Slow	Fast	Slow	Fast	Slow	Fast	Slow	Fast
RIR	34.39	30.09	21.77	20.61	31.47	27.61	25.04	22.25
FIR	34.36	30.05	21.29	17.5	33.59	30.69	26.42	24.53
Proposed method (FIR + no CP)	34.05	31.85	22.49	19.16	32.6	30.78	27.39	23.87
Proposed method (AIR + CP)	34.58	33.13	24.83	19.9	33.08	31.63	28.39	26.15

Figure 3.17 and 3.18 show the PSNR curve of methods using “Carphone” video sequence in slow and fast fading. In the slow fading case, because of channel prediction and adaptive intra refresh rate, the PSNR curve of the proposed method is higher than that of “RIR” and “FIR”. Since channel prediction is less precise in fast fading, average PSNR of “AIR + CP” is lower than “FIR + no CP” from the frame 71 to the frame 100 (Fig. 3.18). However, the results in Table 3.3 show that the average PSNR of “AIR + CP” is still higher than the other methods. Comparisons between the proposed method with some other methods at different bit rates are shown in Appendix B. We can see that in slow fading cases (Fig.B.1-B.16), the proposed method achieves higher average PSNR in all bit rates. However, this may be not true in the case of fast fading. This is because in fast fading, the channel prediction is not as precise as the prediction in the slow fading.

To further illustrate improvement of the proposed method, some frames from the “Carphone” test sequence are extracted for comparison. Figure 3.19 depicts qualities of original 49th frame of “Carphone” sequence and the reconstructed frames from four different methods including “RIR”, “FIR”, “FIR + no CP” and “AIR + CP” in slow and fast fading. It can be subjectively seen that the frame quality in “RIR” is severely affected by error. However, in proposed method, this error can be substantially improved by using intra refresh and channel prediction.

3.5 Summary

In this chapter, we measured the effect of error propagation from the past frame to the next frame. Based on the effect of error propagation, importance of MBs in the current frame is estimated. We also use three-state Markov model to estimate locations of burst in wireless channel. To reduce the number of lost important MBs, an explicit FMO map for the current frame is generated as the rule: The high estimated importance MBs are mapped into slice groups which are transmitted in guard sections of channel. The low estimated importance MBs are mapped into slice groups which are transmitted in burst sections. As the result, our proposed scheme show better results than some other method. In particular, comparing to the previous method using different indicator for generating FMO such as bitcount and spatial-temporal indicator, our method using effect of error propagation has 4 dB higher in average PSNR. Comparing to some other method using intra refresh to stop error propagation such as random and fix intra refresh, our method gives 3dB higher in average PSNR.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

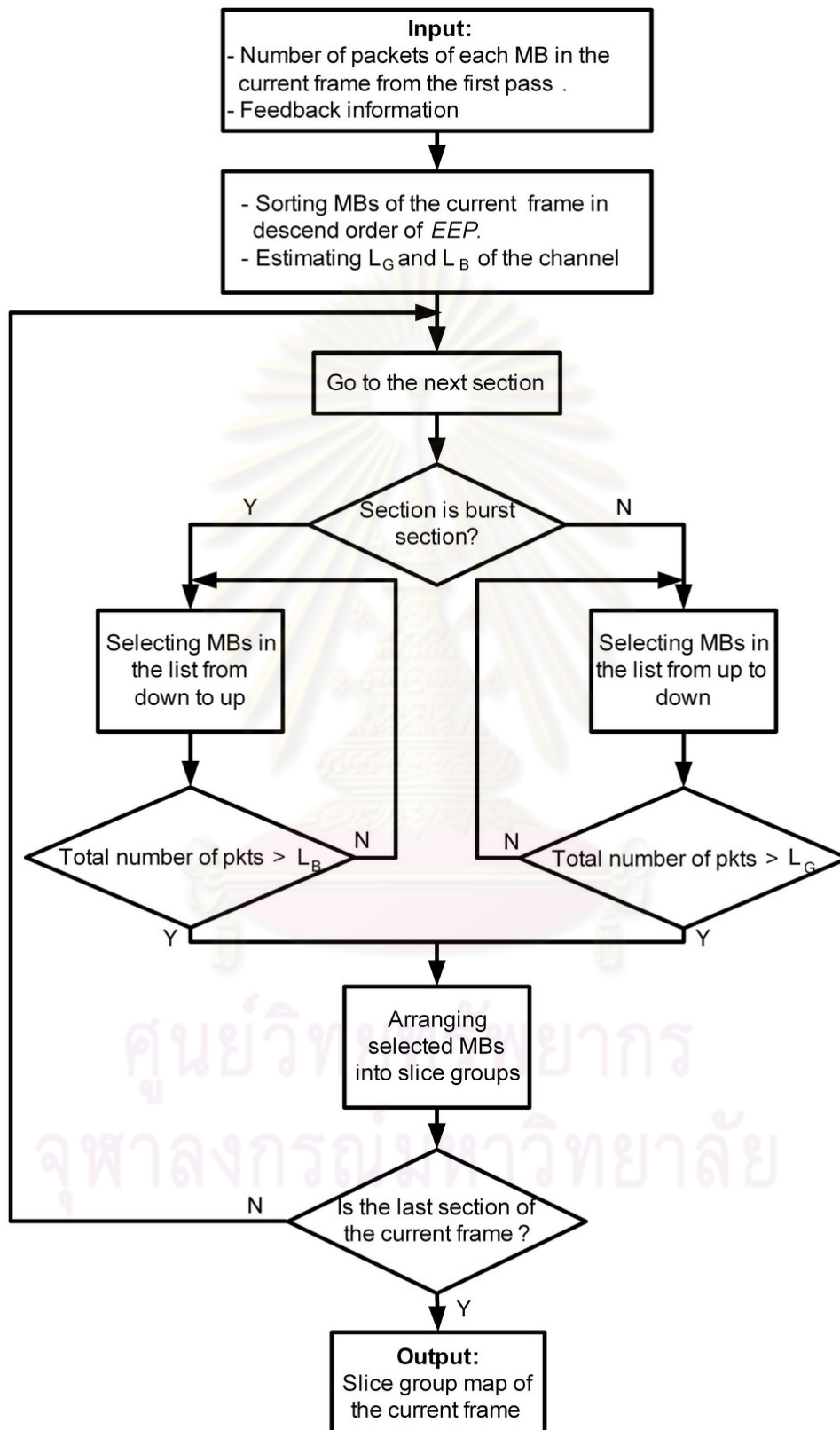


Figure 3.10: Flow chart of the proposed method.

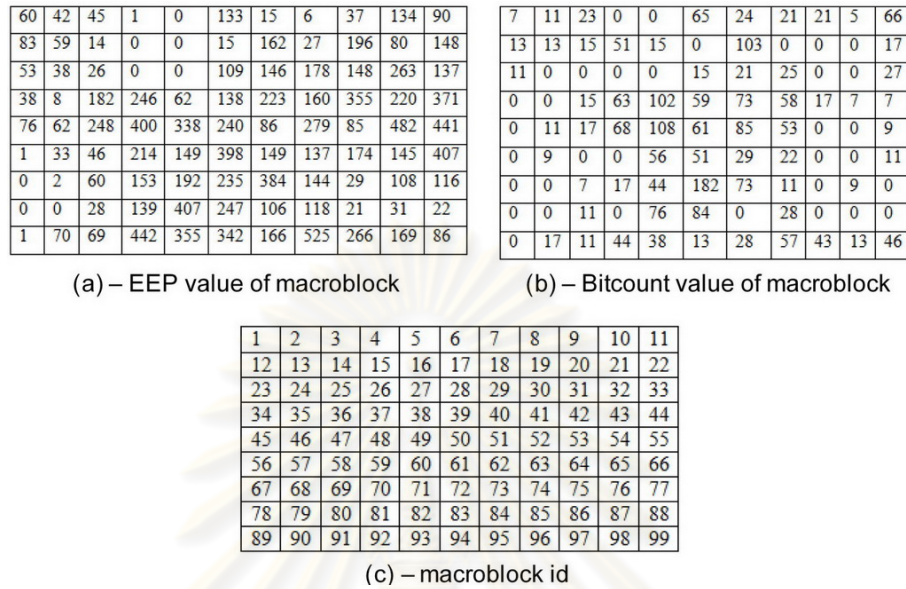


Figure 3.11: EEP, Bitcount and ID values of MBs in frame 29th before sorting.

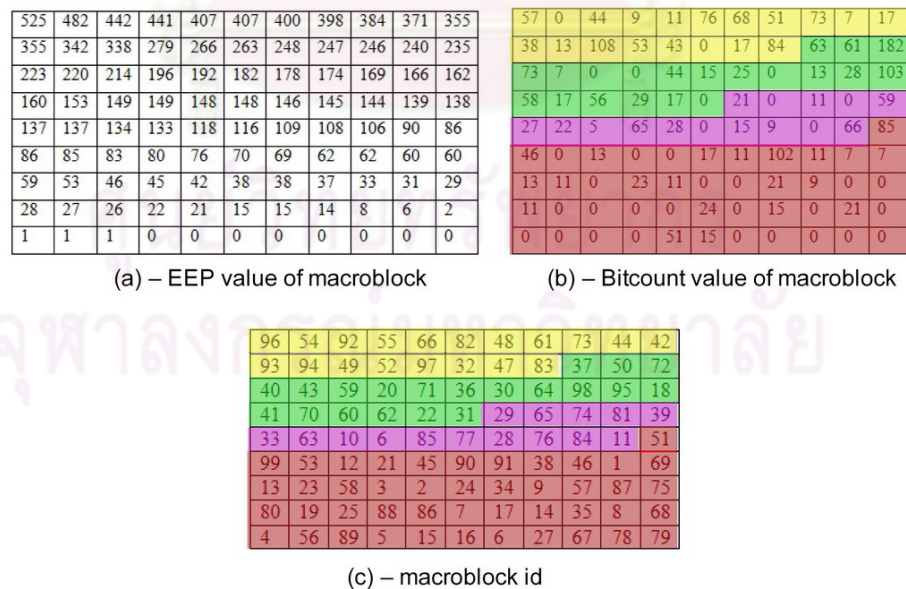


Figure 3.12: EEP, Bitcount and ID values of MBs in frame 29th after sorting.

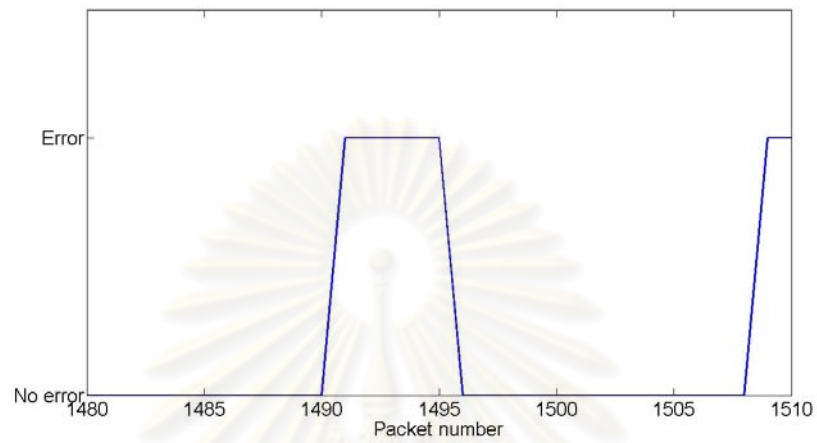


Figure 3.13: Estimation of error burst and guard locations in wireless channel.

32	42	48	52	66	73	83	92	94	96	44
47	49	54	55	61	82	93	97	20	22	36
37	41	43	60	64	72	95	18	30	31	40
50	59	62	70	71	98	6	11	29	39	63
74	76	77	10	28	33	65	81	84	85	1
2	4	6	7	8	13	14	15	19	21	34
38	51	53	57	58	67	75	79	88	89	90
3	5	9	12	16	17	23	24	25	27	35
45	46	56	68	69	78	80	86	87	91	99

Figure 3.14: Explicit FMO map.

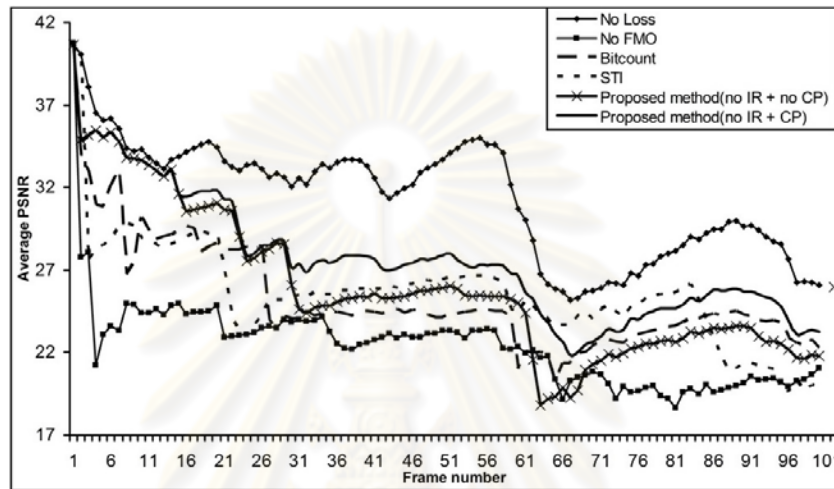


Figure 3.15: PSNR comparison of “Carphone” in slow fading.

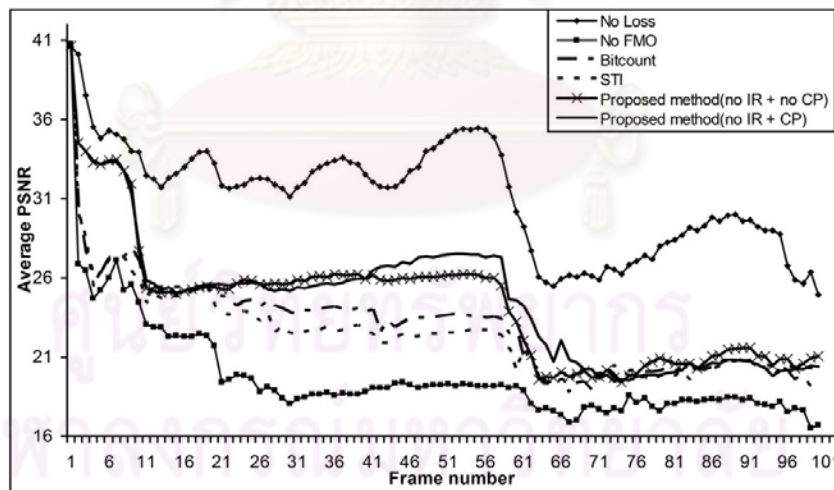


Figure 3.16: PSNR comparison of “Carphone” in fast fading.

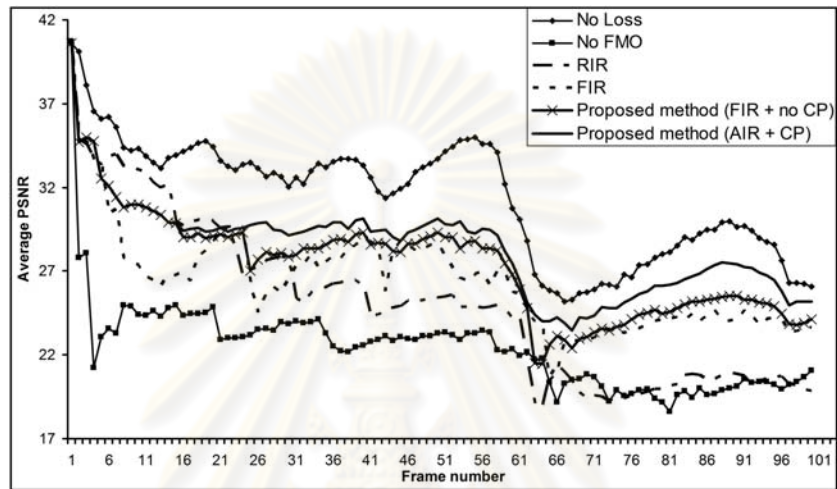


Figure 3.17: PSNR comparison of “Carphone” in slow fading.

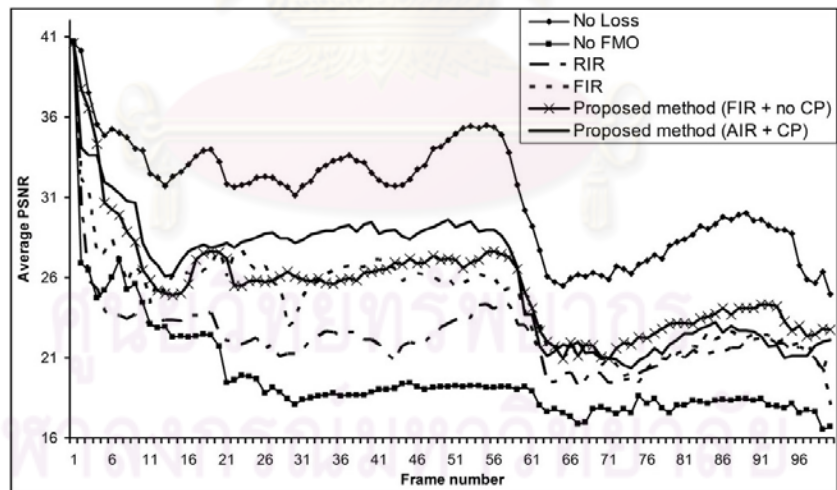


Figure 3.18: PSNR comparison of “Carphone” in fast fading.



(a) Original frame



(b) Comparison in slow fading case



(c) Comparison in fast fading case

Figure 3.19: Visual comparison of frame 49th of “Carphone” sequence between methods in slow and fast fading

CHAPTER IV

Adaptive FMO Using Cross-Layer Application-MAC Layer Scheme

In this chapter, cross-layer scheme is used to generate adaptive explicit FMO map for the purpose of reducing packet loss rate at queues of 802.11e MAC. We show that in the work using data partitioning (DP) [3] or the work without the use of adapting FMO, when transmitting video packets over wireless network supporting QoS, the capacity of queues at MAC layer are not used effectively. The high priority queues are sometimes empty; meanwhile the low priority queues are always in overload. This effectiveness results in unnecessary packet dropping at the low priority. To minimize the number of dropped packets, we propose a method generating explicit FMO map based on feedback information from queues at MAC layer. Based on overflow state at queues, the order of encoding high and low important slice groups in the current frame is decided. By the adjustment FMO map frame by frame, the arriving order of packets to queues is changed. Consequently, the arrival rate of packets to queues is changed to reduce fullness at overflow queues. It results in packet dropping rates at queue are decreased and PSNR of proposed method is increased comparing to some previous methods.

The organization of this chapter is as follows. Section 4.1 is literature review of some previous works. In this section, we discuss the disadvantages of previous works and suggest the method to overcome. In Section 4.2, the method to estimate overflow state of queues at MAC layer is introduced. In Section 4.3, our proposed method to generate adaptive FMO map is presented. The comparisons of proposed method to the previous methods are described in Section 4.4. The results show that our proposed scheme is effective in reducing the packet loss rate while improving the average PSNR compared to the previous method. The conclusion is presented in Section 4.5.

4.1 Introduction

Multimedia applications such as video streaming and telephony are becoming an important part of the network user experience and expectation. As a result, it is important to support such applications in widespread broadband access networks such as 802.11 WLAN [36]. However, in this network, the resource management, adaptation, and protection strategies available in the lower layers of the OSI stack are optimized without explicitly considering the specific characteristics of the multimedia applications. Conversely, multimedia compression and streaming algorithms do not consider the mechanisms provided by the lower layers for error protection, scheduling, resource management and so on [37]. Therefore, it is necessary to control the parameters and operation of each layer in conjunction with the others to achieve optimum performance and avoid service degradation.

There are many researches focus on the interaction between Application and MAC layer with the objective to reduce the packet loss rate [3], [4]. In [3], a method of marking priority for each type of video packet is proposed. Based on data partitioning scheme of H.264, a slice is divided into three parts. Partition A contains the most important information which is crucial to the decoding of the other partitions. Partition B (intra partition) carries intra coded block pattern and intra coefficient. This partition is more important than Partition C because this information can stop further drift. Partition C (inter partition) carries only inter coded block pattern and inter coefficient. In addition to three partitions, H.264 bitstream includes the parameter set concept (PSC) and instantaneous decoding refresh (IDR). PSC contains information such as picture size, display window, optional coding modes employed, MB allocation map. IDR contains only the intra picture where no data partitioning can be applied. At the NAL layer of H.264, each NALU is considered as a packet that contains payload type, which is partition A, B or C of slice. In addition, NALU includes an NAL header having `Nal_Ref_Idx` (NRI) field. The NRI contains two bits that indicate the priority of the NALU payload, where 11 is the highest transport priority, followed by 10, then by 01, and finally, 00 is the lowest. Accordingly, information of partition A obtains the highest priority and

information of partition C obtains the lowest priority. At the IEEE 802.11e MAC layer, there are four classified queues: AC 3 corresponds to the highest priority, and AC 0 is the lowest. The PSC is mapped to the highest priority queue. The IDR and partition A are mapped to the same category AC2 while partition B and C are mapped to AC1. AC 0 is used for best effort traffic. The architecture of system is shown in Fig. 4.1. Although this method achieves better performances in terms of delays and loss rate than the actual WLAN standard and its QoS enhancement mechanism, there exists some limitations. Because the number of packets of partition A is small while that of partition B and C is large. Thus, the capacity of queue AC2 may be in available while the queue AC1 is overload. Consequently, it causes the unnecessary packet dropping in the queue AC1. In these cases, it is better if some packets arriving AC1 can be mapped to AC2 to avoid overload in AC1.

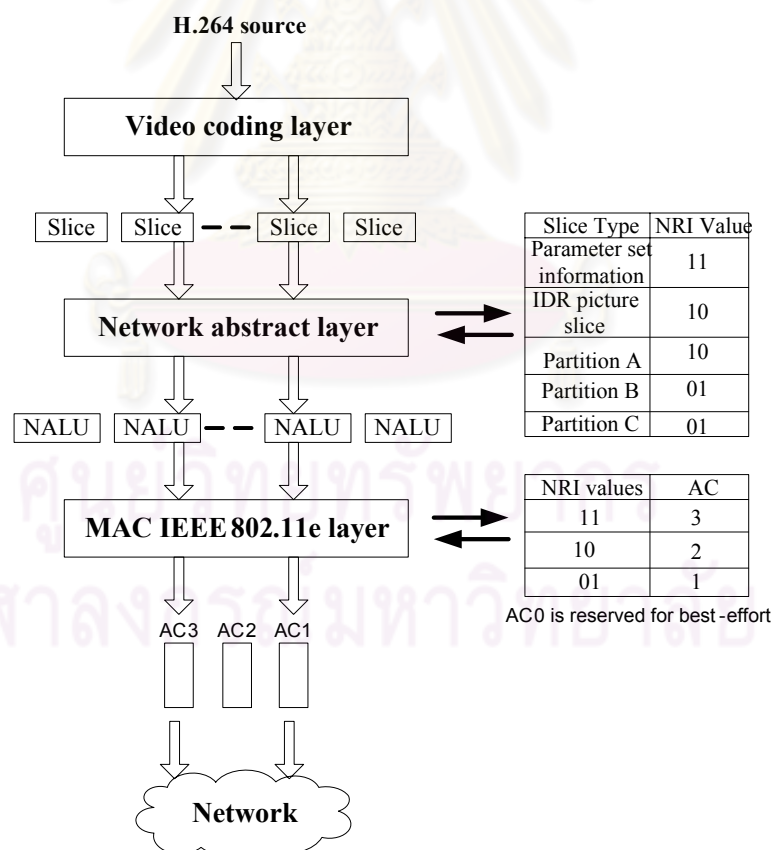


Figure 4.1: Cross layer QoS architecture

To overcome the issue mentioned above, [4] proposes a method to balance traffics

coming to queues. In this method, packets of video data are mapped into AC2, best effort traffics are mapped to AC1 and AC0 is spent for back ground traffic. At the MAC layer, the first queue length of AC2 is checked and compared against a set of threshold values. If the queue length is lower than the lower threshold, the video data is mapped to AC2. But if the queue length is greater than the upper threshold, the video data is directly mapped to lower priority queues AC1 and AC0.

In scalable video coding, video bit stream is organized into separate layers based on the delay deadlines of the various video frames (e.g. I-, P- and B-frames). Based on cross-layer application-MAC layer architecture, [38], [23], [24] uses queue theory to optimize the number of retransmission and packet sizes for each video layers with different QoS requirements. In particular, the network queue and the MAC layer monitor the overflow rate and the packet error rate. Using the information about state of queues, a suitable number of retransmissions are given to achieve the lowest packet loss rate. Meanwhile, authors prove that the packet loss rate also depends on the packet size. Thus, the goal of the cross-layer optimization is to determine the optimal packet size and maximum number of times each packet can be transmitted such that the expected video distortion is minimized under a give delay constraints.

In [39]- [40], different FMO types are conducted to find out which pattern provide the best video quality for a given packet loss scenario, typical for WLAN environments. The results show that the "dispersed" FMO type provides the best PSNR for the case of moderate packet loss. However, with the best PSNR, this method just shows that "dispersed" FMO is good for error concealment at decoder, not for cross-layer optimization between layers at encoder. In other word, the FMO type at application layer does not affect to the lower layer in term of reducing packet loss rate. Therefore, in our proposed method, based on information from MAC layer, FMO at application is created with criterion of reducing the packet loss rate at queues of MAC layer.

4.2 Overflow rate

At the MAC layer, packet losses occur due to two reasons: link erasures and queue overload. In the scope of the thesis, we assume that link erasure is zero. The used queues are drop tail queue. Thus, the packet drop rate at the queues depends on the arrival rate and service rate of queues. If the arrival rate is greater than the service rate, the queue is occupied quickly by waiting packets. If this state occurs for a long time, queue is felt into full state and the arrival packets are dropped. This state is overflow state of the queue.

In this work, we use a simplified buffer analysis based on fluid model. Let L_r be the link retry limit, and P_e be the packet error rate (PER) of the link (without retry), then the mean number of transmissions for a single packet until it is either successfully received or it reached its retry limit can be calculated as [41]:

$$\begin{aligned} s(L_r, P_e) &= 1(1 - P_e) + 2P_e(1 - P_e) + \dots + L_r P_e^{L_r-1}(1 - P_e) + (L_r + 1)P_e^{L_r} \\ &= \frac{1 - P_e^{L_r+1}}{1 - P_e} \end{aligned} \quad (4.1)$$

Let λ be the arrival rate (packets/s). In the fluid model, we calculate the overflow rate as

$$\delta(L_r, P_e) = \frac{s(L_r, P_e)\lambda - C}{s(L_r, P_e)\lambda} \quad (4.2)$$

, where C is the service rate of the link (packets/s). Equation 4.2 shows that overflow occurs only when $s\lambda > C$. By substituting Eq. 4.1 into Eq. 4.2, we have

$$\delta(L_r, P_e) = 1 - \frac{1}{\rho(P_e)} \frac{1}{1 - P_e^{L_r+1}} \quad (4.3)$$

, where $\rho(P_e) = \frac{\lambda}{C(1 - P_e)}$ is the effective utilization factor of the link.

4.3 Cross-layer MAC-Application-layer using Adaptive FMO and Queuing Overflow Rate

For simplification, we assume that there are two queues in MAC layer: AC1 and AC2. The priority of AC2 is higher than the priority of AC1. Assuming that L_r and P_e are constant, from Eq. 4.2 and 4.3, we can see that $\rho(L_r, P_e) \sim \frac{\lambda}{C}$. Therefore, to reduce the overflow rate of a queue, we should reduce the arrival rate of that queue.

In video coding transmission without FMO, MBs in a frame are encoded in raster scan order. Meanwhile, the distribution of important MBs in a frame is almost random. Figure 4.2 shows the importance of MBs in frame 10 of "akiyo" sequence. Numbers in the table show the importance of MBs in the frame. In this case, the importance is measured by the residual of MBs. If MB's residual error is equal to zero, MB is considered "not important" and those MBs are implied to belong to background region of a frame. More important MBs belong to the region of interesting (ROI) of a frame. We can see that, the order of more important MBs and less important MBs are interlaced. Thus, the arriving rate of packets to the high and low priority queue is not able to be controlled.

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	4617	9367	4797	0	0	0	0
0	0	0	0	12687	17697	7717	0	0	0	0
0	0	0	17	30567	22537	11207	2197	0	0	0
0	0	0	0	7897	22387	6847	3477	0	0	0
0	0	0	1797	5107	20237	15957	6787	7007	517	0
0	37	817	47	2297	11867	8127	4387	1637	4717	0
0	0	2507	3017	1577	10927	5107	2677	2827	8007	0
0	1247	2507	67	67	6107	7527	4027	6097	1477	0

Figure 4.2: Importance of MBs in frame 10 of "akiyo" sequence

To adjust the order of arrived packets, we can adjust the order of encoding MBs in a frame by using explicit FMO map. Figure 4.3 shows the explicit FMO map and Fig. 4.4 is the importance of MBs arranged according to the FMO map appropriately. According to this FMO map, more important MBs (MBs in bold) are encoded first. After that, the less important MBs (white MBs) are encoded. Therefore, the more impor-

461	471	491	371	391	371	271	291	271	801	841
871	501	541	531	571	601	641	631	621	681	651
691	671	701	741	731	721	781	751	761	901	941
931	921	981	951	961	971	701	741	731	721	781
751	761	771	791	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15	19	20	21
22	23	24	25	26	30	31	32	33	34	35
36	42	43	44	45	46	47	48	53	54	55
56	57	58	66	67	77	78	79	88	89	99

Figure 4.3: Explicit FMO map of frame 10 of "akiyo" sequence

4617	9367	4797	12687	17697	7717	17	30567	22537	11207	2197
7897	22387	6847	3477	1797	5107	20237	15957	6787	7007	517
37	817	47	2297	11867	8127	4387	1637	4717	2507	3017
1577	10927	5107	2677	2827	8007	1247	2507	67	67	6107
7527	4027	6097	1477	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Figure 4.4: The importance of MBs after rearranging in frame 10 of "akiyo" sequence

tant packets will be mapped into high priority queue first. And then, the less important packets are mapped into low priority queue. If we want flow of less important packets arrives to low priority queue first, the FMO map is converted. The less important MBs are encoded first, followed by the more important MBs.

Figure 4.5 describes the architecture of system using cross-layer scheme to generate explicit FMO map. In this system, after encoding, flow of video packets is mapped into queue AC1 and queue AC2. After encoding each frame, the overflow state of two queues is sent to encoder. Based on this information, encode decides a FMO map for the next frame to adapt with states of queues.

In case of overflow rate of queue AC1 is higher than that of queue AC2, $\sigma_1 \geq \sigma_2$, to reduce the arrival rate of packets coming to queue AC1, FMO map is generated in the way such that the higher important MBs are encoded before the lower important MBs.

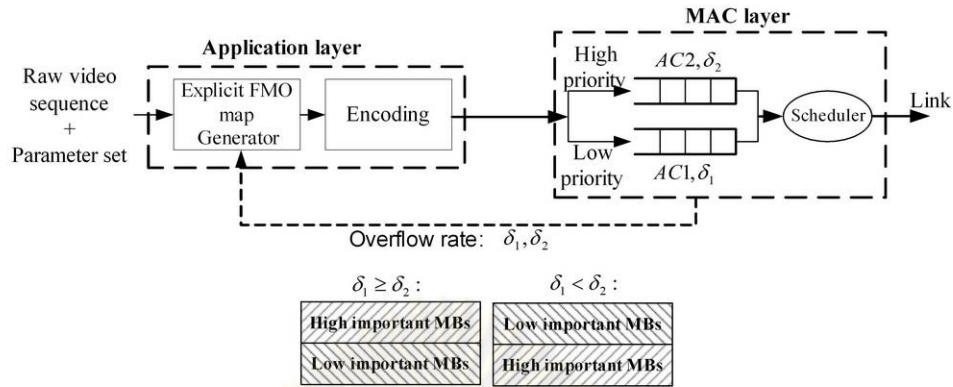


Figure 4.5: Cross-layer architecture of the proposed method

Therefore, the lower priority packets will come to queue AC1 later than the higher priority packets coming to queue AC2. In other word, the arrival rate λ_1 is decreased and arrival rate λ_2 is increased. Consequently, σ_1 is decreased and σ_2 is increased.

In case of $\sigma_2 > \sigma_1$, to reduce the arrival rate of packets to queue AC2, FMO map is converted. Thus the lower important MBs are encoded first. And thus the higher priority packets will come to queue AC2 later than the lower priority packets coming to queue AC1. Hence, in the encoding period of a first half frame, λ_2 is decreased and λ_1 is increased. Consequently, σ_2 is decreased and σ_1 is increased.

4.4 Experiment Results

4.4.1 Simulation setup

In this work, a frame is divided into 8 slice groups. The higher important MBs are contained in four slice groups and the lower important MBs are contained in the other four slice groups. Each slice group is contained in a packet. As the result, there are two types of packets: the high priority packets containing the higher important slice groups and the low priority packets containing the lower important slice groups as shown in Fig. 4.4. The high priority packet is mapped into AC2 and the other is mapped into AC1. The parameters of queues are shown in Table 4.1.

For the simulations, four video sequences Akiyo, Coastguard, Foreman and High-

Table 4.1: Access Category at the MAC layer

Type	Access Category	Value
Voice IP	AC3 (CW _{min} ,CW _{max} ,AIFS,RL)	(7,15,2,4)
Video	AC2 (CW _{min} ,CW _{max} ,AIFS,RL)	(15,31,2,4)
	AC1 (CW _{min} ,CW _{max} ,AIFS,RL)	(31,1023,3,4)
CBR,FTP	AC0 (CW _{min} ,CW _{max} ,AIFS,RL)	(31,1023,7,4)

Table 4.2: The number of video packets and packet size of each method

Method	Frame number	Packet number	Average video packet size
Proposed method (Adaptive FMO)	100	802	80 bit (64 kbps), 145 bit (128 kbps), 410 bit (384 kbps)
Non-Adaptive FMO	100	802	80 bit (64 kbps), 145 bit (128 kbps), 410 bit (384 kbps)
Data Partition	100	867 (64 kbps), 881 (128 kbps), 887 (384 kbps)	74 bit (64 kbps), 133 bit (128 kbps), 372 bit (384 kbps)

way are used. These sequences are coded at 20 fps with bit rate of 64 kbps, 128 kbps and 384 kbps. To examine the efficiency of the cross layer mechanism, we conduct experiments over an 802.11e wireless LAN and implemented in network simulator (NS2) [42] and [43]. In order to evaluate the advantage of the proposed method, the proposed method is compared with two other methods in term of PSNR and packet loss rate. The first method uses DP and the second method uses non-adaptive FMO. In the method using DP, packets of partition A is mapped into queue AC2 while partitions B and C are mapped into queue AC1. In the method using non-adaptive FMO, a frame is also divided into 8 slice groups including 4 higher important slice groups and 4 lower important slice groups. However, without adaptability, high important slice groups are always encoded first. As the result, the arrival rates of packets to queues are considered as constant. Table 4.2 shows parameters of each method.

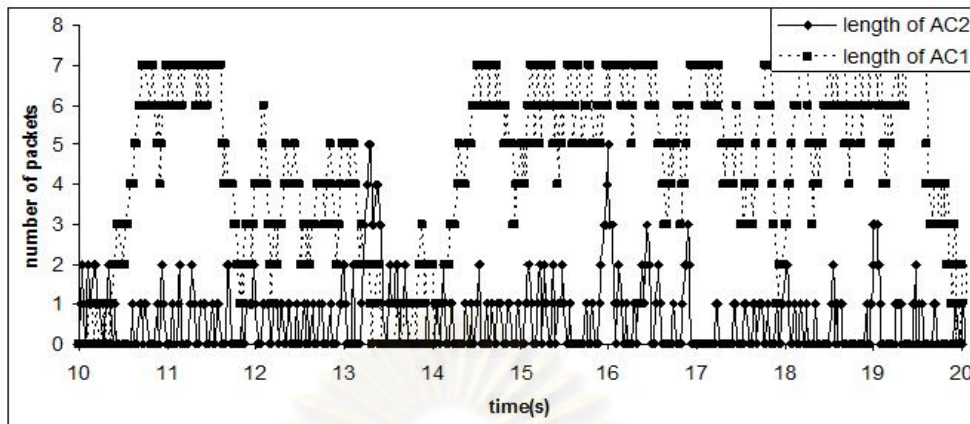


Figure 4.6: Queue length of the method using Data Partition with "coastguard" sequence at 64kbps

4.4.2 Result analysis

a. Comparisons in queue length

Figure 4.6 describes the length of queues when transmitting partition B and partition C. From these measurements, it appears that AC1 (partition C) is always in full state. While state of AC2 is not used effectively. This unbalance causes unnecessary in packet dropping of AC1.

Similarly, in method using FMO without adaptability (Fig. 4.7), the arrival rates of packets to two queues are equal. However, the serve rate of AC2 is greater than serve rate of AC1. Thus, the state of AC1 is always in full while the AC2 is occupied by a small number of packets.

Result in Fig. 4.8 shows that in the proposed method, because of the change of arrival traffic depending on overflow state of queue, the fullness of AC1 (for high important packets) queue is reduced significantly. This is because when AC1 having signal of overflow, the traffic of packets coming to queue is relayed to AC2. Hence, the packet drop rate of AC3 queue is reduced in the proposed method. Because of sharing between two queues, the average length of AC2 in the proposed method is increased. However, this increment is not significant therefore the drop rate of AC2 is not affected.

b. PSNR and Drop rate comparison

From Table 4.3-4.5, we can see that average PSNR of method using FMO without

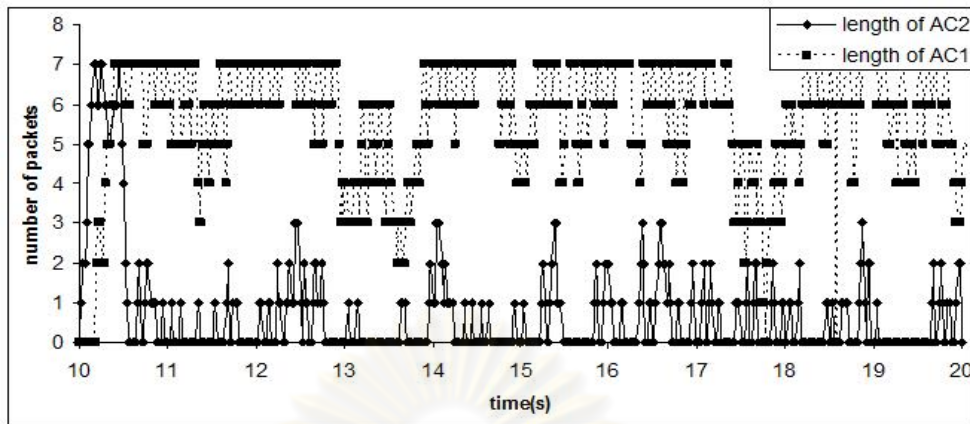


Figure 4.7: Queue length of the method using None-Adaptive FMO with "coastguard" sequence at 64 kbps

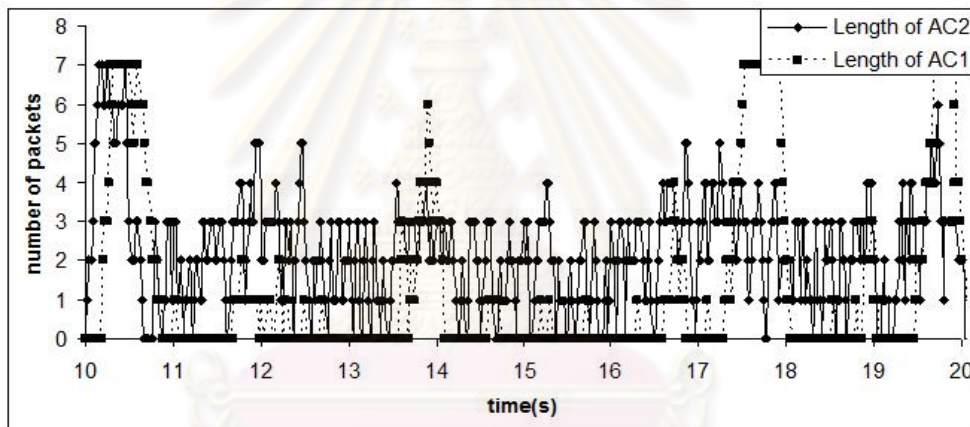


Figure 4.8: Queue length of the proposed method using adaptive FMO with "coastguard" sequence at 64 kbps

adaptability is the lowest; the following is the method using DP. The proposed method with adaptive FMO has the highest average PSNR. The improvement is up to about 6dB compared to the method using FMO without adaptability and up to about 4dB compared to the method using DP. These improvements are explained by packet drop rate at queues shown in Table 4.6-4.11:

- The drop rates at both queues of method using FMO without adaptability are always the highest. Because the number of low and high priority packets are almost the same. In addition, the arrival rate of these packets to queues is con-

stant. Therefore, the drop rates at both queues are high, especially at the queue AC1. For example, as shown in Table 4.7 and 4.8, the drop rate of "akiyo" sequence with 64kbps at AC2 is 0.034 (greater than 0.0 of the method using DP and 0.014 of the proposed method) meanwhile at AC1 is 0.25 (greater than 0.24 of the method using DP and 0.02 of the proposed method). Consequently, the average PSNR of this method is the lowest.

- The drop rate at AC2 of method using DP is the lowest because the number of packets in partition B is very small. However, the drop rate at AC1 of this method is higher than that of the proposed method. Because in a slice groups, the partition C is major. Thus the number of packets in partition C is larger than that of partition A and B. Because of unbalance in using queue capacity, the drop rate in AC1 is very high comparing to the drop rate at AC2. For example, drop rate at AC1 of the method using DP in "akiyo" sequence with 64kbps is 0.32 (much greater than 0.0 at AC2 of this method and 0.02 at AC1 of the proposed method). However, compared to drop rate at AC1 of the method using FMO without adaptability, the drop rate at AC1 of method using DP is still smaller. Furthermore, the important packets at AC3 and AC3 are preserved. Thus, the average PSNR of this method is higher than that of the method using FMO without adaptability.
- Meanwhile, the drop rates at both queues of the proposed method are rather fair and small. For example, the drop rates at both queues of "akiyo" sequence with 64kbps are 0.014 and 0.02, appropriately. It shows that the balance in dispatching traffics to queues is important.

Table 4.10-4.12 show the number of undecodable MBs in each method. Undecodable MB is MB which can not be received by decoder because the packet containing this MB is dropped at MAC layer of transmitter. To reconstruct these lost MBs, the collocated MBs of the previous frame are copied. The results show that:

- The higher bitrate is, the larger number of undecodable MBs is. This is because of the packet size in high bitrate is larger than packet size in low bitrate. Thus,

the queues in the case of high bitrate are full more quickly than in the case of low bitrate.

- Because of low dropping rate, the proposed method has the lowest number of undecodable MBs.

Figure 4.9 and 4.10 shows subjective evaluations of the video quality for "akiyo" and "coastguard" sequence of our proposed method compared to the other methods.



Table 4.3: Average PSNR comparison of three methods at **64 kbps**

	Data partition	None-Adaptive FMO	Adaptive FMO
akiyo	38.52	37.38	42.60
coastguard	25.15	25.95	28.49
foreman	32.30	27.38	40.33
highway	38.99	30.88	36.72

Table 4.4: Average PSNR comparison of three methods at **128 kbps**

	Data partition	None-Adaptive FMO	Adaptive FMO
akiyo	38.20	35.20	43.81
coastguard	22.25	24.86	27.49
foreman	24.50	27.59	29.56
highway	37.74	31.28	35.13

Table 4.5: Average PSNR comparison of three methods at **384 kbps**

	Data partition	None-Adaptive FMO	Adaptive FMO
akiyo	36.20	33.32	36.78
coastguard	21.27	22.19	22.45
foreman	29.16	25.29	25.81
highway	31.16	29.11	29.64

Table 4.6: Drop rate comparison at AC2 queue of three methods with **64 kbps**

	Data partition	None-Adaptive FMO	Adaptive FMO
akiyo	0.0	0.034	0.014
coastguard	0.0	0.027	0.017
foreman	0.003	0.017	0.0
highway	0.0	0.027	0.005

Table 4.7: Drop rate comparison at AC1 queue of three methods with **64 kbps**

	Data partition	None-Adaptive FMO	Adaptive FMO
akiyo	0.24	0.25	0.02
coastguard	0.1	0.22	0.03
foreman	0.2	0.28	0.01
highway	0.18	0.3	0.02

Table 4.8: Drop rate comparison at AC2 queue of three methods with **128 kbps**

	Data partition	None-Adaptive FMO	Adaptive FMO
akiyo	0.000	0.041	0.004
coastguard	0.012	0.036	0.014
foreman	0.005	0.032	0.007
highway	0.000	0.029	0.024

Table 4.9: Drop rate comparison at AC1 queue of three methods with **128 kbps**

	Data partition	None-Adaptive FMO	Adaptive FMO
akiyo	0.239	0.323	0.025
coastguard	0.178	0.285	0.03
foreman	0.212	0.287	0.017
highway	0.206	0.242	0.078

Table 4.10: Drop rate comparison at AC2 queue of three methods with **384 kbps**

	Data partition	None-Adaptive FMO	Adaptive FMO
akiyo	0.0	0.044	0.027
coastguard	0.005	0.07	0.07
foreman	0.002	0.084	0.074
highway	0.012	0.057	0.078

Table 4.11: Drop rate comparison at AC1 queue of three methods with **384 kbps**

	Data partition	None-Adaptive FMO	Adaptive FMO
akiyo	0.32	0.36	0.18
coastguard	0.48	0.45	0.3
foreman	0.31	0.46	0.27
highway	0.17	0.4	0.27

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Table 4.12: Number of undecodable MBs of three methods with **64 kbps**

	Data partition	None-Adaptive FMO	Adaptive FMO
akiyo	603	861	214
coastguard	452	831	138
foreman	669	949	39
highway	703	903	89

Table 4.13: Number of undecodable MBs of three methods with **128 kbps**

	Data partition	None-Adaptive FMO	Adaptive FMO
akiyo	769	1119	114
coastguard	1264	993	103
foreman	1339	978	51
highway	1063	850	306

Table 4.14: Number of undecodable MBs of three methods with **384 kbps**

	Data partition	None-Adaptive FMO	Adaptive FMO
akiyo	1397	1341	742
coastguard	4714	1410	1051
foreman	3349	1441	1126
highway	2112	1409	928

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



Figure 4.9: Subjective evaluation for the "Akiyo" sequence frame 50

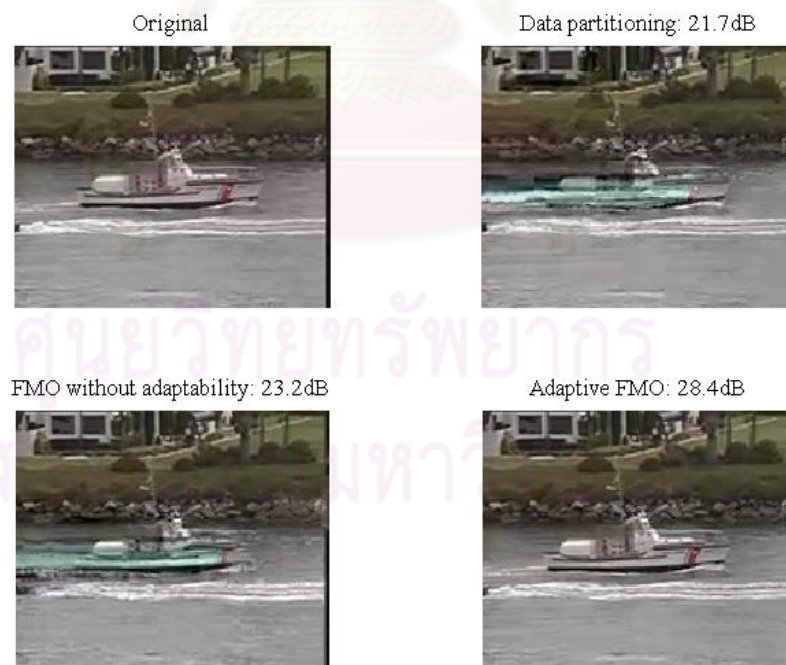


Figure 4.10: Subjective evaluation for the "Coastguard" sequence frame 60

CHAPTER V

Conclusions and Future Works

In this dissertation, we present a new method for error resilient video coding by using adaptive FMO when transmitting video over error prone environment like wireless channel.

In chapter III, the three-state Markov model is used to estimate the locations of burst and guard sections in channel. In addition, the importance of MB is measured based on effect of error propagation. With the predicted information of channel, important MBs are arranged into guard sections and unimportant MBs are arranged into burst sections. Moreover, when considering inter-frame error propagation, a suitable intra refresh rate is selected based on the channel state to reduce the effect of error propagation. Experimental results show that our proposed method gains some improvements in terms of PSNR as compared to some conventional methods that have not taken channel condition and error propagation into consideration in generating FMO map.

Another method using adaptive FMO map is proposed in chapter IV. In this chapter, the ineffectiveness in using queues at MAC layer of some previous method is analyzed by using dropping rate. In these methods, video packets are classified and mapped into queues depending on importance of packets. However, the inflexibility in mapping algorithm results in the unbalance in using queues. At sometimes, higher priority queues is available while the other queue is full. From this limitation, a new method using cross-layer approach is proposed to reduce dropping packet rate, based on feedback information from queues at MAC layer of WLAN 802.11e standard, encoder changes FMO map in such that the arrival rate of packets is changed following the overflow rate of queues. In particular, video packets are also classified into two types of priority and mapped into two queues at MAC layer. However, the overflow rate of queues are tracked by encoder. If overflow rate of a queue is high, the arrival rate of packets to

that queue is adjusted by changing FMO map. Therefore, the arrival rate of packets to queues are adaptive with fullness of queues. The proposed method is compared to the method using data partitioning and the method using FMO without adaptivity. The results show that the proposed method is effective in reducing the packet dropping rate and can improve the PSNR up to 5 dB. Subjective evaluations also show an overall improvement using our proposed method.

The proposed methods using adaptive FMO in this dissertation improve quality of video image significantly comparing with some other methods for error resilient in video coding. However, there are still many research directions in error resilient in wireless video transmission. Possible further works include improving the wireless channel model to better adaptively cope with varying channel condition, improving algorithm for generating FMO map to further improve the performance. In addition, error concealment scheme at decoder should be considered to increase error-proof ability of system.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

REFERENCES

- [1] W. Hantanong, and S. Aramvith. Analysis of macroblock-to-slice group mapping for h.264 video transmission over packet-based wireless fading channel., *48th Midwest Symposium on Ckt and System 2* (Aug. 2005): 1541-1544.
- [2] J. Li, and K. Ngi Ngan. An Error Sensitively-based Redundant Macroblock Strategy for Robust Wireless Video Transmission. *Int'l Conference on Wireless Networks, Communications and Mobile Computing 2* (June 2005): 1118 - 1123.
- [3] A. Ksentini, M. Naimi, and A. Gueroui. Toward an Improvement of H.264 Video Transmission over IEEE. 802.11e through a Cross-layer Architecture. *IEEE Communications Magazine* 44 (2006): 107-114.
- [4] C.-H. Lin, C.-K. Shieh, C.-H. Ke, N. K. Chilamkurti and S. Zeadally. An Adaptive Cross-Layer Mapping Algorithm for MPEG-4 Video Transmission over IEEE 802.11e EDCA. *Springer Netherlands* 42 (Dec. 2009): 223-234.
- [5] ITU. *Advanced Video Coding for generic audiovisual services*. ITU-T Recommendation H264. March 2005.
- [6] S. Wenger, and M. Horowitz. Fmo: Flexible macroblock ordering., *Fairfax (USA)*. May 2002.
- [7] R.D.Cajote, S. Aramvith, R.C.L. Guevara, and Y. Miyanaga. Fmo slice group maps using spatial and temporal indicators for h.264 wireless video transmission. *Circuits and Systems. ISCAS-2008*. (2008): 3566 - 3569.
- [8] T. H. Vu, and S. Aramvith. An error resilience technique based on fmo and error propagation for h.264 video coding in error-prone channels. *Proc. of the 2009 IEEE international conference on Multimedia and Expo*. (2009): 205-208.

- [9] S. Im, and A.J. Pearmain. Unequal error protection with H.264 flexible macroblock ordering. *Proc. of SPIE, Visual Communications and Image Processing*.
- [10] ITU. *RFC: 3984 RTP payload*. ITU-T Recommendation H264. March 2005.
- [11] T. S. Rappaport. *Wireless Communications Principles and Practice*. New-York: Prentice Hall. 1996.
- [12] S. R. Saunders. *Antennas and Propagation for Wireless Communication Systems*. New-York: John Wiley and Sons, 1999.
- [13] M. Zorzi, R. R. Rao, and L. B. Milstein. ARQ error control for fading mobile channels. *IEEE Trans. On Veh. Tech.* 46 (May 1997): 445-455.
- [14] Y. S. Baguda, N. Fisal, S. H. Syed, S. K. Yusof, M. S. Abdullah, A. Mohd and A. Zuljarmawan. Mobile streaming of H.264 video over Gilbert-Elliott's Channel. *World Academy of Science, Engineering and Technology* 48 (Dec. 2008): 344-347.
- [15] S. Aramvith, I. -M. Pao, M. -T. Sun. A Rate-Control Scheme for Video Transport over Wireless Channels. *IEEE Trans. on Circuits and System for Video Technology* 1 (May 2001): 569-580.
- [16] K. Stuhlmiller, N. Frber, M. Link, and B. Girod. Analysis of Video Transmission over Lossy Channels. *IEEE Journal on Selected in Communications* 18 (June 2000): 1012-1032.
- [17] H. S. Wang, and N. Moayer. Finite-state Markov channel-a usefull model for radio communication channels. *IEEE Trans. on Vehicular Technology* 44 (Feb. 1995): 163-171.
- [18] B. D. Fritchman. A binary channel characterization using partitioned Markov chains. *IEEE Trans. Inform. Theory* 13 (Apr. 1967): 221-227.

- [19] T. Sato, K. Tokuda, M. Kawabe, and T. Kato. Simulation of burst error models and adaptive error control scheme for high speed data transmission over analog cellular system. *IEEE Trans. Veh. Technol.* VT-40 (May 1991): 443-452.
- [20] T. Sato, M. Kawabe, T. Kato and A. Fukasawa. Throughput Analysis Method for Hybrid ARQ Schemes Over Burst Error Channels. *IEEE Trans. Veh. Technol.* 40 (Feb. 1994): 110-118.
- [21] W. C. Jakes, and Jr. Editor. *Microwave Mobile Communications*. John Wiley and Sons, 1974.
- [22] T. C. Chen, L.F. Chang, A.H. Wong, M.T. Sun, and T.R. Hsing. A real-time software based end-to-end wireless visual communications simulation platform. *Proceedings SPIE Visual Communications and Image Processing'95* 3 (May 1996): 1068-1074.
- [23] M. van der Schaar and S. Shankar. Cross-Layer Wireless Multimedia Transmission: Challenges, Principles and New Paradigms. *IEEE Wireless Communication Magazine* 12 (Aug. 2005): 50-58.
- [24] M. van der Schaar and D. Turaga. Cross-Layer Packetization and Retransmission Strategies for Delay-Sensitive Wireless Multimedia Transmission. *IEEE Trans. on Multimedia* 9 (2007): 158-197.
- [25] M. van der Schaar, D. Turaga, and R.S Wong. Classification-Based System for Cross-Layer Optimized Wireless Video Transmission. *IEEE Trans. on Multimedia* 8 (2006): 1082-1095.
- [26] M. van der Schaar and S. Shankar. New Fairness Paradigms for Wireless Multimedia Communication. *Pro. IEEE ICIP* 3 (Sept. 2005): 704-707.
- [27] Yao Wang, Jörn Ostermann and YaQuin Zhang. *Video Processing and Communications*. Prentice Hall, 2002.

- [28] B. Girod, and N. Farber. *Error-resilient standard compliant video coding* [Online]. 1998. Available from: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.6841>.
- [29] G. Reyes, A. Relbman and S. F. Chang. A corruption model for motion compensated video subject to bit error. *Proc. Packet Video Workshop*.
- [30] Chih-Ming Chen, Chia-Wen Lin, Hsiao-Cheng Wei, Yung-Chang Chen. Robust video streaming over wireless LANs using multiple description transcoding and prioritized retransmission. *Journal of Visual Communication and Image Representation* 8 (2007): 191-206.
- [31] Z. He, J. Cai and C. W. Chen. Joint Source Channel Rate-Distortion Analysis for Adaptive Mode Selection and Rate Control in Wireless Video Coding. *IEEE Trans. On Circuit and Systems for Video Tech* 12 (2002): 511-523.
- [32] R. Zhang, S. L. Regunathan, and K. Rose. Video coding with optimal inter/intra-mode switching for packet loss resilience. *IEEE Journal on Selected Area in Comm.* 18 (Jun. 2000): 966-976.
- [33] Jin Xu, and Zhimei Wu. Joint Adaptive Intra Refreshment and Unequally Error Protection Algorithms for Robust Transmission of H.264/AVC Video. *Multimedia and Expo, 2006 IEEE International Conference*. (July 2000): 693-696.
- [34] J. Y. Shih, and W. J. Tsai. A New Unequal Error Protection Scheme Based on FMO. *ICIP 2008, 15th IEEE Int'l Conference*. (Oct. 2008): 3068-3071.
- [35] Karsten Sühning. *H.264 JM Ref. Software* [Online]. 2010. Available from: <http://iphome.hhi.de/suehring/tml/download/>.
- [36] IEEE 802.11e. Wireless LAN Medium Access Control (MAC) Enhancements for Quality of Service (QoS). *802.11e draft 8.0*.

- [37] Y. L. B. Girod, M. Kalman and R. Zhang. Advances in channle-adaptive video streaming. *Wireless Communications and Mobile Computing* 2 (Sept. 2002): 549-552.
- [38] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu. Adaptive Cross-Layer Protection Strategies for Robust Scalable Video Transmission over 802.11 WLANs. *IEEE Journal on Selected Areas of Communications (JSAC)* 32 (Dec. 2003): 1752-1763.
- [39] Y. P. Fallah, D. Koskinen, A. Shahabi, F. Karim and P. Nasiopoulos. A Cross Layer Optimization Mechanism to Improve H.264 Video Transmission over WLANs. *proc. of Consumer, Communications and Networking Conference (CCNC)*. (Jan. 2007): 875-879.
- [40] S. Benayoune, N. Achir, K. Boussetta and K. Chen. A MAC centric Cross Layer approach for H.264 video streaming over HSDPA. *Journal of Communications* 4 (Oct. 2009): 691-699.
- [41] Q. Li and M. van der Schaar. Providing Adaptive QoS to Layered Video over Wireless Local Area Networks through Real-time Retry Limit Adaptation. *IEEE Trans. on Multimedia* 6 (Apr. 2004): 278-290.
- [42] *The network-simulator ns-2* [Online]. 1989. Available from: <http://www.isi.edu/nsnam/ns/>.
- [43] Cheng-Han Lin and Chih-Heng. *Evaluation of video stream quality over IEEE 802.11e EDCF*. [Online]. 2009. Available from: <http://140.116.72.80/~smallko/ns2/ns2.htm>.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

APPENDIX A

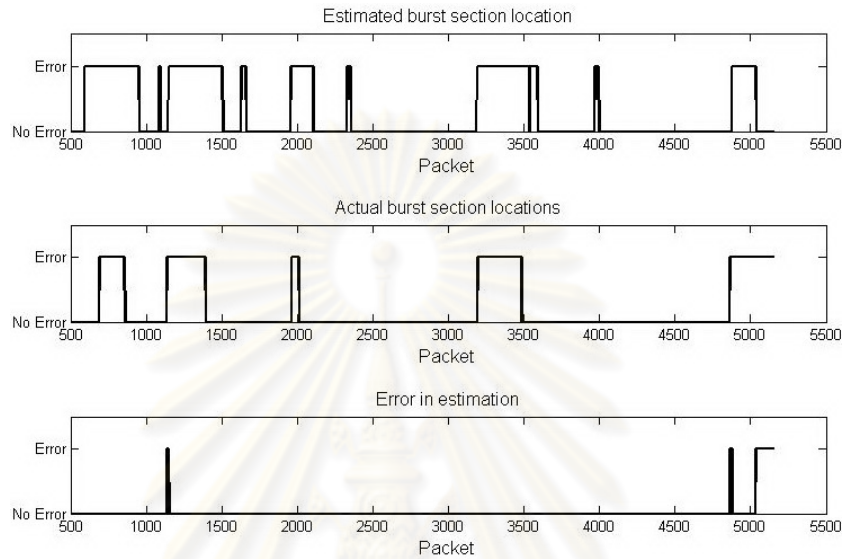


Figure A.1: Comparison actual and estimated channel state for "Akiyo" sequence in slow fading, 64 kbps.

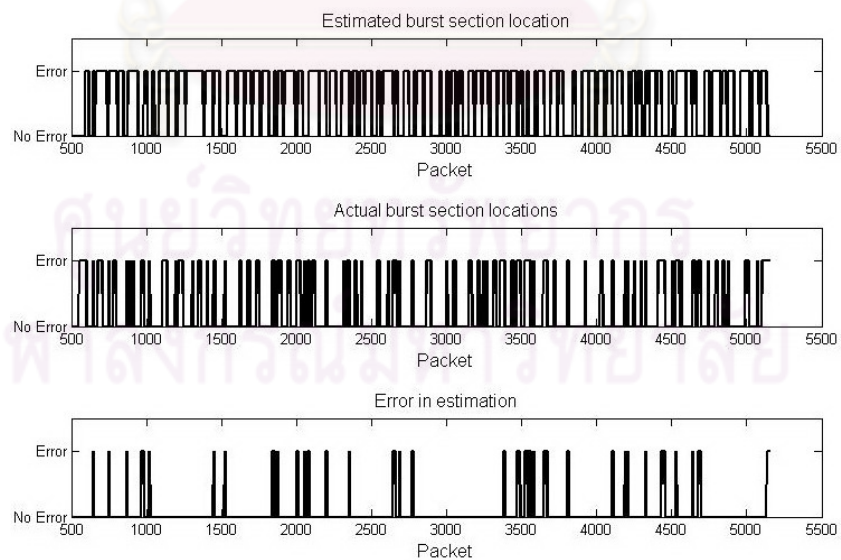


Figure A.2: Comparison actual and estimated channel state for "Akiyo" sequence in fast fading, 64 kbps.

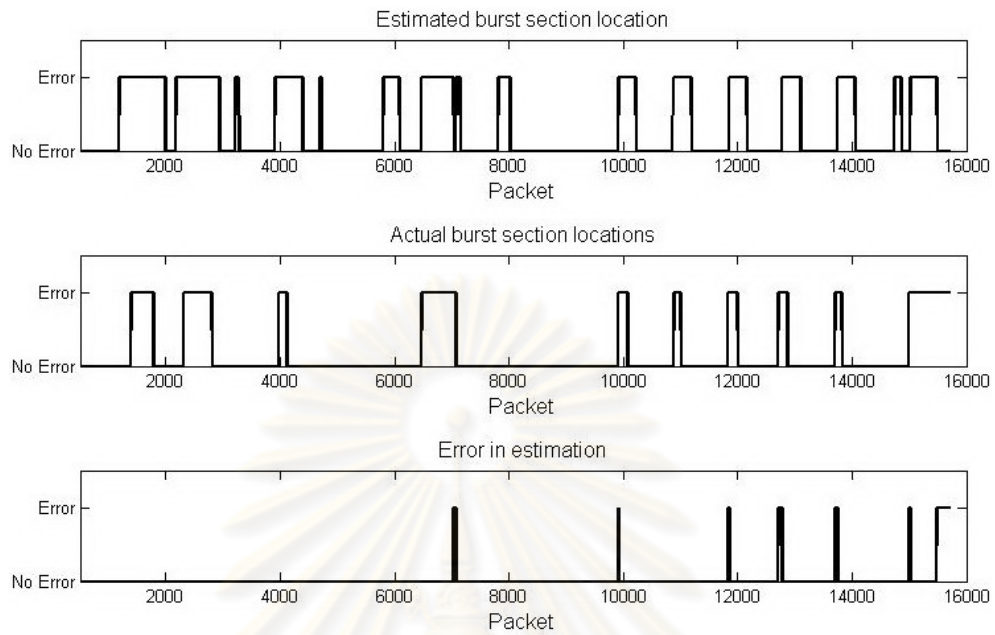


Figure A.3: Comparison actual and estimated channel state for "Akiyo" sequence in slow fading, 128 kbps.

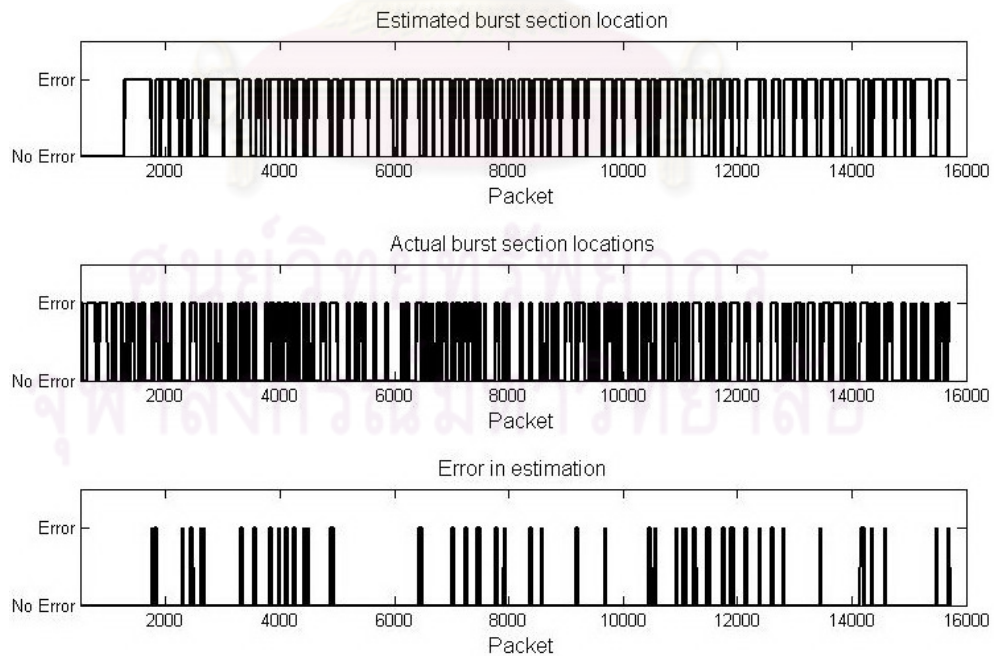


Figure A.4: Comparison actual and estimated channel state for "Akiyo" sequence in fast fading, 128 kbps.

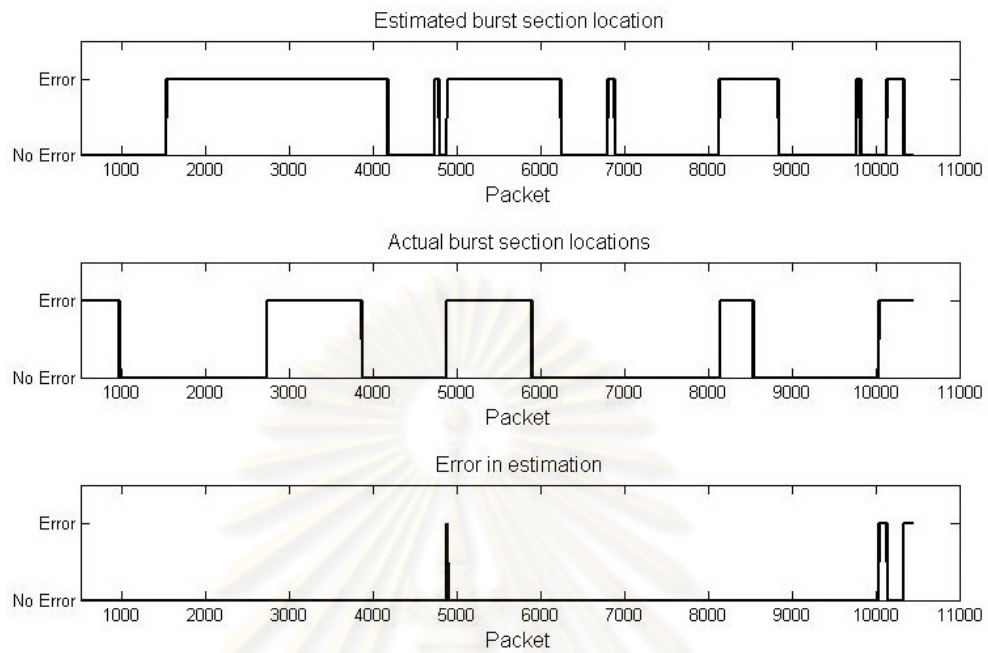


Figure A.5: Comparison actual and estimated channel state for "Akiyo" sequence in slow fading, 256 kbps.

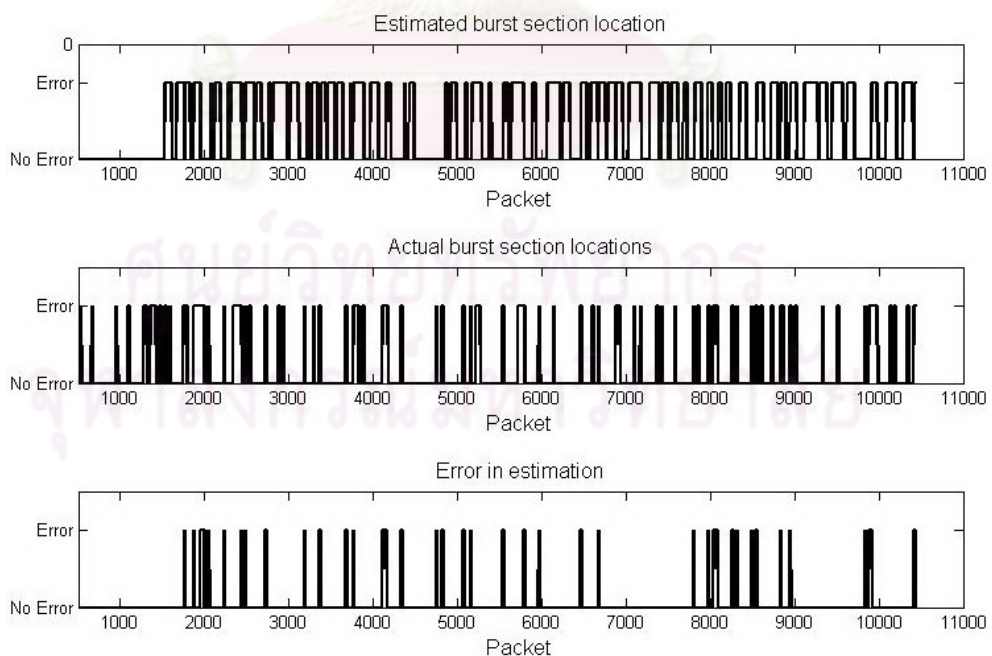


Figure A.6: Comparison actual and estimated channel state for "Akiyo" sequence in fast fading, 256 kbps.

APPENDIX B

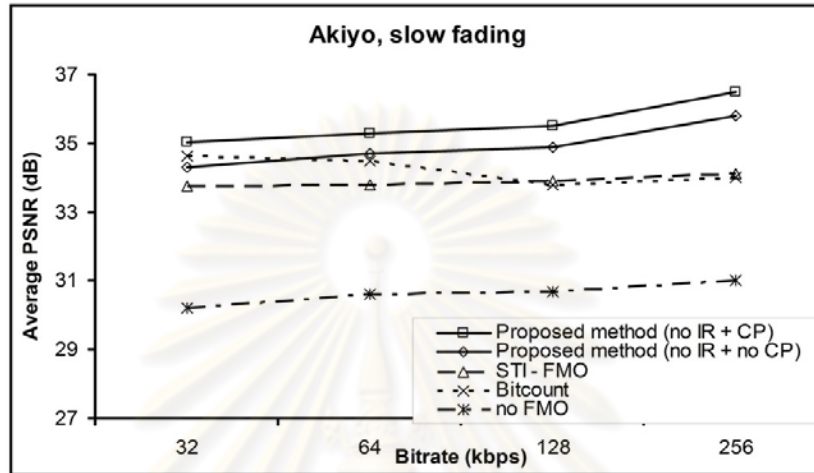


Figure B.1: PSNR comparison of “Akiyo” in slow fading at different bit rates.

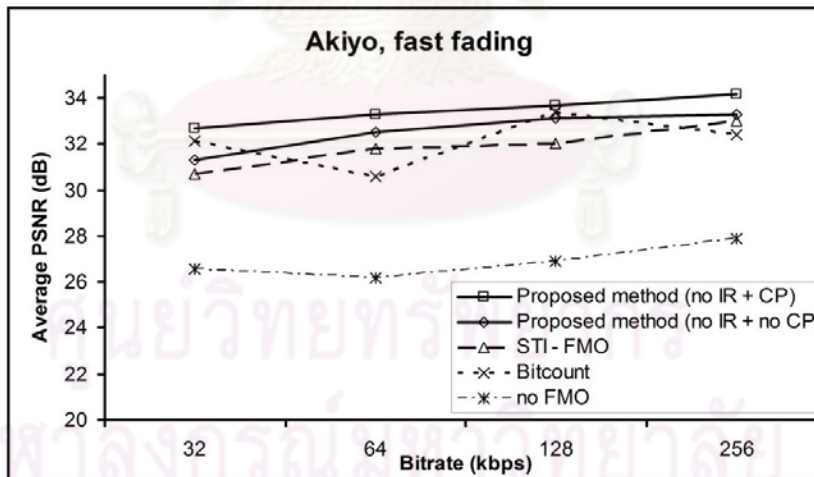


Figure B.2: PSNR comparison of “Akiyo” in fast fading at different bit rates.

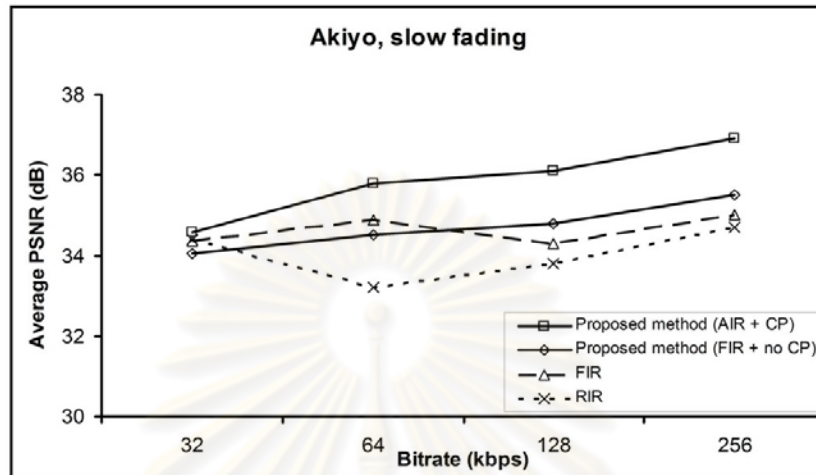


Figure B.3: PSNR comparison of "Akiyo" in slow fading at different bit rates.

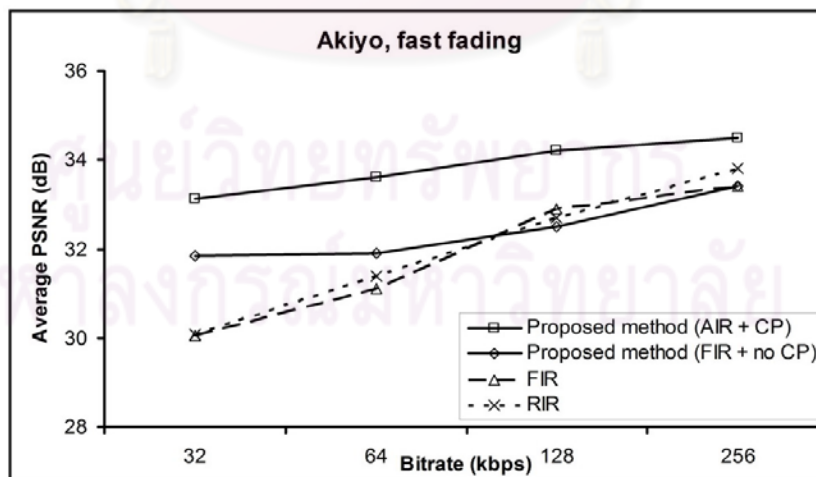


Figure B.4: PSNR comparison of "Akiyo" in fast fading at different bit rates.

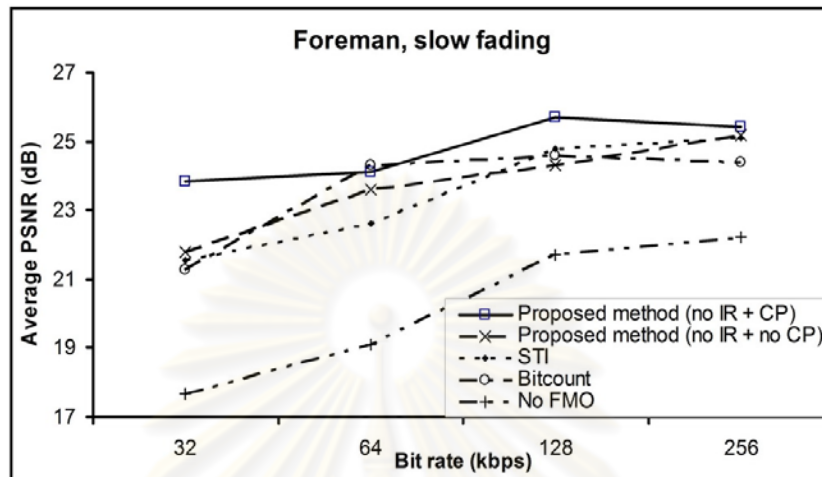


Figure B.5: PSNR comparison of "Foreman" in slow fading at different bit rates.

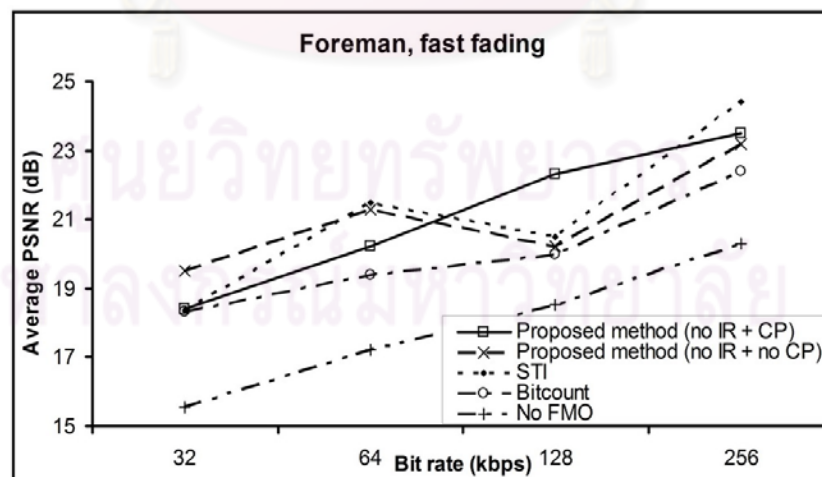


Figure B.6: PSNR comparison of "Foreman" in fast fading at different bit rates.

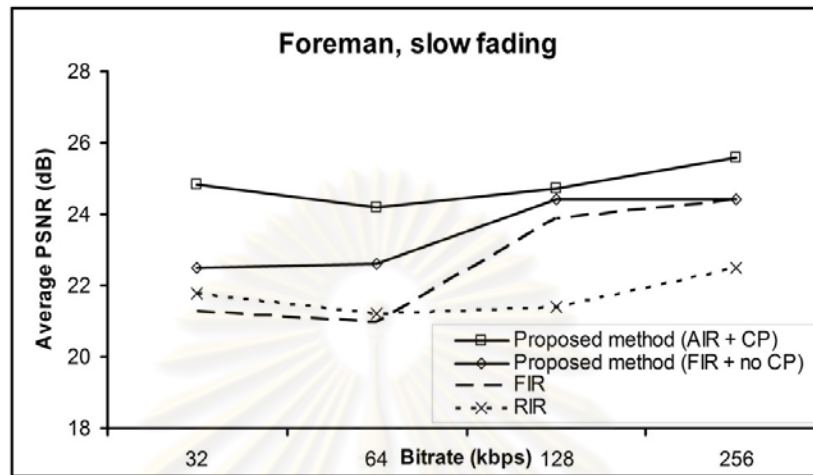


Figure B.7: PSNR comparison of “Foreman” in slow fading at different bit rates.

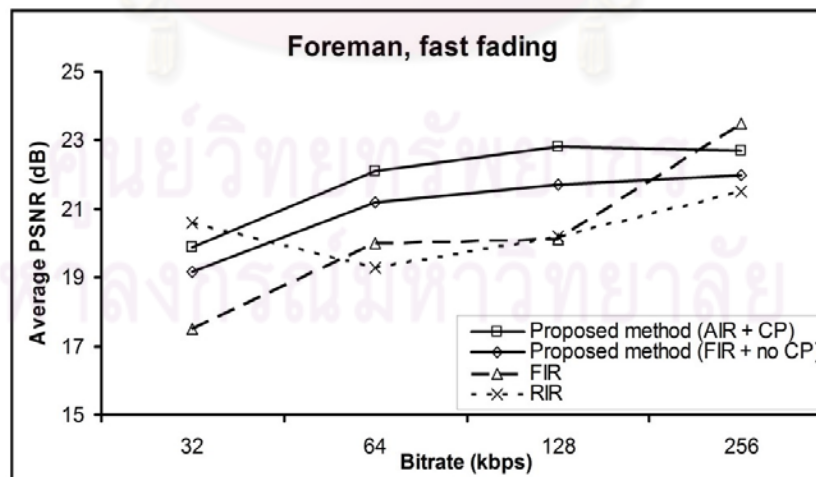


Figure B.8: PSNR comparison of “Foreman” in slow fading at different bit rates.

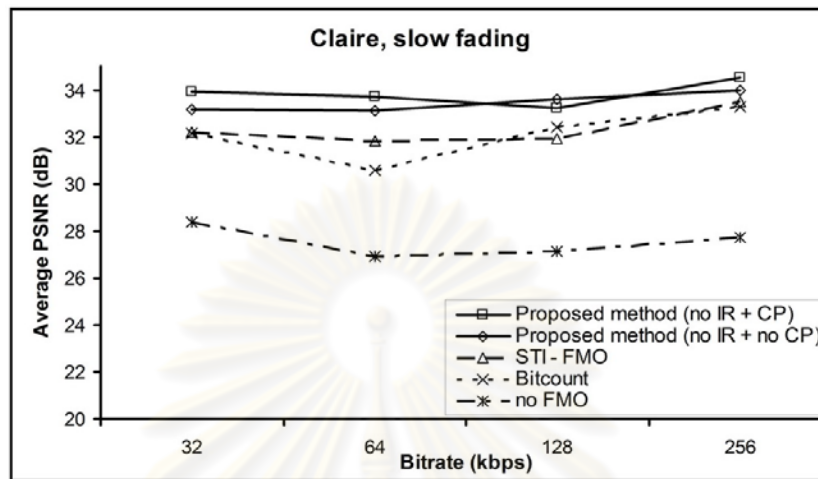


Figure B.9: PSNR comparison of "Claire" in slow fading at different bit rates.

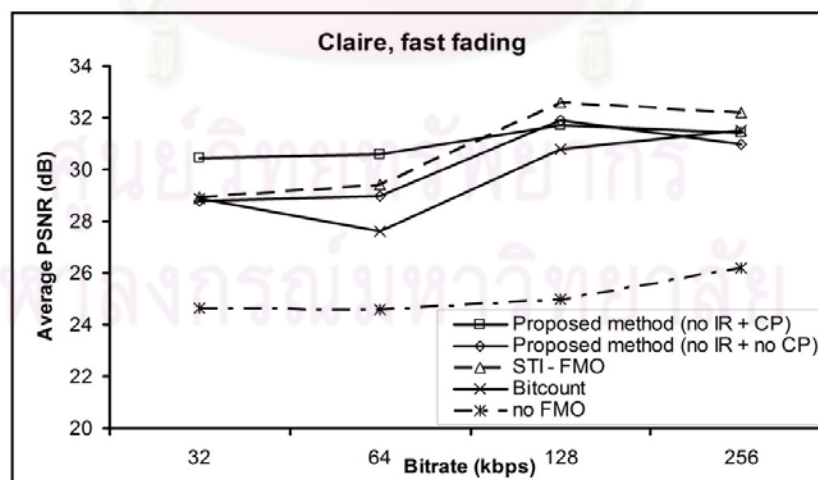


Figure B.10: PSNR comparison of "Claire" in fast fading at different bit rates.

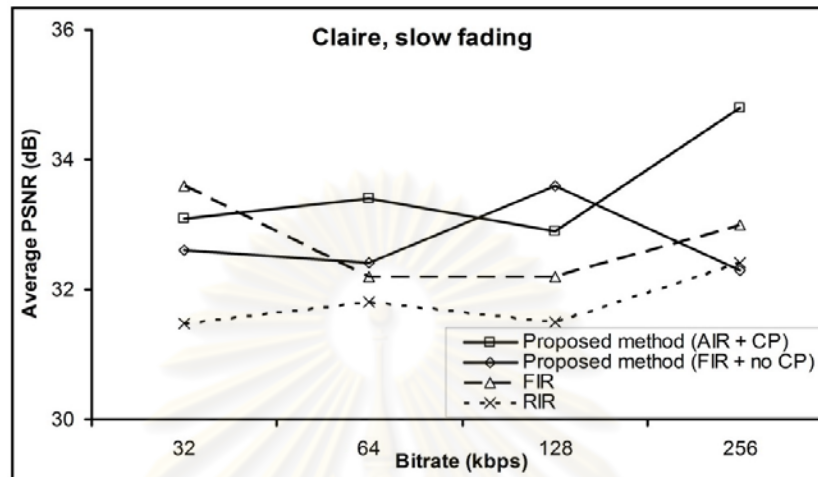


Figure B.11: PSNR comparison of “Claire” in slow fading at different bit rates.

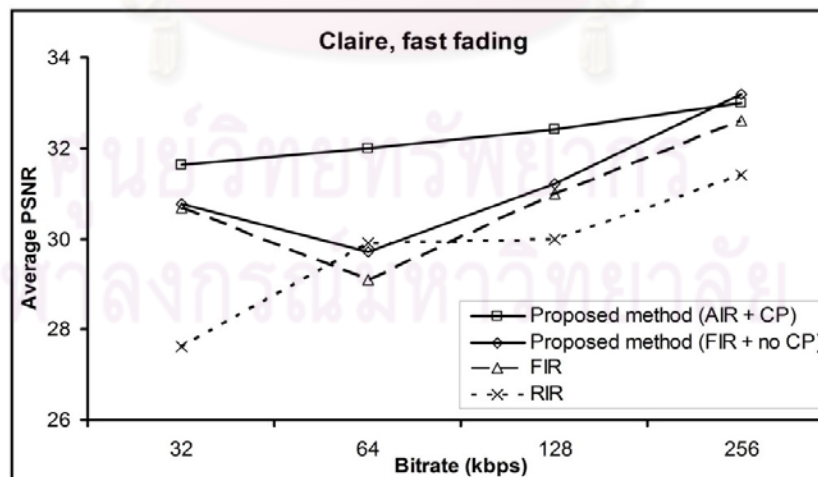


Figure B.12: PSNR comparison of “Claire” in slow fading at different bit rates.

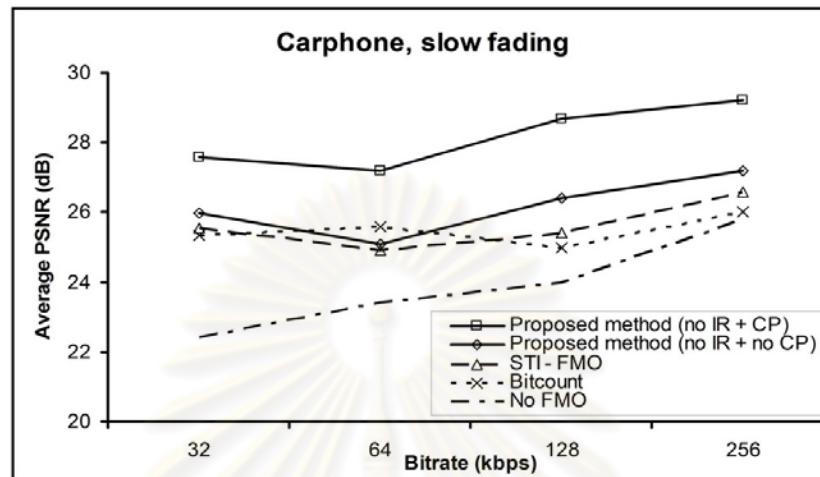


Figure B.13: PSNR comparison of “Carphone” in slow fading at different bit rates.

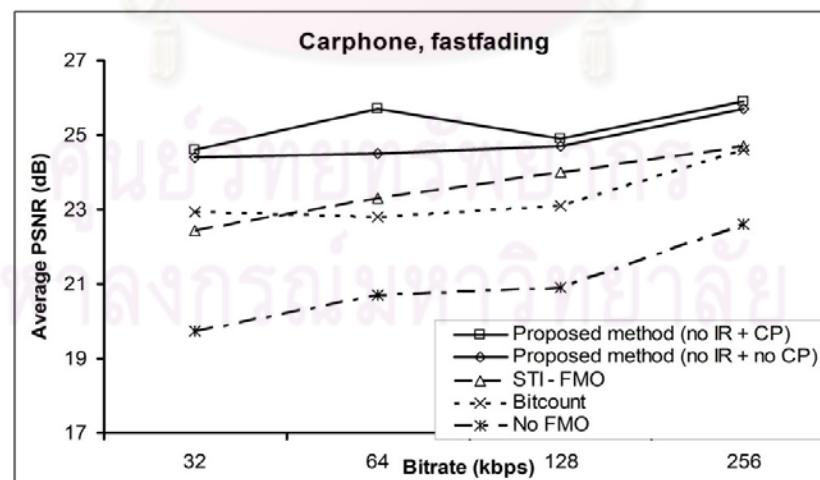


Figure B.14: PSNR comparison of “Carphone” in slow fading at different bit rates.

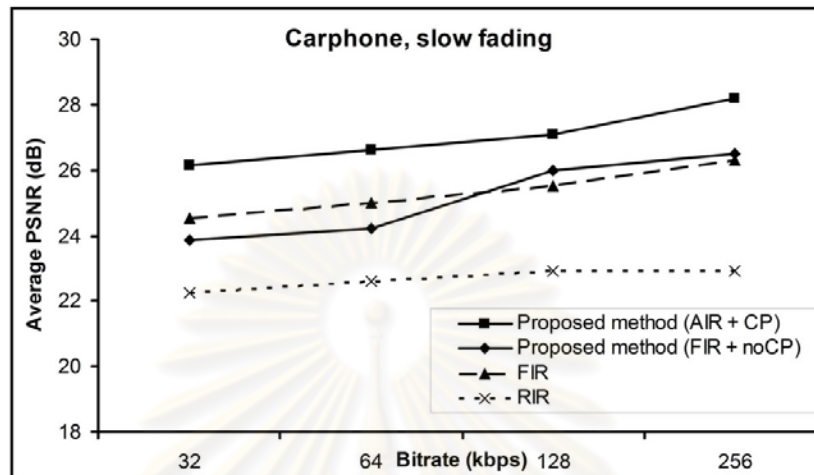


Figure B.15: PSNR comparison of “Carphone” in slow fading at different bit rates.

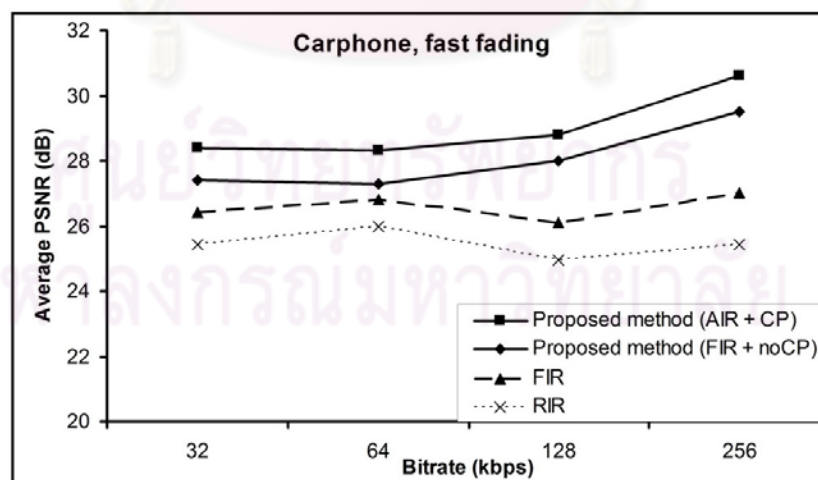


Figure B.16: PSNR comparison of “Carphone” in slow fading at different bit rates.

BIOGRAPHY

Vu Huu Tien was born in Hanoi, Vietnam, in 1979. He received his Master's degree in electrical engineering from Hanoi University of Technology, Vietnam, in 2005. He has been granted a scholarship by the AUN/SEED-Net (www.seed-net.org) to pursue his Doctoral degree in electrical engineering at Chulalongkorn University, Thailand, since 2007. He conducted his graduate study with the Digital Signal Processing Research Laboratory, Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University. His research interest includes Multimedia and Video Wireless Transmission.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย