Joint Source-Channel Coding for Image Transmission over Underlay Multichannel Cognitive Radio Networks

Long Pham
ltp6546@rit.edu

Follow this and additional works at: https://scholarworks.rit.edu/theses
Joint Source-Channel Coding for Image Transmission over Underlay Multichannel Cognitive Radio Networks

Long T. Pham
Joint Source-Channel Coding for Image Transmission over Underlay Multichannel Cognitive Radio Networks

LONG T. PHAM
July 2019

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

R·I·T | KATE GLEASON
College of ENGINEERING

Department of Computer Engineering
Committee Approval:

Dr. Andres Kwasinski  Advisor  Date
Department of Computer Engineering

Dr. Shanchieh Yang  Date
Department of Computer Engineering

Dr. Panos Markopoulos  Date
Department of Electrical Engineering
Acknowledgments

I would like to thank my Thesis Advisor, Dr. Andres Kwasinski, for all his support and guidance, not only with this thesis but also throughout my years at RIT. Without Dr. Kwasinski’s mentorship and direction, none of this is possible. I would also like to thank Dr. Shanchieh Yang and Dr. Panos Markopoulos for being part of my Master’s thesis committee.

I would like to thank all the professors, TAs, and staff members throughout my time here at RIT. I have come a long way since being a freshman thanks to their efforts in teaching me and challenging me to new heights.

I would like to thank fellow class of 2018 graduates, especially the senior design group, for making this college journey of mine all the more fulfilling. I’ve learned so much from them, inside and outside the classrooms.

I would like to thank my parents and my younger brother. They are always there when I need them the most. Their support and encouragement mean the world to me.
Abstract

The increasing prominence of wireless applications exacerbates the problem of radio spectrum scarcity and promotes the usage of Cognitive Radio (CR) in wireless networks. With underlay dynamic spectrum access, CRs can operate alongside Primary Users, the incumbent of a spectrum band, as long as they limit the interference to the Primary Users below a certain threshold. Multimedia streaming transmissions face stringent Quality of Services constraints on top of the CR interference constraints, as some packets in the data stream have higher levels of importance and are the most vulnerable to packet loss over the channel. This raises a need for Unequal Error Protection (ULP) for multimedia streams transmissions, in which the channel encoder assigns different amount of error correction to different parts of the data stream, thereby protecting more the most valuable parts of the stream from packet loss problems. This research presents an end-to-end system setup for image transmission, utilizing ULP as part of a Joint Source-Channel Coding scheme over a multichannel CR network operating through underlay dynamic spectrum access. The setup features a Set Partitioning in Hierarchical Trees (SPIHT) source encoder, and Reed-Solomon forward error correction channel coding, and uses their properties to devise an ULP framework that maximizes the quality of the received image.
## Contents

Signature Sheet i

Acknowledgments ii

Abstract iii

Table of Contents iv

List of Figures vii

List of Tables x

Acronyms xi

1 Introduction 2

2 Background 5

2.1 Cognitive Radio ........................................................................ 5

2.1.1 Overview ........................................................................... 5

2.1.2 Cross-layer Paradigm ......................................................... 6

2.2 Set Partitioning in Hierarchical Trees ....................................... 8

2.3 Joint Source-Channel Coding .................................................. 9

2.4 Unequal Loss Protection ......................................................... 12

2.4.1 Reed-Solomon (RS) Coding ............................................... 13
CONTENTS

3 Previous Work

3.1 Unequal Error Protection ........................................ 16
3.2 Cross-layer Design for ULP in CR Networks .................. 16
3.3 Neural Network Cognitive Engine for Underlay DSA ............ 18
3.4 JSCC for Progressive Transmission of Embedded Source Coders ... 19
3.5 Cooperative Learning for Reduced Complexity CR ............ 19

4 Proposed Methodology

4.1 Overall System Setup ........................................... 21
4.2 Implementation Details ......................................... 22
   4.2.1 Source Encoder - SPIHT .................................... 22
   4.2.2 Channel Encoder ........................................... 24
   4.2.3 Rest of System ............................................ 31

5 Experimental Evaluation

5.1 Experimental Setup ............................................ 32
   5.1.1 ELP Test ................................................ 32
5.2 Results and Analysis .......................................... 34
   5.2.1 SPIHT Encoder ........................................... 34
   5.2.2 Channel Encoder ......................................... 35

6 Conclusions & Future Work

v
Bibliography 57
# List of Figures

2.1 Three main DSA paradigms: (a) Interweave (b) Underlay (c) Overlay [4] ................................................................. 6

2.2 Parent-offspring dependencies in SPIHT [9] ................................................. 9

2.3 Traditional coding model (top) vs. JSCC (bottom) ................................. 10

2.4 An example of Reed-Solomon code .......................................................... 13

2.5 JSCC setup with error protection at the link layer .................................. 15

3.1 The hill-climbing algorithm in progress to assign FEC values to different streams [2] ................................................. 17

3.2 Implementing ULP for SPIHT bit stream [16] .......................................... 18

3.3 Progressive transmission of five source packets [18] .............................. 20

4.1 End-to-end system setup ........................................................................ 21

4.2 Block diagram of the research tool’s usage in this work .......................... 23

4.3 Average secondary network throughput under different PN loads [17] 25

4.4 Illustration of an FDD frame [22] ............................................................. 27

4.5 Bits selection from the output SPIHT stream ......................................... 31

5.1 Original Lena image ............................................................................. 33

5.2 Original Goldhill image ....................................................................... 33

5.3 Original Barbara image ....................................................................... 34
5.4 Comparison of SPIHT encoding efficiencies: (a) with the tool (b) original code [8] .................................................. 35
5.5 Comparison of SPIHT encoding efficiencies with the tool (a) compression ratio 8 (b) compression ratio 40 ........................................ 36
5.6 Lena’s average code rate selection results for ULP and ELP, codeword length 14 .................................................. 39
5.7 Goldhill’s average code rate selection results for ULP and ELP, codeword length 14 .................................................. 40
5.8 Barbara’s average code rate selection results for ULP and ELP, codeword length 14 .................................................. 41
5.9 Lena’s average PSNR results for ULP, codeword length 14 .......... 42
5.10 Goldhill’s average PSNR results for ULP, codeword length 14 .......... 43
5.11 Barbara’s average PSNR results for ULP, codeword length 14 .......... 43
5.12 Lena’s decoded images using ULP with codeword length 14 .......... 44
5.13 Goldhill’s decoded images using ULP with codeword length 14 .......... 45
5.14 Barbara’s decoded images using ULP with codeword length 14 .......... 46
5.15 Lena’s average PSNR results for ULP relative to ELP, codeword length 14 .................................................. 47
5.16 Goldhill’s average PSNR results for ULP relative to ELP, codeword length 14 .................................................. 47
5.17 Barbara’s average PSNR results for ULP relative to ELP, codeword length 14 .................................................. 48
5.18 Lena’s average code rate selection results for ULP and ELP, codeword length 15 .................................................. 49
5.19 Goldhill’s average code rate selection results for ULP and ELP, codeword length 15 .................................................. 50
5.20 Barbara’s average code rate selection results for ULP and ELP, codeword length 15 .................................................. 51

5.21 Lena’s average $\overline{PSNR}$ results for ULP, codeword length 15 ................................................................. 52

5.22 Goldhill’s average $\overline{PSNR}$ results for ULP, codeword length 14 ................................................................. 52

5.23 Barbara’s average $\overline{PSNR}$ results for ULP, codeword length 15 ................................................................. 53

5.24 Lena’s average $\overline{PSNR}$ results for ULP relative to ELP, codeword length 15 .................................................. 53

5.25 Goldhill’s average $\overline{PSNR}$ results for ULP relative to ELP, codeword length 15 .................................................. 54

5.26 Barbara’s average $\overline{PSNR}$ results for ULP relative to ELP, codeword length 15 .................................................. 54
List of Tables

4.1 PN load and secondary network throughput, PN relative throughput change = 2% .................................................. 26

5.1 $PSNR$ values classification for the test images .......................... 42
Acronyms

CR

Cognitive Radio

DSA

Dynamic Spectrum Access

ELP

Equal Loss Protection

FEC

Forward Error Correction

JSCC

Joint Source and Channel Coding

LIP

List of insignificant pixels

LIS

List of insignificant sets

LSP

List of significant pixels

LTE

Long Term Evolution
Acronyms

PSNR

Peak-Signal-to-Noise Ratio

RB

Resource Block

RS

Reed-Solomon

SINR

Signal-to-Interference-and-Noise Ratio

SNR

Signal-to-Noise Ratio

SPIHT

Set Partitioning in Hierarchical Trees

ULP

Unequal Loss Protection
The radio spectrum consists of extremely valuable frequency bands used for many purposes. Telecommunications, in particular, make heavy usage of the electromagnetic waves in this spectrum; not to mention radio broadcasting, communications between aircrafts and ships, radar, and industrial, scientific, and medical (ISM) uses. However, increasing demands in the spectrum, fueled by the ever growing number of connected devices and bandwidth demanding applications, leads to more congestion and interference within the fixed spectrum. Another serious problem is spectrum access in locations where users cannot utilize frequency bands unused by their incumbent owners due to strict regulations. A number of techniques have emerged to remedy these issues, among them, cognitive radio (CR).

Cognitive radio, as the name suggests, is an "intelligent" radio that gathers information about its surroundings and adjusts itself to operate in that environment [1]. As a result, the presence of CR will ensure that congestion and interference in the network are reduced as much as possible. Cognitive radio can also be deployed to mitigate the problem of spectrum access, by allowing the users to access the supposedly available frequency bands at the right time when the incumbent owners are not using them. With such capabilities, CR networks have various applications, including multimedia
transmissions.

Multimedia transmissions, namely images and videos, are especially effective in CR networks because these networks allow effective spectrum access for transmissions. However, multimedia transmissions have certain constraints on delay and distortion. An image, for example, when transmitted through the channel, should not lose too much information for it to be unrecognizable from the receiver side. This is difficult to achieve due to the problem of packet loss which happens naturally over CR networks. Some image packets carry substantial information about the image itself and therefore are less desirable to be lost. If a face portrait is to be transmitted, packets that allow recognition of the person (packets A) are more important than packets that show hair textures (packets B). Therefore, losing packets A during transmission is much more detrimental than losing packets B. In order to remedy this issue, different packets of varying levels of importance should be protected in varying degrees, so that more important packets have more robust error correction for transmissions over lossy conditions. The Unequal Loss Protection (ULP) [2] framework exists to combat packet loss problems for such transmissions, by assigning error-correction bits for packets through an algorithm that favors protecting more important packets. This thesis will follow this approach to facilitate image transmissions in a multichannel CR network. The overall system design can be described as follows: the input image is first compressed using SPIHT, an embedded coding algorithm which efficiently compresses packets with varying degrees of importance; then, a ULP framework is designed such that the channel encoder assigns different amounts of error correction bits to these packets in order to maximize the expected quality of the reconstructed image, measured in Peak Signal-to-Noise-Ratio (PSNR). The framework also takes into account various channel properties, such as the number of channels available, and the available channel bandwidths due to the primary user’s occupation of the band, As important packets are more protected, good media quality can be obtained.
even if 20% of packets are lost through the channel.
Chapter 2

Background

2.1 Cognitive Radio

2.1.1 Overview

Cognitive Radio (CR) is an "intelligent" wireless communication system capable of sensing the environment surrounding it, then uses the increased knowledge to "learn" and adapt its operating parameters accordingly, hence the name "cognitive". Cognitive radio is designed with the goals of best utilizing the radio spectrum and establishing reliable communications [1, 3]. The radio spectrum itself is a fixed natural resource heavily regulated by governments; however, it is the under-utilization of the radio spectrum that is a concern. The problem occurs when primary users, being licensed owners of certain frequency bands, frequently vacate them; when at the same time, other frequency bands are heavily used by secondary users. This creates spectrum holes, which are defined as bands of frequencies assigned to a primary user but not used by such user at a given time and location. The spectrum holes, as part of the Dynamic Spectrum Access (DSA) paradigm, represent opportunities for secondary users to access the spectrum, providing that they vacate promptly once the primary user arrives.
CHAPTER 2. BACKGROUND

Figure 2.1: Three main DSA paradigms: (a) Interweave (b) Underlay (c) Overlay [4]

As seen in Figure 2.1a, at time slot $t_1$, the secondary user is able to transmit at frequency band B5 but not in bands B1, B2 or B3; while at the next time slot $t_2$, the frequency bands B1, B2 or B3 are all available but B5 is not.

However, secondary users need not to find spectrum holes to be able to use the bands; they can be assigned as the secondary network and are allowed to coexist with the primary network, under the premise that the secondary network does not produce excessive interference that can disrupt the operation of the primary network. In Figure 2.1b, the secondary user has to keep its transmit power at a respectable level, but otherwise can use any frequency band, even if it is occupied by a primary user. This is what defines the Underlay DSA paradigm [4, 5], for which as long as the interference limit is kept, secondary users can transmit without the knowledge of the arrivals of the primary user.

2.1.2 Cross-layer Paradigm

A traditional OSI network protocol architecture model typically consists of seven layers, each performing a separate function; for example, the Application layer is re-
sponsible for high-level APIs and the whole application itself, the Physical layer at the other end of the layer stack is responsible for transmitting and receiving bits over the physical channel. For OSI models, the key aspect is that implementation details of a layer are typically not known by other layers; and that layers only exchange information with their neighboring layers [6]. The layered structure also allowed modular design for standardization and for layers to develop independently of other layers as newer technologies are deployed. However, its performance in wireless networks is severely limited by the transmission medium, in which the layered architecture is not very effective. For example, TCP/IP’s lack of communication between the link layer (which handles the channel errors), and the transport layer (which performs congestion control) hinders performance in the case of transmission in fading wireless networks [6, 7].

The Cross Layer (CL) design exists to address this and other analogous performance issue. Layers retain their core responsibilities as defined in the OSI model; however, different layers that are not neighbors can interact and coordinate to optimize certain protocols. In the case of CR, cross-layer design is particularly effective and efficient. A CR has to be able to perform four essential functions: spectrum sensing (monitoring the spectrum and identifying frequency bands appropriate for transmissions), spectrum allocation (selecting the best frequency band out of the candidates obtained through sensing), spectrum sharing (coordinating access to the spectrum to avoid interference to the primary users and collisions among CR users), and spectrum mobility (moving to a different spectrum hole should a primary user requires the incumbent band). The cross-layer design allows each of these functions to span multiple layers, increasing their overall effectiveness [6].
2.2 Set Partitioning in Hierarchical Trees

Before discussing Set Partitioning in Hierarchical Trees (SPIHT), it is worth noting that this thesis deals only with image transmissions over CR networks. SPIHT is a powerful image compression method, providing many benefits altogether that very few other methods can achieve such as high image quality, progressive image transmission, fast coding and decoding, and error protection. SPIHT accomplishes all of these because by its definition, it divides an image into subbands of increased spatial frequency resolution for the lower frequencies through the means of wavelet transform. The application of wavelet transforms on the vertical and horizontal axes of the image produce subband coefficients which are then sorted by their order of significance [8]. Figure 2.2 shows the tree structure of the input bits, from which they are classified into three separate sets of data: list of insignificant sets (LIS), list of insignificant pixels (LIP), and list of significant pixels (LSP). The SPIHT algorithm consists of four distinct steps: initialization, sorting pass, refinement pass, and quantization-step update; during the sorting pass phase, non-significant LIP pixels get moved to the LSP, as well as significant sets being removed from LIS, while during the refinement pass phase, the LSP entries get selected in order of their coefficient significance. The output stream after encoding, therefore, contains bits in decreasing order of significance, with more bits meaning finer resolution for the coefficients quantized representation. The important property of SPIHT is that the output bit stream can be truncated anywhere and the encoded image can still be reconstructed at varied levels of quality from the most significant bits in the stream after truncation.
Joint Source-Channel Coding (JSCC) is a coding approach that can be particularly useful for media transmission, as the source and channel coding rate can be easily and jointly adapted for different needs [10]. Traditionally, transmission of information over a medium requires an independent pair of source and channel codecs [11]. From the transmitter, the information source will go through the source encoder first, followed by the channel encoder; the order is reversed on the receiver side, where the source decoder follows the channel decoder. Source coding tends to reduce the number of transmitted bits as it aims to represent the source in the most efficient manner while maintaining fidelity, and channel coding tends to increase the number of transmitted bits in order to make information more resilient to negative channel effects.
Figure 2.3: Traditional coding model (top) vs. JSCC (bottom)

With JSCC, the parameters of the two codecs will be jointly configured as a tandem. Figure 2.3 shows this distinction. The primary goal of the code rate adaptability is to balance distortions from both the source encoder, the "source coding distortion" \( D_S \), and the channel encoder, the "channel-induced distortion" \( D_C \) [11]:

\[
D = D_S + D_C
\]  

(2.1)

Let the source signal be denoted as \( s[n] \). It is then sampled, quantized, transmitted, and finally is performed reverse operations as the transmitter side to produce the output signal \( \tilde{s}[n] \). Because \( s[n] \) is sampled, it can be expressed in the following form:

\[
s[n] = \sum_{k=1}^{N} s_k \phi_k(t) \quad k = 1, 2, \ldots, N
\]  

(2.2)

where the quantities \( \phi_k(t) \) are part of the orthonormal basis used to reconstruct \( s(n) \).
The received signal $\tilde{s}[n]$ can be expressed in the same form:

$$
\tilde{s}[n] = \sum_{l=1}^{N} \tilde{s}_l \phi_l(t) \quad l = 1, 2, \ldots, N
$$

With these quantities being defined as well the transmitted symbol duration being $T$ and the quantized sample $s_k$ being $\hat{s}_k$, the source coding distortion can be expressed as follows:

$$
D_S = E \left[ \sum_{k=1}^{N} (s_k - \hat{s}_k)^2 \right]
$$

Similarly the channel-induced distortion can be expressed as follows:

$$
D_C = E \left[ \sum_{k=1}^{N} (\hat{s}_k - \tilde{s}_k)^2 \right]
$$

The end-to-end distortion $D$ essentially covers all the effects affecting the transmission, including source and channel distortions [11]. Typically with more compression at the source, there is room for more channel coding error protection which reduces the channel-induced distortion; however it comes at a cost of more source coding distortion. In addition, with more compression at the source, the part after compression is more important. Therefore, the amount of increase in source coding distortion depends in not only the compression ratio, but also the amount of error protection. This is part of the interaction between the source and channel coding with regards to minimizing the end-to-end distortion.
Once the JSCC system has its rate configured to minimize the end-to-end distortion, it is used to transmit images with the goal of achieving the highest quality possible at the receiver side. From the transmitter to the receiver, an image has to face end-to-end distortion, channel packet loss, and many other factors which cause its quality to decrease. The image is considered to be contaminated with noise, and more noise contamination further decreases image quality, which is commonly measured in PSNR as follows:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right),$$

where $MSE$ indicates the mean squared-error between the source image and the reconstructed image. Higher PSNR values means lower MSE and thus higher decoded image quality.

### 2.4 Unequal Loss Protection

Unequal Loss Protection (ULP) is one of the JSCC approaches that is especially suitable for embedded and coded image transmission. As the name suggests, ULP assigns different error-correction bits to different parts of the bit stream, as opposed to Equal Loss Protection (ELP) where all parts of the bit stream are protected equally; ULP thus allows for more efficient error protection: in the case of ELP, less important packets might need more error-correction bits to go with than needed. Joint Source Channel Coding usually consists of a compression scheme that is embedded such as SPIHT, and a variable-rate channel coding scheme. For the specific purpose of transmitting images where certain parts of the encoded stream are more important than others, a channel coding scheme like Reed-Solomon which could work hand-in-
hand with SPIHT (by its nature sorting bits by their significance). Forward Error Correction is performed by giving each message fragment the appropriate amount of correction bits based on their importance to combat packet loss issues [12], hence the name "Unequal Loss Protection". The use of a JSCC approach with ULP and SPIHT being utilized will make sure that information of different importance can be protected against transmission errors with codes of different strength in their protections.

2.4.1 Reed-Solomon (RS) Coding

RS coding is an error correction mechanism that has widespread usage, ranging from compact discs to spacecrafts [13]. As a form of binary linear block codes, a RS code can detect transmission errors and correct them [11]. More specifically, if given that the RS code is \((n, k)\), where \(n\) is the codeword length in bits and \(k\) is the input length in bits; the code can detect a number of \(d\) errors providing that \(d \leq n - k\), and can correct a number of \(t\) errors providing that \(t \leq \lfloor \frac{n-k}{2} \rfloor\).

![Figure 2.4: An example of Reed-Solomon code](image)

Figure 2.4 showed that the codeword contains \(k\) bits of information and \(2t\) bits of error correction and therefore has length \(n = k + 2t\). At the decoder side, if the RS
code is \((n, k)\), as long as the decoder receives \(k\) symbols or more out of the \(n\) symbols sent, the original input bits are protected and properly decoded. For transmissions of images and videos where different parts of the bit stream have different levels of importance, RS code allows for better protection of the most important segments in the stream by reducing the value of \(k\) (the number of input bits), while keeping \(n\) fixed, thereby increasing the value of \((n - k)\), the number of redundancy bits for error correction. If \(k\) is lower, the probability that the decoder receives \(k\) or more symbols (or loses \((n - k)\) or less symbols) is reduced as well.

2.4.2 Link Layer Protection

In addition to RS coding, there is another set of error protection at the link layer (more specifically, Turbo code) that occurs when an image is transmitted through a LTE channel. Figure 2.5 shows the JSCC setup from the transmitter side with this error protection block included. The technique utilized to perform the error protection task at the link layer is link adaptation - where the modulation and coding is adapted based on the condition at the link: which systems are being used, the link quality, etc. This adaptive modulation scheme, for the purpose of this thesis, is assumed to be already in place; as well as the modulation placed at resource blocks (RB) that notifies the receiver whether the RB has transmitted its content successfully.
Figure 2.5: JSCC setup with error protection at the link layer
Chapter 3

Previous Work

3.1 Unequal Error Protection

In [2], Mohr and Riskin presented a channel encoding algorithm that was particularly effective for transmitting images under various packet loss rate conditions. They utilized RS Coding to protect different parts of the bit stream with different amount of error correction bits. A hill-climbing algorithm, an illustration of which is shown in Figure 3.1, was used to find the most optimal assignment of error corrections. Only one channel was used, although it can assume any type of packet loss distribution model such as Poisson or exponential. The result was that the image quality gracefully degrades as packet loss rates increase.

3.2 Cross-layer Design for ULP in CR Networks

Prior work has been conducted on designing cross-layer CR networks that implement error correction schemes such as ULP and Equal Loss Protection (ELP). A. Chaoub and E. Ibn-Elhaj studied secondary multimedia transmissions in interlaced DSA CR networks under delay constraints, which facilitate the cross-layer design and the use
CHAPTER 3. PREVIOUS WORK

Figure 3.1: The hill-climbing algorithm in progress to assign FEC values to different streams [2]

of ULP [14]. The primary user pattern in the network is observed as a Poisson process [15], allowing secondary users to predictably transmit whenever the primary user does not occupy the frequency bands. The system is JSCC with SPIHT acting as the source coding scheme, RS Coding as the channel coding scheme, and the ULP and ELP frameworks as forward error correction mechanisms.

Taken into account various factors such as subchannel fading, primary user arrival rate, and secondary transmission miss detection rate, a metric was devised to rank subchannels in order of suitability for transmissions. An optimization problem is solved to determine how many redundancy bits to add to each message fragment, with a goal of maximizing the PSNR of the received media. This is shown in Figure 3.2, where the bit stream is divided into “layers” and then from each layer is further divided into message fragments that form parts of the packet to be transmitted. This approach yields excellent results, as the quality of transmitted images degrades gracefully as packet losses increase for the network.
3.3 Neural Network Cognitive Engine for Underlay DSA

Shah Mohammadi and Kwasinski presented in [17] a cognitive engine operating efficiently within the Underlay DSA paradigm. As described in Section 2.1.1, for this paradigm, secondary users (SU) can transmit as long as the interference they cause on the primary network remains below a certain threshold. The threshold might be difficult to determine if no information is exchanged between the primary and the secondary networks. As such, Shah Mohammadi and Kwasinski proposed a fully autonomous and distributed scheme, using a cognitive engine that allows SUs to predict their transmission effects on the primary network and adjust accordingly. The predictions are made based on primary link throughput estimations, which are inputs to NARX-NN, a state-of-the-art neural network that is excellent for time series predic-
tion, and therefore a great fit for predicting the primary link throughput as a time series. This approach is notable for its efficiency: The accuracy of predictions closely match those from performing exhaustive search, all without information coming from the primary network.

3.4 JSCC for Progressive Transmission of Embedded Source Coders

Chande and Farvardin in [18] devised a JSCC scheme which allowed transmissions of sources compressed by embedded source coders, such as SPIHT, over a memoryless noisy channel. They used rate-compatible punctured convolutional codes for channel coding, while also devised mathematical expressions for the expected distortion, expected PSNR, and the average useful source coding rate. These quantities were then optimized through solving dynamic problems subject to a certain rate constraint. Figure 3.3 shows the inverse code rate profile for a transmission of five packets. The profile decreases as the packet indices increase. The labels 1 through 5 indicate the order to transmit bits within packets. With this, they achieved optimal progressive transmission at all transmission rates [18].

3.5 Cooperative Learning for Reduced Complexity CR

Kwasinski and Wang tackled in [19] the issue of reducing the complexity of cross-layer CR networks through cooperative learning. The cross-layer approach, while effective, requires more effort to solve the resource allocation problem. Cooperative learning was proposed to reduce the complexity while not compromising performance. In this framework, the "learning" that needs to be done is for resource allocation through
reinforcement learning. Secondary users distribute among themselves the tasks to learn; afterwards they all share the newly acquired knowledge. It resulted in the complexity being reduced in half while performance was reduced by an acceptable amount.

Figure 3.3: Progressive transmission of five source packets [18]
Chapter 4

Proposed Methodology

4.1 Overall System Setup

Figure 4.1 shows the proposed end-to-end system setup in the secondary network, where it is operating under the underlay DSA paradigm.

![Figure 4.1: End-to-end system setup](image)

This setup is a standard JSCC setup with source and channel codecs being indepen-
CHAPTE. 4. PROPOSED METHODOLOGY

dently designed. Regular JSCC setup is very effective for multimedia transmissions because of its ability to trade off between distortion and compression rate to reduce consumption of bandwidth [20]. This setup is similar to those proposed by Chaoub and Ibn-Elhaj [14], Mohr and Riskin [2], and Chande and Farvardin [18]. However, Chaoub and Ibn-Elhaj’s model operates under the interweave DSA paradigm, where the secondary network is ”opportunistic” in its transmissions. This scheme finds spectrum holes through the estimation of the primary network’s arrival as a Poisson process. The setup in this thesis operates under the underlay DSA paradigm and can transmit at any time and at any frequency band, providing that the interference temperature caused at the primary network is kept at a reasonable level. Mohr and Riskin uses only a single channel while incorporating different channel loss profiles such as uniform, binomial, and exponential distributions [2]. This thesis allows for multi-channel transmissions. Finally Chande and Farvardin utilizes rate-compatible punctured convolutional codes as channel codes so that the transmission is optimized at any transmission rate; whereas this work uses Reed-Solomon codes.

4.2 Implementation Details

4.2.1 Source Encoder - SPIHT

This is the first stage of the overall system, in which the bitmap image is source-encoded using SPIHT. Depending on several factors (wavelet filter type, number of filtering levels, number of bit planes to decode, and compression ratio), the resulting encoded bit stream will change despite the starting image being the same. This bit stream will then go through the channel encoding stage and the channel, before being decoded and examined for performance. Additionally, a research tool developed by Andres Kwasinski is utilized in this stage for its capability in performing end-to-end
SPIHT encoding and decoding as well as evaluating results.

![Block diagram](image)

**Figure 4.2:** Block diagram of the research tool’s usage in this work

Figure 4.2 shows the block diagram describing the contribution of the tool to this research. As seen above, the tool accepts the bitmap raw image and the SPIHT encoding parameters as inputs. It then goes through the SPIHT encoding stage, after which the generated bit stream is extracted to go through the next stages in the setup. At the decoding stage, the stream is truncated at various percentage amounts, where only a percentage of the stream, starting from the beginning of the stream itself, is used for decoding. The cutoff percentage amounts are chosen to be from 10% (keep majority of stream) to 95% (keep only the important part), with 10% increment. With each cutoff percentage amount, the decoding stage produces a different received image and an associated SNR value for that image. From these SNR values, given the input image and any cutoff percentage amount, the SNR can be calculated through the use of linear interpolation. This is utilized later in the process, because it is desired to keep as much of the bit stream as possible to ensure high PSNR output. Knowing the PSNR values at different cutoff percentages helps finding out which part of the stream is valuable to the SPIHT decoder, and makes the error protection mechanism effective over lossy channels.
4.2.2 Channel Encoder

The channel encoder is designed in a way to maximize the expected PSNR for the image, taken into account various factors such as the PSNR obtained at different source bit stream truncation cutoff percentages, the number of channels available for transmission, the capacity of such channels, and the number of packets intended to be sent.

\[ a) \text{ RS Coding} \]

RS coding is used here as erasure codes to provide the forward error correction (FEC) property to the bit stream, now placed in source packets, and protect it from channel errors. For the code \( RS(N, k) \) containing \( N \) packets (or symbols), the entire codeword is in error if there are more than \( t = \lfloor \frac{N-k}{2} \rfloor \) packet losses. If the probability of packet loss is \( \pi \), for each value of \( m > t \), the probability of the codeword being in error is equivalent to the probability of losing any \( m \) packets out of the original \( N \):

\[
P_e(N, m, \pi) = \binom{N}{m} \pi^m (1 - \pi)^{N-m} \tag{4.1}
\]

Therefore, the probability of decoding error for code \( RS(N, k) \) is the probability of losing \( t \) or more packets based on Equation 4.1, and also as described in [21]:

\[
P_e(N, t, \pi) = \sum_{m=t+1}^{N} \binom{N}{m} \pi^m (1 - \pi)^{N-m} \tag{4.2}
\]

\[ b) \text{ Channel Properties} \]

As the setup is based upon the secondary network transmission, the conditions at the primary network (PN) dictate how much data, including redundancies, can be sent
through the channel. Kwasinski and Shah Mohamadi in [17] were able to implement a cognitive engine that allows the secondary users to transmit autonomously at the largest possible throughput value allowed by the PN. The result of their work is shown in Figure 4.3 where the PN load and the secondary network throughput was calculated under different operating conditions. Out of these results, utilized in this thesis is the curve where the relative throughput change at the PN is less than 2% and seven probe messages are transmitted, because this condition affects the primary network the least (only 2% relative throughput change compared to 5% and 10% in others). For this curve, the values of the PN load and the secondary network throughput formed a linear relationship, allowing the throughput to be estimated based on the PN load at any point in time.

![Figure 4.3: Average secondary network throughput under different PN loads [17]](image)

If given that the maximum PN relative throughput change is 2%, and denote the primary network load to be $X$ and the secondary network throughput in Mbps to be
Table 4.1: PN load and secondary network throughput, PN relative throughput change = 2%

<table>
<thead>
<tr>
<th>Primary Network Load</th>
<th>Secondary Network Throughput [Mbps]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.16</td>
<td>0.1325</td>
</tr>
<tr>
<td>0.32</td>
<td>0.1</td>
</tr>
<tr>
<td>0.48</td>
<td>0.07</td>
</tr>
<tr>
<td>0.64</td>
<td>0.05</td>
</tr>
</tbody>
</table>

$Y$, the relationship between $X$ and $Y$ can be approximated from Figure 4.3 as

$$Y = aX + b$$  \hspace{1cm} (4.3)

Where $a$ is calculated to be -0.1734 and $b$ is calculated to be 0.1535, using linear regression for the data points given in Table 4.1. Because $X$ is a random variable, $Y$, as a function of $X$, is also a random variable. As the primary network load, we assumed that $X$ follows an uniform distribution between 0.16 and 0.64, which means its cdf is

$$F_X(x) = \frac{x - 0.16}{0.48}.$$  \hspace{1cm} (4.4)

From equation 4.3 the cdf of $Y$ is calculated as,

$$F_Y(y) = 1 - F_X \left( \frac{y - b}{a} \right), \quad a < 0.$$  \hspace{1cm} (4.5)

Combining (4.4) and (4.5) the cdf of $Y$ is

$$F_Y(y) = 1 - \frac{y - b - 0.16}{0.48}$$

$$= 1 - \frac{y - b - 0.16a}{0.48a}$$

$$= \frac{-1}{0.48a}y + z,$$  \hspace{1cm} (4.6)
and, therefore, the pdf of $Y$ is

\[ f_Y(y) = F_Y'(y) = \frac{-1}{0.48a} = \frac{1}{0.0832}. \]  

(4.7)

$Y$, therefore, is uniformly distributed between 0.1325 and 0.05.

c) LTE

The setup for this research consists of a multichannel CR network, where there are multiple LTE channels. Each channel corresponds to one resource block as part of a LTE frame lasting 10ms in duration. The frame can contain 20 RBs horizontally and up to 50 RBs vertically if the bandwidth is 10Mhz. Figure 4.4 shows an illustration of an FDD frame, where a frame contains 120 RBs (20 horizontal, 6 vertical for 1.4MHz bandwidth) at a duration of 0.5ms each.

![LTE FDD Frame](image)

**Figure 4.4:** Illustration of an FDD frame [22]

Knowing the value of $Y$ and the RB duration (0.5ms), the possible amount of bits transmitted over the channel can be calculated. Per each RB, if the throughput in the channel is $Y$ [Mbps]. Each RB lasts for 0.5ms, therefore in a second there are
1s/0.5ms = 2000 RBs. This leads to the number of bits per RB as:

\[
\frac{1000000Y\text{[bits/sec]}}{2000\text{[RB/sec]}} = 500Y\text{[bits/RB]}. \tag{4.8}
\]

As \( Y \) takes values between 0.05 and 0.1325, the number of bits per RB is between 25 and 66. For a horizontal segment of the frame with 20 RBs in it, the capacity of a channel, then, is anywhere from 500[bits/frame] to 1320[bits/frame]. The more channels that exist, the higher the amount of bits that could be sent over through every channel without exceeding each channel capacity.

d) ULP FEC Allocation Problem Description

The ULP assignment of error-correction bits to packets in the stream needs to maximize the expected quality of the output image, or in other words, the expected PSNR of the output image, denoted as \( \overline{\text{PSNR}} \). Given RS code \((N,k)\) where the probability of symbol error is \( \pi \), as well as the total amount of source packets being \( S \), \( \overline{\text{PSNR}} \) is calculated using the following formula:

\[
\overline{\text{PSNR}} \triangleq \sum_{i=0}^{S} \text{PSNR}(P_i) P_{i|0}(\pi) \tag{4.9}
\]

where: \( \text{PSNR}(P_i) \) is the PSNR performance of the source coder after the \( i^{th} \) packet is decoded successfully, and the conditional probability \( P_{i|0}(\pi) \), introduced in [11] as the probability that the first \( i \) source packets are decoded correctly given that the
first 0 packets are decoded correctly, is defined as

\[
P_{i|0}(\pi) \triangleq \begin{cases} 
P_e(N, 1, \pi), & i = 0, \\ 
P_e(N, i + 1, \pi) \prod_{j=1}^{i}(1 - P_e(N, j, \pi)), & i = 1, 2, \ldots, S - 1, \\ 
\prod_{j=1}^{S}(1 - P_e(N, j, \pi)), & i = S, 
\end{cases} \quad (4.10)
\]

where \(P_e(N, t, \pi)\) was defined in (4.2).

Denote the family of error correction-detection channel codes by \(\mathcal{C} = \{c_1, c_2, \ldots, c_N\}\) and the Reed-Solomon code rates to be \(r_c(c_i)\), \(i = 1, \ldots, N\). This means that a codeword for the source packet with length \(k\) bits and protected by code \(c_i\) has length \(k/r_c(c_i)\) bits. Now define the probability of packet decoding failure for a given channel with code \(c_i \in \mathcal{C}\) to be \(P_e(c_i)\) (so \(P_e(c_0) = 1\)). With that in mind, the ULP allocation comes into play to assign a channel code to each source packet before transmission.

A code allocation policy \(\phi\) is created to allocate channel code \(c^i_{\phi} \in \mathcal{C}\) to the \(i^{th}\) packet out of the source coder. The policy \(\phi\) contains a sequence of channel codes \(\{c^1_{\phi}, c^2_{\phi}, \ldots, c^N_{\phi}\}\). The normalized transmission rate for such policy (in channel bits per source sample), given that \(r_s\) is the bits per sample source packet, is therefore

\[
R_{T_x} \triangleq \sum_{i=1}^{S} \frac{r_s}{r_c(c^i_{\phi})}. \quad (4.11)
\]

The optimization problem then is defined as

\[
\max PSNR \text{ subject to } R_{T_x} \leq R \quad (4.12)
\]

with \(R\) being the total transmission capacity available.
CHAPTER 4. PROPOSED METHODOLOGY

e) Solving the ULP FEC Allocation Problem

For RS code, the family code to pick from is \{1/N, 2/N, \ldots, N/N\}. Each source packet out of S in the stream will be assigned a code from this list, and therefore there are \(N^S\) possible ways to assign FEC bits to the packets. To find the best ULP FEC allocation, every single combination out of these \(N^S\) ways is examined, and the combination that yields the highest expected PSNR is chosen to encode the packets.

From (4.10) it could be seen that the conditional probability \(P_i|0(\pi)\) is constant if \(S\), the number of packets, as well as \(\pi\), the probability of symbol error, stay the same. If \(P_i|0(\pi)\) is constant, according to Equation 4.9, the value of the quantity \(PSNR(P_i)\) determines the maximum \(PSNR\) for the image.

The calculation of \(PSNR(P_i)\) depends on many factors. First, from the channel statistics, it is possible to find the maximum capacity of a channel to send bits through. Equation (4.8) can be used to find the maximum amount of bits a channel can handle per frame, which is 20 times of the amount the channel can handle per resource block. The total amount of bits the channel can handle is simply the sum of its subchannels. For example, if the subchannels have the Y-values equal to \{0.05, 0.08, 0.1\}, then their respective bits per resource block is \{25, 40, 50\}, which leads to the capacity per channel being \{500, 800, 1000\} and the total amount of bits possible to send equal to 2300. Denote this quantity, the calculated channel capacity, as \(V\). Then the channel encoder can only send \(V\) amount of bits through, and since the stream is already SPIHT-encoded, the channel encoder will send the first \(V\) bits of the stream as shown in Figure 4.5. Because there are \(S\) packets and the codeword length is \(N\), that leaves \(\lfloor \frac{V}{SN} \rfloor\) bits per packet, information or redundancy. A RS Code \((N,k)\) will select \(k\lfloor \frac{V}{SN} \rfloor\) bits from the stream as data, while leaving the rest of the source packet to be redundancy bits. From here, the output from the source encoder, as mentioned in Section 4.2.1, is used to calculate the value of \(PSNR(P_i)\).
BIT STREAM

BIT STREAM AFTER SPIHT

| K1 | K2 | K3 | K4 | K5 |

TOTAL BITS TO SEND

| V |

BITS FROM THE STREAM TO SEND

| K1 | K2 | K3 | K4 |

**Figure 4.5:** Bits selection from the output SPIHT stream

The number of bits theoretically received up to the $i^{th}$ packet is calculated, which will give the packet loss rate as $1 - \frac{\text{number of bits received}}{\text{bit stream original length}}$. \(PSNR(P_i)\) can be found by performing linear interpolation on the PSNR versus packet loss rate curve.

### 4.2.3 Rest of System

Assume that the channel loss profile indicates that the transmission success probability for every packet to be 80%. The transmission criteria is that all packets should ideally transmit in one channel only, however each packet can be split up in many parts to transmit at once over multiple channels. Once the packets go through the lossy channels, the channel decoder is notified at the RB level whether the content in the RBs are lost over the channel, and if so, at which location in the frame. These information are useful at determining the faulty RS codes, which have more than the acceptable amount of packet losses, to discard them. If the code containing layer $i$ is discarded, no more decoding is further needed; the stream has to receive layer $i$ before it receives layer $i + 1$ for the purpose of the source decoder.
Chapter 5

Experimental Evaluation

5.1 Experimental Setup

The images used to test the system setup are 512 x 512 bitmap images of Lena, Goldhill and Barbara, as shown in Figures 5.1, 5.2, and 5.3 respectively. For SPIHT encoding, all of the SPIHT properties are selected so that the compression of the stream is the most effective: SPIHT utilizes the 9/7-tap biorthogonal wavelet filter, with 8 filtering levels and 11 bit planes to decode. The compression ratio is set to 8, which is equivalent to the bits per pixel rate being exactly 1.0. Number of packets ($S$) is set to 5, codeword length ($N$) for RS coding is set to 14. The primary network load is randomized from 0.16 to 0.64 using (4.7), from which the channel capacity is calculated and become part of the algorithm to solve the ULP FEC allocation problem.

5.1.1 ELP Test

In order to test the effectiveness of ULP, an ELP test was also used. Every possible code rate permutation is redesigned to have the same code rate across all layers. For example, a permutation of $\{1, 3, 5, 7, 9\}$ is redesigned to be $\{5, 5, 5, 5, 5\}$ if this
Figure 5.1: Original Lena image

Figure 5.2: Original Goldhill image
assignment is the ELP allocation with maximum $\overline{PSNR}$. At this point, all layers should have about the same amount of source packets, and therefore about the same amount of error correction packets. Each of the redesigned combinations, then, will produce a maximum $\overline{PSNR}$ value and the most optimal code rate assignment, which is then compared to the results obtained through ULP.

5.2 Results and Analysis

5.2.1 SPIHT Encoder

In order to measure the SPIHT coding tool’s effectiveness, a comparison was made between the coding efficiencies from the tool and from the original SPIHT work. Figure 5.4 illustrates the result of this comparison between these two SPIHT encoding processes: Figure 5.4a was obtained through the tool and contained a collection of values at different packet loss levels, while Figure 5.4b was obtained through the
original SPIHT work and contained a collection of different bits per pixel (bpp) rate. It could be seen that the performances are very similar in both case for both Lena and Goldhill images; however, the original SPIHT simulation results does not feature PSNR values at 0.2bpp or less. There is a significant dropoff in PSNR at around 95% packet loss rate. Overall, the similarity in terms of results for the SPIHT Encoder ensures that the bit stream is encoded correctly using the given tool.

Figure 5.4: Comparison of SPIHT encoding efficiencies: (a) with the tool (b) original code

Figure 5.5 again demonstrates that the tool performs as expected. The pattern of PSNRs is similar for both images, and both suffer a dropoff in quality near the 95% mark.

5.2.2 Channel Encoder

The system setup was simulated for three different images of Lena, Goldhill and Barbara, four different channel packet loss rates of 5%, 10%, 20% and 40%, five
different number of channels of 3, 6, 12, 24 and 48, and two different codeword length of 14 and 15. The varying channel packet loss rates were chosen such that the setup can be tested under varying packet loss conditions, with the most extreme condition being 40% channel packet loss rate. The number of channels were selected such that a varying amount of channel bandwidth was tested with the setup, and that at 48 channels, taking up three horizontal bands of a LTE frame, the channel bandwidth is large enough to accommodate all of the source bit stream. Because the primary network load is random, the secondary network throughput and the channel bandwidths are also random. Therefore, to accurately assess the results of the setup, 20 simulations were conducted, each with a randomized primary network load; the code rate selections as well as the PSNR results were averaged over these 20 runs.

a) Codeword length 14

Figure 5.6 shows the results of code rate selection for the Lena image, averaged after 20 simulations, with codeword length 14 and for both ULP and ELP frameworks. In this figure, there are five bar graphs for the varying number of channels. Each bar graph contains four groups of bars which represents the varying channel packet loss.
loss rate. Then each group has five bars representing the ULP rate selection for each of the five layers. For all groups in the graph, the ULP rate selection indicates the number of information packets included in the codeword of length 14. Each group also contains a horizontal line, indicating the ELP code rate selection.

It could be seen from Figure 5.6 that for all groups in the graph, the ULP rate selection is unequal for all layers: the rate is non-decreasing from the first layer to the last in each group, as is to be expected because higher loss rate will require larger error protection. In other words, the first layer is protected the most and no less than the second layer, the second layer is protected no less than the third layer, and so on. The reason is that as the bit stream from SPIHT encoding is embedded, layers from first to last have decreasing amount of importance, and furthermore, a layer can only be used for decoding if all the previous layers have been received with no errors. As indicated in (4.9), maximizing $\overline{PSNR}$ is to maximize $P_{i|0}(\pi)$ when possible, and maximizing $P_{i|0}(\pi)$ generally means increasing the protection for the more important layers to increase the probability of successfully decoding packets from these layers.

The ELP rate selection for all groups, compared to the ULP rate selection, tends to protect the first three layers less (higher code rates) and the last two layers more (lower code rates). This is an expected result from the ELP test detailed in section 5.1.1. For both ULP and ELP, within the same bar graph, increasing the channel packet loss rate would mean decreasing the coding rate. This is done to further protect the layers at the expense of sending less information through the channel, because losing important packets through the channel is worse than sending only a few of them through successfully, for decoding purposes. Given a fixed channel packet loss rate, the ULP rate selection largely remains the same while the ELP rate selection only decreases slightly as the number of channels increase. This, once again, can be attributed to the fact that the bit stream from SPIHT encoding is embedded; the ULP FEC allocation algorithm tries the best to protect the earlier part of the
stream, and therefore increasing the channel bandwidth to transmit only helps with transmitting the latter part of the stream, which has very little effect on the decoded image quality because of its lower importance. Both ULP and ELP code rate selection values in all groups, when rounded to the nearest integer, are mostly even due to the effect of padding.

Similar conclusions can also be drawn from Figures 5.7 and 5.8, which show the results of code rate selection for the Goldhill and Barbara images, averaged after 20 simulations, with codeword length 14 and for both ULP and ELP frameworks. For Goldhill, the ELP rate selection for all groups, compared to the ULP rate selection, tends to protect the first two layers less (rather than three for Lena) and the last three layers more (rather than two for Lena). This is possibly a result of the Goldhill image relying on less amount of important packets to decode than Lena, a consequence of different image compositions: one describes a scenery and the other describes a person’s face.

Figure 5.9 shows the $\text{PSNR}$ results for Lena, averaged after 20 simulations, with codeword length 14 and using the ULP framework. The benchmark line indicates the PSNR value of the Lena image when transmitted without errors. Each curve represents the $\text{PSNR}$ for a channel packet loss rate with varying number of channels. The $\text{PSNR}$ corresponds directly to the decoded image quality; the higher the $\text{PSNR}$ value, the better the decoded image quality is. For the test images in this thesis, the $\text{PSNR}$ is considered good, acceptable, or bad based on the values in Table 5.1. Good or benchmark $\text{PSNR}$ means that the decoded image quality is excellent; acceptable $\text{PSNR}$ means that despite some distortions, the decoded image is still recognizable; and bad $\text{PSNR}$ means that the decoded image is no longer recognizable.

It is clear from the figure that as channel packet loss rate increases, the $\text{PSNR}$ decreases. This can be explained by the fact that increasing channel packet loss
Figure 5.6: Lena’s average code rate selection results for ULP and ELP, codeword length 14.

Rate generally means $P_{i|0}(\pi)$ decreases, which from the calculation of $\overline{PSNR}$ in (4.9) indicates that the $\overline{PSNR}$ should be reduced. However the dropoff in $\overline{PSNR}$ from 20% channel packet loss rate to 40% is quite large, reducing the decoded image quality from acceptable to bad, compared to the dropoff in $\overline{PSNR}$ from 5% to 10% or 10%
Figure 5.7: Goldhill’s average code rate selection results for ULP and ELP, codeword length 14 to 20%. The 40% channel packet loss rate has a profound impact on the image transmission over the channel; the algorithm prefers to send only a few information packets than to send more and risk losing them, significantly reduces the PSNR over
such lossy conditions. It can also be seen that increasing the number of channels also improves the $\text{PSNR}$, especially from 3 channels to 6 as the increased bandwidth allows more transmissions of important packets in the image.
CHAPTER 5. EXPERIMENTAL EVALUATION

<table>
<thead>
<tr>
<th>Images</th>
<th>Benchmark</th>
<th>Good</th>
<th>Acceptable</th>
<th>Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>39.950</td>
<td>26 - 39.95</td>
<td>20-26</td>
<td>&lt;20</td>
</tr>
<tr>
<td>Goldhill</td>
<td>35.873</td>
<td>23 - 35.873</td>
<td>19-23</td>
<td>&lt;19</td>
</tr>
<tr>
<td>Barbara</td>
<td>35.820</td>
<td>23 - 35.82</td>
<td>19-23</td>
<td>&lt;19</td>
</tr>
</tbody>
</table>

Table 5.1: $PSNR$ values classification for the test images

Similar conclusions can also be drawn from Figures 5.10 and 5.11, which show the $PSNR$ results for the Goldhill and Barbara images, averaged after 20 simulations, with codeword length 14 and using the ULP framework.

![Figure 5.9: Lena’s average $PSNR$ results for ULP, codeword length 14](image)

Figure 5.12 show the decoded images of Lena under varying number of channels and varying channel packet loss rates, as a manifestation of Figure 5.9. All images in this figure have acceptable or better $PSNR$ values, as evidenced by how recognizable the Lena image is even when the channel packet loss rate is 20% and only 3 channels are available for transmission. At 24 channels or higher, the image quality is very good as there are no visible blurs and distortions over the image. Similar conclusions can be made from Figures 5.13 and 5.14 where the decoded images of Goldhill and Barbara under varying number of channels and varying channel packet loss rates are shown.
Figure 5.10: Goldhill’s average $\overline{PSNR}$ results for ULP, codeword length 14

Figure 5.11: Barbara’s average $\overline{PSNR}$ results for ULP, codeword length 14

as manifestations of Figure 5.10 and 5.11.

In terms of $\overline{PSNR}$ comparison between ELP and ULP, Figure 5.15 indicates that ULP is slightly better for the Lena image. The figure shows the $\overline{PSNR}$ values for
Figure 5.12: Lena’s decoded images using ULP with codeword length 14

Lena with codeword length 14 using the ULP framework, using the $\overline{PSNR}$ values obtained through the ELP framework as a reference. Each curve represents the $\overline{PSNR}$ difference for a channel packet loss rate with varying number of channels, where each point in the curve is calculated in this formula:

$$\overline{PSNR} \text{ difference} = \overline{PSNR} \text{ for ULP} - \overline{PSNR} \text{ for ELP}$$  \hspace{1cm} (5.1)

All of the relative values in Figure 5.15 are positive, showing the higher $\overline{PSNR}$ values for ULP over ELP under the same operating conditions. This can be explained by
Figure 5.6, in which ELP tends to protect more important layers less and less important layers more than ULP does, resulting in the $\overline{PSNR}$ value being smaller as $P_{i\mid 0}(\pi)$ is smaller for ELP due to the higher code rates chosen at the more important layers. This is also observed in Figures 5.16 and 5.17 for Goldhill and Barbara respectively, which confirms ULP’s superior performance over ELP for codeword length 14.

b) Codeword length 15
Figure 5.14: Barbara’s decoded images using ULP with codeword length 14

Figure 5.18 shows the results of code rate selection for the Lena image, averaged after 20 simulations, with codeword length 15 and for both ULP and ELP frameworks. The results are very similar to the codeword length 14 case, with the notable exception being the code rate selection not being non-decreasing at 40% channel packet loss rate and at high number of channels of 24 and 48. At these operating conditions, the second and third layers are protected more than the first; however because at 40% channel packet loss rate the image quality is bad enough for the image to be
unrecognizable, these results do not mean much with regards to successful decoding of the image. Both ULP and ELP code rate selection values in all groups from the bar graph, when rounded to the nearest integer, are mostly odd due to the effect of
CHAPTER 5. EXPERIMENTAL EVALUATION

Figure 5.17: Barbara's average PSNR results for ULP relative to ELP, codeword length 14 padding. Similar conclusions can also be drawn from Figures 5.19 and 5.20, which show the results of code rate selection for the Goldhill and Barbara images, averaged after 20 simulations, with codeword length 15 and for both ULP and ELP frameworks.

Figure 5.21 shows the PSNR results for Lena, averaged after 20 simulations, with codeword length 15 and using the ULP framework. It could be seen from the figure that except for the data points in the 40% channel packet loss rate curve, the difference in performance between codeword length 15 and 14 is small enough that the decoded images would look extremely similar in both cases. This is also observed in Figures 5.23 and 5.22 where the PSNR results for Barbara and Goldhill are shown.

In terms of PSNR comparison for ELP and ULP, Figure 5.24 indicates that ULP is slightly better for the Lena image. The figure shows the PSNR values for Lena with codeword length 15 using the ULP framework, using the PSNR values obtained through the ELP framework as a reference. With the exception of the data points in the 40% channel packet loss rate curve (where the decoded image quality is bad
Figure 5.18: Lena’s average code rate selection results for ULP and ELP, codeword length 15 anyway), all of the relative values in Figure 5.24 are positive, showing the higher PSNR values for ULP over ELP under the same operating conditions. This, once
Figure 5.19: Goldhill’s average code rate selection results for ULP and ELP, codeword length 15

again, can be explained by Figure 5.18, in which ELP tends to protect more important layers less and less important layers more than ULP does, resulting in the $\text{PSNR}$
Figure 5.20: Barbara’s average code rate selection results for ULP and ELP, codeword length 15

value being smaller as $P_{i|0}(\pi)$ is smaller for ELP due to the higher code rates chosen at the more important layers. This is also observed in Figures 5.25 and 5.26 for Goldhill
and Barbara respectively, which confirms ULP’s superior performance over ELP for codeword length 15 as well.
Figure 5.23: Barbara’s average $PSNR$ results for ULP, codeword length 15

Figure 5.24: Lena’s average $PSNR$ results for ULP relative to ELP, codeword length 15
CHAPTER 5. EXPERIMENTAL EVALUATION

Figure 5.25: Goldhill’s average $\overline{PSNR}$ results for ULP relative to ELP, codeword length 15.

Figure 5.26: Barbara’s average $\overline{PSNR}$ results for ULP relative to ELP, codeword length 15.
Chapter 6

Conclusions & Future Work

The ever increasing demands for spectrum access and the growing amount of multimedia communication needs will only increase the usage of CRs in wireless networks for multimedia transmissions as means to dynamically access the radio spectrum while operating with the best possible throughput amount. This work proposed an end-to-end JSCC system setup for transmissions of images over a lossy multichannel CR network operating with the underlay DSA paradigm. The system contains a pair of jointly-configured source-channel codecs: the source encoder is in the form of SPIHT, while the channel encoder processes information about the bit stream and the channel characteristic so that it can assign packets the appropriate amount of FEC bits to maximize the expected PSNR of the output image. Theoretical analysis of the code rate selection process is supported by the practical simulations of an image transmission under this system.

For future work, a new algorithm should be devised to quickly search for the optimal FEC assignment without sacrificing accuracy. The used exhaustive search algorithm has sizable performance drawbacks, especially if the number of bit stream partitions increases; although the algorithm did provide the best performance that could be achieved with the system. The overall system setup can be further modified to ac-
commodate more types of source information, such as audio and video. Finally, other channel coding schemes such as Luby Codes or Rate-Compatible Punctured Convolutional Codes are also viable under the JSCC coding scheme in this work.
Bibliography


