Improving the Bandwidth Efficiency of Multiple Access Channels using Network Coding and Successive Decoding

Mohammad Jabbari Hagh

A Thesis
In the Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy at
Concordia University
Montreal, Quebec, Canada

February 2013

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Mohammad Jabbari Hagh**

Entitled: **Improving the Bandwidth Efficiency of Multiple Access Channels using Network Coding and Successive Decoding**

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Electrical and Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. G. J. Gouw

_____ External Examiner
Dr. S. Valaee

_____ External to Program
Dr. H. Harutyunyan

_____ Examiner
Dr. W.E. Lynch

_____ Examiner
Dr. Y.R. Shayan

_____ Thesis Supervisor
Dr. M.R. Soleymani

Approved by _____
Dr. J.X. Zhang, Graduate Program Director

February 2013

_____
Dr. Robin Drew, Dean
Faculty of Engineering and Computer Science

Improving the Bandwidth Efficiency of Multiple Access Channels Using Network Coding and Successive Decoding

Mohammad Jabbari Hagh, Ph.D.
Concordia University, 2013

In this thesis, different approaches for improving the bandwidth efficiency of Multiple Access Channels (MAC) have been proposed. Such improvements can be achieved with methods that use network coding, or with methods that implement successive decoding. Both of these two methods have been discussed here.

Under the first method, two novel schemes for using network coding in cooperative networks have been proposed. In the first scheme, network coding generates some redundancy in addition to the redundancy that is generated by the channel code. These redundancies are used in an iterative decoding system at the destination. In the second scheme, the output of the channel encoder in each source node is shortened and transmitted. The relay, by use of the network code, sends a compressed version of the parts missing from the original transmission. This facilitates the decoding procedure at the destination. Simulation based optimizations have been developed. The results indicate that in the case of sources with non-identical power levels, both scenarios outperform the non-relay case.

The second method, involves a scheme to increase the channel capacity of an existing channel. This increase is made possible by the introduction of a new Raptor coded interfering channel to an existing channel. Through successive decoding at the destination, the data of both main and interfering sources is decoded.

We will demonstrate that when some power difference exists, there is a tradeoff between achieved rate and power efficiency. We will also find the optimum power allocation scenario for this tradeoff. Ultimately we propose a power adaptation scheme that allocates the optimal power to the interfering channel based on an estimation of the main channel's condition.

Finally, we generalize our work to allow the possibility of decoding either the secondary source data or the main source data first. We will investigate the performance and delay for each decoding scheme. Since the channels are non-orthogonal, it is possible that for some power allocation scenarios, constellation points get erased. To address this problem we use constellation rotation. The constellation map of the secondary source is rotated to increase the average distance between the points in the constellation (resulting from the superposition of the main and interfering sources constellation.) We will also determine the optimum constellation rotation angle for the interfering source analytically and confirm it with simulations.

To my parents, my sister,
and the Atlantic Ocean.

Acknowledgments

First and foremost, I would like to express my gratitude to my supervisor, Professor M. Reza Soleymani, for giving me the opportunity to work in his group and for his support and encouragement throughout my research not only as a professor but also as a friend. Without his guidance and instruction, I would have not been able to complete this research.

I would like to thank all members of my Ph.D. defense committee: Prof. William E. Lynch, Prof. Hovhannes A. Harutyunyan and Prof. Yousef R. Shayan. I appreciate their valuable comments and all the time they spent reading this thesis. My especial thanks go to the external examiner, Prof. Shahrokh Valaee, of University of Toronto, for his technical insight. I also want to thank all current and previous members of the Wireless and Satellite Communications Laboratory for their help and support. In addition, I wish to thank NSERC CRD as well as InterDigital Canada Ltd. and PROMPT.

More than anyone else, I would like to thank my parents, Ahad and Marjan. I would not be where I am today without their love, encouragement and unlimited sacrifices. I only hope I have made their support worthwhile.

My thanks also go to my beloved sister, Mahsa, for her support.

Contents

List of Figures

List of Tables

List of Abbreviation

ACM	Adaptive Coding and Modulation
AF	Amplify and Forward
APSK	Amplitude and Phase Shift Keying
ATM	Asynchronous Transfer Mode
AWGN	Additive White Gaussian Noise
BEC	Binary Erasure Channel
BER	Bit Error Rate
BP	Believe Propagation
BSC	Binary Symmetric Channel
CC	Constellation Constrained
CDMA	Code Division Multiple Access
CF	Compress and Forward
CoC	Convolutional code
CRC	Cyclic Redundancy Check
CSCG	Circular Symmetric Complex Gaussian
CSI	Channel State Information
CTP	Constant Total Power
DF	Decode and Forward
DVB-RCS	Digital Video Broadcasting-Return Channel via Satellite
DVB-S	Digital Video Broadcasting-Satellite
FDMA	Frequency Division Multiple Access

FEC	Forward Error Coding
FER	Frame Error Rate
IP	Internet Protocol
ISF	Interfering Source First
LDPC	Low Density Parity Check
LLR	Log-Likelihood Ratio
LT	Luby Transform code
MAC	Multiple Access Channel
MAP	Maximum a posteriori
MARC	Multiple Access Relay Channel
MF-TDMA	Multi-Frequency Time Division Multiple Access
MIMO	Multiple-Input Multiple-Output
MMSE	Minimum Mean Square Error
MPEG	Moving Picture Experts Group
MSF	Main Source First
NP-hard	Nondeterministic Polynomial-time hard
PNC	Physical layer Network Coding
PSK	Phase Shift Keying
RS	Reed Solomon
SNR	Signal to Noise Ratio
SRP	Separate Relay Power
TDMA	Time Division Multiple Access

List of Symbols

α	Power allocation ratio
β	Power level ratio
γ	Pass loss exponent
ϵ	Raptor overhead
Δ	Actual and analytic power gap
λ	LLR value for PSK bits
$\Lambda(x)$	LDPC degree distribution
σ	Gaussian noise standard deviation
θ^*	Analytical optimum rotation angle
θ_{opt}	Optimum rotation angle
$\bar{\theta}_{opt}$	Average optimum rotation angle
ω	Confidence coefficient
$\Omega(x)$	LT degree distribution
b	MPSK symbol
$\mathcal{C}$	Raptor pre-code
C	Channel capacity
d	Distance between constellation points
D	Distance between constellation points with least distance
$\bar{D}$	Average D over α values
D_i	Decision Region for symbol i
E_s	Energy per symbol

$\mathbb{F}$	Finite field
I_{LDPC}	BP iteration number for LDPC
I_{LT}	BP iteration number for LT
I_{Turbo}	Turbo iteration number
k	length of source message
L_c	Channel reliability value
$m_{i,o}$	Message passed from input to output nodes in BP
$m_{o,i}$	Message passed from output to input nodes in BP
n	Codeword length
P_d	Power difference
P_{er}	Erasure probability
T	Decision region borderline
$x(i)$	constellation point for symbol i
X	MARC channel inputs
y	Received signal at the destination
Y	MARC channel outputs
W	MARC source number
z	Circular symmetric complex gaussian noise
Z_0	Channel output LLR

Chapter 1

Introduction

1.1 Problem Statement

One of the main challenges in the telecommunications industry is achieving high rates with high bandwidth efficiency. In most cases, a bandwidth efficiency increase will result in more complex decoding methods. However, based on the most widely used methods, required bandwidth may or may not be increased. In recent years, numerous methods have been proposed in literature using increased or fixed bandwidth. One popular method that requires a bandwidth increase is the use of relays. If a relay uses an orthogonal channel, both bandwidth and decoding complexity will increase.

Relays help sources and destinations to exchange information more reliably, and make transmissions more cost-efficient. During recent years, from a processing point of view, relays were mostly passive. In other words, the relays simply repeated what they received from the sources. In the best case, they decoded data and re-encoded it before sending it on to the next relay or destination node.

Recently, a high-layer coding method called "network coding" has been introduced. The main idea behind network coding is to actively participate in routing data from the sources to the destinations. With network coding, relays can add data that they have gathered from different sources and send data combinations to the next layer of the network. It has been shown that network coding can achieve ca-

pacities that were not achievable with traditional routing methods. Network coding has moved network data distribution systems one step ahead and makes use of their potential capacity.

Yet network coding can also send extra redundancy to the destination node. This redundancy or extra parity will help the destination nodes to decode data more efficiently. Even in cooperative scenarios where source nodes act as relay nodes for their partners, network coding can be used to combine local and external data and transmit them as one package.

As mentioned above, another method for increasing bandwidth efficiency is to increase the rate of a fixed bandwidth, although some performance may be lost. One way of improving performance can be finding methods to exploit an existing channel between the sources and the destination via other interfering sources. In this method, the new interfering source should use the bandwidth available to the original source, and should transmit data so that the data streams from both the original and the new source can be decoded successfully at the destination. Since we are adding a new interfering source to the existing link, it would be practical to put a new condition on the system: that the original transmitter should remain intact and ignorant of the new source. This means that the original transmitter should not change anything in its encoding and transmission stages. This new condition will allow the scheme to apply to any existing multicast network without updating its numerous source nodes.

In this thesis, we will examine both of the above methods. We will use relays and network coding to improve the decoding performance of a two-source, one-destination network. This method increases the bandwidth requirement of the system. Later, we will use another method based on successive decoding for another two-source and one-destination network. We will introduce an interfering source to an existing link between a main source and a destination, and transmit data to the destination so that the data stream from each source can be decoded successfully. This will increase

our rate, yet the bandwidth requirement of the system will not increase.

1.2 Literature Review

1.2.1 Joint Decoding

The joint decoding of network codes with channel or source codes has been extensively discussed in literature. The framework of joint decoding using network codes was investigated in [1]. The authors demonstrate source-channel-network separation while codes are linear and examine the joint design of source-channel-network codes.

Later, several papers exploited this possibility and presented remarkable codes and strategies. In [2], iterative decoding for channel and network codes was introduced. In a two-way channel network each user iteratively decodes gathered data from the other sources and the relay. Turbo code is suggested as the channel code. Together, the channel codes of each user and the network code form a distributed network code which can be iteratively decoded. Lower bounds were derived as well. In another iterative decoding scheme, Hausl et al. [3] used Low Density Parity Check (LDPC) codes for the iterative decoding of network-coded data from two sources and one relay at the destination.

The network coding part is adapted to tanner graphs and one Belief Propagation (BP) decoder can iteratively decode two codewords simultaneously.

Sarshar et al. [4] take up the problem of broadcasting data from one source to multiple destinations. They propose a joint network-source coding scheme which is based on description codes and through this method they increase transmission diversity. They find optimum routings to send different descriptions of the code, which will finally be combined at each destination to get the full code.

In [5], Kliewer and Ho suggested use of Nested codes as a means of joining network codes and channel codes. Nested codes are designed for multi-source, multi-

destination networks with one relay. Here, the relay multiplies data received from each source by a unique generator polynomial, combines the data, and broadcasts it. It has been shown that this method will result in fewer relay transmissions, if feedback is allowed. Otherwise, it assures higher reliability.

In [6] a network coding interpretation of network topologies was suggested. Here, Bao et al. considered a cooperative network in which each node at its second phase works as a relay for others nodes and sends a network code composed of received codewords. Inspired by BP decoding, they used a tanner graph decoding method to verify network codes. The work in [6] was based on lower-triangular LDPC codes, and this research was extended to LDPC codes in [7]. The outage probability involved in this method was discussed in detail in [8]. Later, in [9], joint channel-network coding was introduced. The authors used LDPC codes as channel codes, and by extending the parity check code of the channel code, they managed to jointly decode channel and network codes with low complexity. They employ network codes as a generalization of channel codes.

1.2.2 Estimation-Based Network Coding

Physical layer network coding (PNC) has been discussed in [10], [11] and [12]. In these papers, Zang et al. argued that in wireless networks, while the relay receives a combination of data from the sources, it does not need to extract data from each source and again combine the network codes for relaying. They proposed the Channel-decoding- Network-Coding process and focused on extracting network codes from received signals without any decoding. In [13] they discussed the effect of finite field and infinite field network coding on their proposed scheme.

Joint Network Coding and Superposition Coding (JNSC) is a scheme based on an information exchange loop. It was proposed in [14]. For a four-node network with one relay, this joint coding scheme uses iterative decoders for decoding data received

from information exchange loops. Several topologies for these loops were suggested, and it was shown that in some cases JNSC outperforms pure time division and pure network coding schemes.

In [15] a new decoding and encoding scheme for relays was proposed. On two-way channels where channels from the sources to the relay are noisy and the cost criterion is source distortion, the normal Maximum *a posteriori* (MAP) decoding and network coding is suboptimal. A new method of joint decoding/estimation and network coding has been introduced where decoding is based on the Minimum Mean Square Error (MMSE) estimate.

An algebraic superposition-based relaying system was proposed in [16]. In this cooperative scenario, nodes also work as relays and each node sends an algebraic superposition of its locally generated data, as well as the relayed data of its neighbor node. Considering the fact that the relayed data from its neighbor is in fact its own local data each node can exploit the data of its neighbor and use it in its next codeword, which will be generated by the network coding (algebraic superposition) of the local data and decoded data.

The correlation between the sources in frequency-selective fast fading channels was studied by Ser et al [17]. They introduced an iterative, distributed source-channel network decoding method that benefits from correlation and from the frequency selectivity of the channel. Another iterative joint channel-network coding was discussed in [18] In this scheme turbo codes serve as channel codes, and the relay derives the Log-Likelihood Ratio (LLR) of the network code from its inputs without decoding. At the destination this LLR will be used as *a priori* data for the iterative turbo decoder.

1.2.3 Network Coding over Relay Strategies and Parity Generation

The performance of network coding over different strategies such as Decode and Forward (DF) and Amplify and Forward (AF) is discussed extensively in the literature. An alternative method to DF relays was suggested in [19]. As a result of the use of two-way channels, Bi-directional Amplification of Throughput (BAT-relaying) was introduced.

BAT-relaying combines the data gathered from the sources; the network codes this data and broadcasts it. DF BAT-relaying was previously presented in literature however; AF BAT-relaying was not. It is based on inherent pocket combining, which is facilitated by the simultaneous employment of multiple access channels. Both schemes were examined and it was shown that, in noiseless channels, BAT-relaying with AF is superior to DF.

In [20] the outage probability and coverage area of networks with noisy source relay channels are discussed. The authors make a comparison between no-coding and network coding scenarios. This comparison shows that when the quality of a source-relay channel is high, the no-coding case outperforms network coding. However, in noisy channels, network coding is more reliable. Laneman [21] compared network-coded AF and repetition DF in networks with different geometries. Outage probability was used as a basis for comparison, and it was concluded that in many cases, especially when the relay is close to the source nodes, AF outperforms repetition DF.

Pepovsky and Yomo [22] investigated all known schemes for use of network coding in wireless communications and derived achievable rates for these schemes. For DF, AF, and Joint Decode-and-Forward (JDF), achievable rates were found. For Denoiseand- Forward (DNF), an upper bound for the achievable rate was determined. It has been shown that the achievable rate for JDF achieves upper bound of DNF in some instances. In [23] authors presented two network coding based scenarios for

relay cooperation, one based on iterative decoding and another one with two phase collaborative decoding.

Parity generation in relays is one of the methods which have also been studied in detail. Kim [24] devised a concatenated random parity forwarding technique for multi-hop wireless networks. These networks use multiple relays to transmit data from multiple sources to their destination. In this arrangement each relay adds additional parity to its collected data and sends it to the next relay. This additional parity is generated through network coding. Each relay combines its collected codewords to get this new codeword as an additional parity. While this method obviously decreases the rate as the relays get closer to destination, it can also use a lower energy level for transmitting data for the same reason.

Coded cooperation based on space-time transmission, and turbo coding was first discussed in [25]. The authors proposed two schemes, the first of which uses space-time transmission.

Here each of two nodes at the first phase sends its own data only. In the second phase each node along its own parity, sends parity bits of its partner with space time coding. Hence a gain in diversity is achieved. In the first phase of turbo coding cooperation each node sends the symmetric data along the output of the first parity encoder while ignoring the output of the second decoder. At the second phase each node tries to decode the data of its partner, and then re-encode it to send the output of the second parity encoder of its partner.

Progressive network coding is a scheme proposed by Bao and Li [26] in order to conserve bandwidth. In this scheme, which involves a chain of nodes and unlike repetition and forward scheme, each intermediate node re-encodes the data after decoding it and transmits a subcodeword. The destination receives all these codewords and uses an iterative decoder, as if it is dealing with a giant parallel concatenated (network) code.

1.2.4 Protocols and Soft Output Relays

In more practical works several protocols were suggested. Kuek et al. [27] studied four node networks with two-way channels and, instead of just combining data in nodes, used pre-cancellation and eliminate prior messages. Ultimately, they suggested a protocol for networks with even number of nodes.

Opportunistic methods were discussed in [28]. In an Opportunistic Network Coded Cooperation (ONCC) scheme, the relay decides whether it should help one source or two sources with the help of network coding. It has been demonstrated that this method achieves an optimal diversity-multiplexing tradeoff. The authors then extend their work in [29] to a protocol called Selective Network-Coded Cooperation (SNCC). In SNCC a relay node implements network coding based on reliable information (i.e. if the source-relay channel is of high quality they use its data in their network code). It has been shown that implementing this protocol in the networks with several sources, destinations, and relays avoids error propagation and achieves better diversity-multiplexing than other protocols in literature. In an independent Yomo and Popovsky [30] take up the same problem, discuss scheduling strategies, and analyze the average capacity of such a network.

In [31] three different protocols for scheduling data transition in a two -way channel are discussed, and their bounds and achievable rates have been derived for them. In the first protocol both sources transmit simultaneously and relay broadcasts at the second phase. In the second protocol each source has its own time slot for broadcasting, and the third protocol is a hybrid of the two previous protocols. Here, some time slots for both sources transmit and in some just one of them transmits. It has been shown that in some situations the hybrid case achieves higher rates than the first two protocols. Later Koike et al. [32] proposed adaptive coding for the hybrid protocol based on channel state information (CSI) that resulted from adapting coding and signal constellation. Later they extended their work to frequency-selective fast

fading channels [33].

In [34], in order to overcome the half-duplex nature of relay transitions, Bi-directional Successive Relaying based on Physical Network Coding (PNC-BSR) was proposed. The main idea here involves using two relays such a way that, while one of them receives data, the other one transmits. Each relay combines the data received in the previous phase from its partner relay and two sources and sends this gathered data. Through this method, the destination receives three network-coded copies of each codeword. It has been shown that this method improves the throughput. Lv and Yu [35] extended PNC-BSR to the Successive Relaying Scheme (SRS) with multiple relays and multiple sources and destinations. Relay nodes are paired to change half-duplex connections to full-duplex ones. As a result, SRS can improve spectral efficiency.

In [36] the effect of network coding on different scenarios of multicast in a IEEE 802.11 multi-hop wireless network was investigated. More generally, Jin et al. [37] compared network coding and non-network coding scenarios on multi-hop networks on networks with or without fading in different topologies. Sagduyu and Ephremides [38] and [39] investigated the joint implementation of network coding and MAC in wireless ad hoc networks. Hamra and Turletti [40] addressed the use of network coding on mesh networks. It was shown that in applications such as file sharing over wireless mesh networks some improvements are achieved. However, in some cases, traditional repetition schemes perform better. Katti et al. [41] also used network coding in intermediate nodes of mesh networks and introduced a new architecture named COPE, which improves overall network throughput.

In their investigation of networks with changing topologies, Fragouli et al. [42] proved that network coding will logarithmically aid energy consumption. They support their theory with several investigations of ad hoc networks and cellular networks. In [43] an algorithm based on the Ford-Fulkerson algorithm is proposed in order to

achieve maximum flow in these networks.

As for soft output relays, in [44] a soft value-based decoding named "continuous network coding" is proposed. In this procedure, instead of 0-1 bits, an *a posteriori* probability is taken into consideration, and through this change the chance of error propagation is reduced. The data is not decoded completely at the relay and soft information is relayed, which results in better performance compared to hard decision relays.

Soft output relay was discussed by Yang et al. in [45]. Under this arrangement, the relay gets data from two sources, soft decodes them, and calculates the LLR of the network code based on the LLR derived from each decoder. Then, through an analog channel, the LLRs are transmitted to the destination. Here, an iterative decoder gets data from both sources, decodes them, and after each iteration compares them with the LLRs received from the relay. This method is called joint channel-network coding.

In order to overcome diversity inefficiency in network coding, Wang et al. [46] developed complex field network coding. Through the implementation of this method, a network with more than two sources can achieve full diversity regardless of its Signal to Noise Ratio (SNR).

Multiple Input Multiple Output (MIMO) techniques and network coding were both used by Fasolo et al. [47]. These techniques transmit network-coded data derived from buffers and send it over MIMO channels. At the destination a joint MIMO-network code decoder is used. The idea is to lower network coding to the physical layer introduced by [10]. The decoder is MIMO H matrix with matrix interpretation of network coding.

Fountain codes were used in work of Wicaksana et al. [48]. In their scheme relay just uses Cyclic Redundancy Check (CRC) to check the incoming data and XORs CRC passed codewords with previously saved correct codewords. Just one fountain decoder is needed at the destination. Tight upper bounds have also been . Chen et

al. [49] investigated the effect of network coding on Distributed Antenna Systems (DAS). It is shown that this method results in better diversity with lower cost and more spectral efficiency.

1.2.5 Cooperation in Multiple Access Channels

In recent years, the use of rateless codes has been proposed for orthogonal and non-orthogonal MAC. Kurniawan et al. [50] presented a network coding-based approach for cooperation through relay with rateless codes. In their scheme, each codeword contains partial data from its following codeword, and through this method they reduced the complexity of their method compared to other rateless cooperative methods. Vellambi [51] used rateless codes for multihop wireless networks. In this scheme, each node sends a fraction of the codewords required for successful decoding to the destination node. Through this method, energy consumption is reduced without any loss of reliability.

In [52] iterative decoding was used for a MAC cooperative scheme to decode transmitted messages. Yang and Madsen [53] used rateless codes in low power regime cooperation on MAC to achieve near-optimal performances. This work was extended by Uppal et al. in [54] with multiplexed rateless coding. It has been determined that the performance of their proposed scheme in full duplex and half duplex modes are very close. Later, in [55], the authors presented a rateless protocol for the half duplex case using feedback from the destination and a combination of DF and Compress and Forward (CF) schemes.

Bursalioglu et al. [56] investigated lossy multicast over binary symmetric broadcast channels. They used optimization for concatenated refinement source coding with channel coding. Due to the fact that their scheme required a vast range of rates, Raptor code was chosen as the channel code.

For the channels with side information, where the destination node contains some

information from the source data before the transmission, Sejdinovic et al. [57] used Raptor codes to address the problem. Gong et al. [58] introduced two physical layer approaches - one based on superposition coding, and the other based on Raptor code - for joint decoding of a relay-aided transmission. They optimized both cases based on the EXtrinsic Information Transfer (EXIT) function analysis. For the same classic relay case, Ravanshid et al. [59] proposed a mixed combining scheme for signal combination in the relays using rateless codes. This mixed combining scheme is a combination of the energy combining and information combining methods previously proposed in the literature.

Hagh et al. [60] and [61] studied the addition of an interfering source with a non-orthogonal channel with Raptor code. They also presented a power adaptation method in [62] in order to find the optimal power level for the interfering channel. Gong et al. [63] proposed layered coding for multiple-source interfering channels. They used a group decoder with successive decoding and optimized the rate allocation for different layers of code using Raptor code.

1.2.6 Forward Error Coding on DVB

The Forward Error Coding (FEC) for Digital Video Broadcasting (DVB) standards is still studied extensively. Papaharalabos et al. [64] studied and compared the 3rd Generation Partnership Project 2 (3GPP2) turbo code with other optimized turbo codes, LDPC codes, and Rate Compatible Irregular Repeat-Accumulate (RC-IRA). They concluded that the Consultative Committee for Space Data Systems (CCSDS) turbo codes perform better than the 3GPP2 turbo code. As for LDPC FEC codes, in [65] the tradeoff between complexity and performance for sub-optimal decoding process was investigated.

Recently, Raptor codes have been introduced in DVB FEC (mostly in the application layer) due to their near-optimal performance. Mladenov et al. [66] proposed an

incremental enhanced Gaussian elimination (IEGE) decoding algorithm for Raptor codes, which was used at IP Datacast over DVB services.

Luby et al. [67] proposed an exact physical channel model for the use of Raptor codes in a universal mobile telecommunications system (UMTS), and investigated the rate and other settings for the Raptor code and Turbo codes present in the DVB standard.

1.3 Thesis Outline

So far, we have reviewed the literature on relay cooperation with network coding, cooperative MAC and MARC networks, and FEC on DVB systems. The rest of this thesis is organized as follows:

In Chapter 2, we review the background information on network coding, MARC, Raptor codes, and DVB standards.

In Chapter 3, we discuss the use of network coding in MARC networks. In MARC networks one or more relays aid data flow from source nodes to destination nodes. Network coding is a new method in network layering that provides the possibility of making combinations in the relay node. Hence, the relay nodes are able to influence data flow in a network.

Under this method two schemes are proposed for the employment of network coding in MARC networks. In the first scheme, in addition to the redundancy that is generated by the channel code, additional redundancy is generated by the network coding. These redundancies will be used in an iterative decoding system to decode the original messages broadcasted by the source nodes. In the second scheme, the output of the channel encoder in each source node is shortened and transmitted. The relay uses the network coding to send a compressed version of the missing parts of the original transmission, which facilitates the decoding procedure for the destination.

In Chapter 4, we propose a scheme for increasing the channel capacity of an existing channel. This increase is made possible by the introduction of a new interfering channel to an existing DVB channel. The interfering channel uses Raptor code. Through successive decoding at the destination, the data of the main and interfering sources is decoded. We examine the case of sources with equal power transmission levels. However, as in all MAC detection methods, there should be a power difference between the two sources in order to achieve higher rates.

We will then demonstrate that when the power difference exists, there is a tradeoff between the achieved rate and the power efficiency, and we will find the optimum power allocation scenario for this tradeoff. We have proposed a power adaptation scheme that allocates the optimal power to the interfering channel based on an estimate of the main channel's condition. This estimate is obtained from the amount of overhead required by the destination for the successful decoding of the message. Therefore, the interfering source is able to adapt itself to the system without having any access to the CSI of the main channel.

In Chapter 5, we expand upon our work to raise possibility of decoding either the secondary source data or the main source data first. We will investigate the performance and delay for each decoding scheme. Since the channels are non-orthogonal, it is possible that, for some power allocation scenarios, constellation points get erased. To address this problem, we use constellation rotation.

The constellation map of the secondary source is rotated to increase the average distance between the points in the constellation resulting from superposition of the main and interfering sources' constellations. We analytically determine the optimum constellation rotation angle for the interfering source and confirm it with simulations.

Finally, in Chapter 6, we conclude our work and offer suggestions for future research.

Chapter 2

Background

2.1 Introduction

In this chapter, we will review some background concepts which will be used in later chapters. First, we will have a look at network coding and MARC networks (Their joint use is discussed in Chapter 3.) Later, we will explain Raptor code basics and the DVB standards that are the foundation of the proposed schemes in Chapters 4 and 5.

2.2 Network Coding

Consider an acyclic directed graph with a set of nodes that includes the source nodes, the intermediate nodes, and the sink nodes; and a set of edges, which are directed, error-free, and can transmit one symbol in each transmission. The task is to multi-cast common information from the source nodes to the sink nodes through the intermediate nodes. This means that the source nodes multicast symbols and the intermediate nodes transmit them to the sink nodes by simply forwarding them or in some cases by creating a new code from their inputs, alternatively referred to as network coding.

The transmitting symbols are the elements of a finite element field $\mathbb{F}$ which is

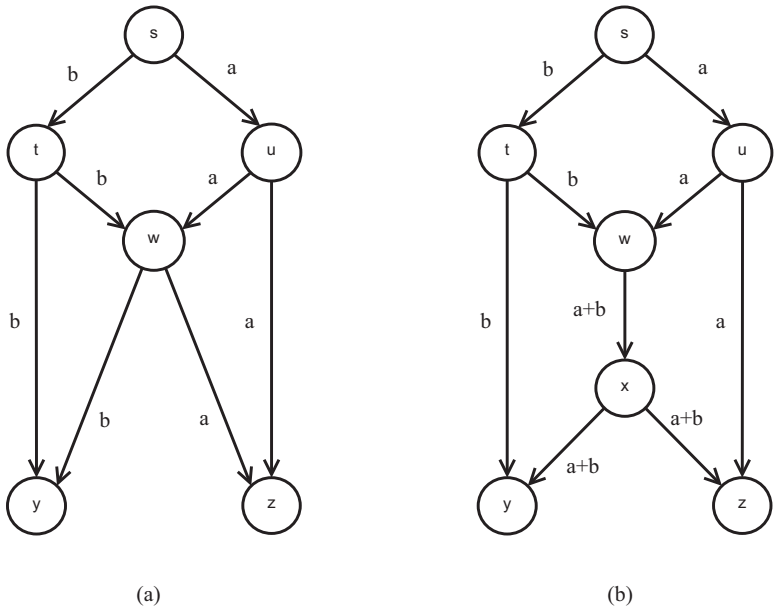

Figure 2.1: Networks with multicasts from s to y and z

selected with respect to the number of network sinks [68]. The network capacity is given by the Max-Flow Min-Cut Theorem [69] and indicates the maximum number of simultaneously transmittable symbols from the source nodes to the sink nodes.

We can explain the use of network coding with a simple example. Consider the network in Fig. 2.1(a), source s can multicast symbols a and b to sinks y and z easily. Node w forwards symbol a to sink z and symbol b to y. But in Fig. 2.1(b) it is impossible to multicast the symbols a and b to the sink nodes without using network coding. To solve this problem using coding it is sufficient to exclusive-OR (XOR) a and b in the node w and transmit it to the sinks through node x. In each of the sinks we have a or b and $a \oplus b$, so both symbols are decodable.

2.2.1 Max-Flow Min-Cut Theorem

Ahlswede *et al.* [69] showed that the network capacity given by the Max-Flow Min-Cut theorem is achievable with network coding. Consider $\mathcal{G}(\mathcal{V}, \mathcal{E})$ to be a directed graph with node set $\mathcal{V}$ and edge set $\mathcal{E}$. The network has a source node s and L sink

nodes $t_1, t_2, ...t_L$. The capacity and flow of each edge between nodes i and j, $(i,j) \in \mathcal{E}$ is shown by $\mathcal{C}_{ij}$ and $\mathcal{F}_{ij}$ while $0 \leq \mathcal{F}_{ij} \leq \mathcal{C}_{ij}$. A subgraph $\mathcal{G}_l$ from s to t_l $(l = 1, ..., L)$ is defined as $\mathcal{G}_l(\mathcal{V}, \mathcal{E}_l)$ where:

$$\mathcal{E}_l = \{(i,j) \in \mathcal{E} : (i,j) \text{ is on a directed path from } s \text{ to } t_l\} \tag{2.1}$$

A flow $\mathcal{F}$ is defined as $\mathcal{F} = \{\mathcal{F}_{ij} : (i,j) \in \mathcal{E}_l\}$ and starts from s and ends in t_l in $\mathcal{G}$ such that for $i \in \mathcal{V} \backslash \{s, t_l\}$:

$$\sum_{k:(k,i)\in\mathcal{E}} \mathcal{F}_{ki} = \sum_{j:(i,j)\in\mathcal{E}} \mathcal{F}_{ij} \tag{2.2}$$

In other words, the incoming and outgoing flow should be equal for all nodes, except for the source and sink nodes.

The value of each flow is determined as follows:

$$|\mathcal{F}| = \sum_{j:(s,j)\in\mathcal{E}} \mathcal{F}_{sj} - \sum_{i:(i,s)\in\mathcal{E}} \mathcal{F}_{is} = \sum_{i:(i,t_l)\in\mathcal{E}} \mathcal{F}_{it_l} - \sum_{j:(t_l,j)\in\mathcal{E}} \mathcal{F}_{t_l j} \tag{2.3}$$

Flow $\mathcal{F}$ is max-flow if $\mathcal{F}$ is a flow from s to t_l and its value is larger than any other flow from s to t_l. In one-source, multiple-sink graphs, the value of this max-flow $\mathcal{F}$ is the capacity of the graph.

A cut divides the node set $\mathcal{V}$ to two sets $\mathcal{S}$ and $\mathcal{T}$ in a way that $s \in \mathcal{S}$ and $t_l \in \mathcal{T}$ for $l = 1, 2, ..., L$. The number of all possible cuts in graph $\mathcal{G}$ is equal to $2^{|\mathcal{V}|-2}$. The capacity of a cut $\mathcal{C}_{\mathcal{S}\mathcal{T}}$ is defined as:

$$\mathcal{C}_{\mathcal{S}\mathcal{T}} = \sum_{i\in\mathcal{S}, j\in\mathcal{T}, (i,j)\in\mathcal{E}} \mathcal{C}_{ij} \tag{2.4}$$

A min-cut is a cut in the graph that among all other possible cuts, it has the lowest capacity. For example, in Fig. 2.2, a possible cut can be $\mathcal{S} = \{s, a, b, d\}$ and

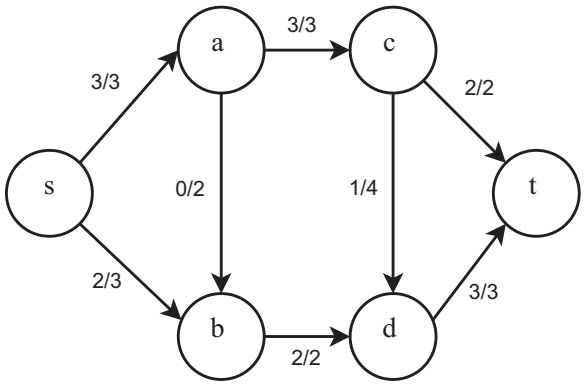

Figure 2.2: Edge capacities and flow for a single sink graph, each fraction shows $\mathcal{F}_{ij}/\mathcal{C}_{ij}$

$\mathcal{T} = \{c, t\}$. The capacity of this cut is 6. However, for another cut $\mathcal{S} = \{s, a, b\}$ and $\mathcal{T} = \{c, d, t\}$ the capacity will be 5, and this is the minimum capacity of all other cuts too. Hence, the min-cut capacity is 5.

The Max-Flow Min-Cut theorem states that the max-flow is equal to the capacity of the min-cut. Hence, the max-flow of the graph in Fig. 2.2 will be also 5, which can be seen from the combination of outgoing flows from s or incoming flows to t.

2.2.2 Linear Coding

Li *et al.* [70] demonstrated that linear network coding can be used to multicast symbols at a rate equal to the network capacity. Koetter and Medard [71] introduced an algebraic framework for linear network coding and presented a polynomial time algorithm to verify a constructed network code. Ho *et al.* [68] used this framework to show that linear network codes can be efficiently constructed by employing a randomized algorithm.

Assume the general case of a K source graph, where $\mathcal{X}_{s_i}$ is the source process for source s_i $(i = 1, 2, .., K)$ and $\mathcal{Y}_j$ is the output random process of node j. The intermediate node output is a linear combination of the inputs to the node j from

source processes and the output random processes of other nodes:

$$\mathcal{Y}_j = \sum_{i:(i,j)\in\mathcal{E}} a_{ij}\mathcal{X}_{s_i} + \sum_{k:(k,j)\in\mathcal{E}} f_{kj}\mathcal{Y}_k \tag{2.5}$$

and the output process at destination node t_i regarding source s_j is as follows:

$$\mathcal{Z}_{t_ij} = \sum_{k:(k,t_i)\in\mathcal{E}} b_{ijk}\mathcal{Y}_k \tag{2.6}$$

where $\{a_{ij}, f_{ij}, b_{ijk} \in \mathbb{F}_{2^u}\}$ are sequences of length-u bit vectors. These coefficients can be shown with matrices $A = (a_{ij})$, $F = (f_{ij})$ and $B_{t_i} = (b_{ijk})$ respectively. These three matrices, altogether represent the linear code $\{A, F, B_{t_1}, ..., B_{t_L}\}$ and its transition matrix $\mathcal{M}(A, F, B_{t_1}, ..., B_{t_L})$, for each sink node t_l converts the input matrix $[\mathcal{X}_{s_1}, \mathcal{X}_{s_2}, ..., \mathcal{X}_{s_K}]$ to output matrix $[\mathcal{Z}_{t_l1}, \mathcal{Z}_{t_l2}, ..., \mathcal{Z}_{t_lK}]$:

$$[\mathcal{X}_{s_1}, \mathcal{X}_{s_2}, ..., \mathcal{X}_{s_K}]\mathcal{M}(A, F, B_{t_l)} = [\mathcal{Z}_{t_l1}, \mathcal{Z}_{t_l2}, ..., \mathcal{Z}_{t_lK}] \tag{2.7}$$

For each sink node t_l, the transition matrix $\mathcal{M}$ can be also defined as $\mathcal{M} = AGB_{t_l}^T$ where:

$$G = (1 - F)^{-1} = 1 + F + F^2 + ... \tag{2.8}$$

In [71] it was shown that in order to have a feasible solution to the above mutlicast problem, in which all sink nodes receive the information transmitted by the source nodes, the transition matrix $\mathcal{M}$ should be non-singular. In [68] it was proved that the determinant of the above matrix is equal to the determinant of its corresponding Edmonds matrix:

$$|\mathcal{M}| = (-1)^{K(|\mathcal{E}|+1)} |\mathcal{M}_e| \tag{2.9}$$

where $\mathcal{M}_e$ is the Edmonds matrix:

$$\mathcal{M}_e = \begin{bmatrix} A & 0 \\ 1-F & B_{t_l}^T \end{bmatrix} \tag{2.10}$$

2.2.3 Complexity and Optimizations

Jaggi *et al.* [72] proposed a centralized polynomial time algorithm for constructing network codes based on deterministic algorithms and a random search. Fragouli *et al.* [73] derived code design algorithms for networks based on the graph coloring techniques.

Lehman and Lehman [74] presented bounds on coding field size and classified networks due to their source/sink sets. They also proved that for non-multicast networks, finding the network capacity and even determining if it is solvable with linear network coding is a NP-hard problem. Medard *et al.* [75] proposed a time-variant coding called vector linear coding to construct code for non-multicast networks. Dougherty *et al.* [76] showed, that vector linear network coding may not achieve network capacity in non-multicast networks. Ratnakar *et al.* [77] suggested code construction methods for multiple unicast networks based on state-space realizations and linear programming. Ho *et al.* [78] developed the latter approach for wired and wireless networks.

Using encoding nodes instead of router nodes results in the more efficient use of network resources. Regarding the problem of encoding node reduction, Fragouli *et al.* [73] showed that $d-1$ coding nodes are enough for networks with two unit rate sources and d sinks. Langberg *et al.* [79] derived lower and upper bounds for a number of coding nodes. Kim *et al.* [80] showed that both the approaches in [73] and in [79] are suboptimal in some networks and presented a Genetic Algorithm based method to minimize the number of coding nodes. Lun *et al.* [81] proposed a decentralized

optimization based method for achieving minimum multicast cost. Bhattad *et al.* [82] used a linear programming approach to minimize the number of encoding nodes, but the complexity of their method is exponential.

2.3 Multiple Access Relay Channel Networks

The Multiple Access Relay Channel is a model for networks in which a finite number of sources multicast information to a destination node through relay nodes. These networks are widely used in sensor networks and ad hoc networks. The classic single-source relay network was introduced by Cover and El Gamal [83]. They developed two coding scenarios, which were later named DF and CF. [84] presented an upper bound for MARC achievable rates using cut-sets. Later Kramer et al. [85] studied the possible multicast strategies for general relay networks in detail and obtained achievable rates for Gaussian cases. Sankaranarayanan et al. [86] discussed both coding strategies in MARC networks and simulated them in Wireless examples.

2.3.1 Relay Strategies

The relay node can choose different strategies, some of which regenerate data and some of which deal with the amplification of signals.

Decode and Forward In DF the relay receives the signal from the source (or sources), tries to decode the signal and receive the original message. If this process is successful, the relay re-encodes the message to get the original codeword and transmit it. This method ensures that if the relay is sending any data, it is reliable and error-free.

Compress and Forward CF is like the DF scenario, but in this case, after correctly decoding and re-encoding the codeword, a compressed or quantized version of the

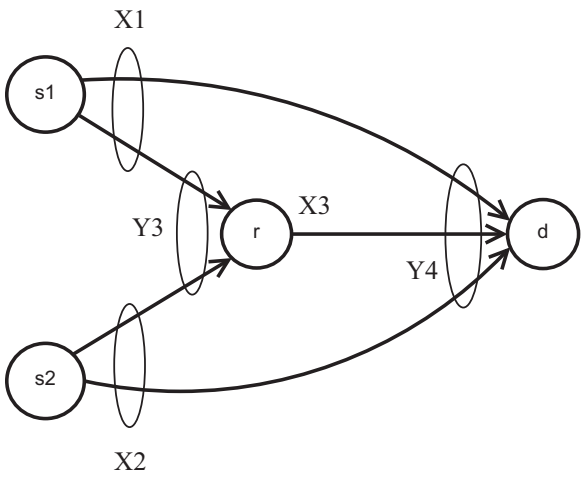

Figure 2.3: Two Source One Relay MARC Model

codeword will be transmitted. This method consumes less bandwidth and usually is used when the role of the relay is not of vital importance.

Amplify and Forward AF is the traditional strategy has been implemented for relays. There is no need to decode and re-encode in AF. The received signal is just amplified and forwarded to the next relay or destination node.

2.3.2 Capacity and Achievable Rates

The W-source discrete memoryless MARC consists of W messages w_i, W+1 channel inputs X_i and two channel outputs Y_{W+1} and Y_{W+2}. The source output X_i (i=1,2,..,W) is a function of the message W_i at the ith source while X_{W+1}, the output of relay to the channel, is a causal function of its received symbols, Y_{W+1}. Finally, the $W+1$ channel outputs at the destination are called Y_{W+2} and are used to jointly decode the messages from all W sources. Fig. 2.3 shows this model for a two-source MARC.

Sankaranarayanan et al. [87] tightened the upper bound proposed in [84] and

found the following upper bound for the MARC achievable rates:

$$\sum_{i\in G} R_i \leq \min\left\{I(X_G; Y|X_{G^c}, X_{W+1}, U), I(X_G, X_{W+1}; Y_{W+2}|X_{G^c}, U)\right\} \tag{2.11}$$

for all $G \subseteq S$ where the union is over all input distributions

$$p(u).\left(\prod_{i=1}^{W} p(x_i|u)\right) p(x_{W+1}|u, x_1, x_2, .., x_W)$$

where $S = \{1, 2, \ldots, W\}$, G^c is complement of G in S, $X_G = \{X_i : i \in G\}$, $X \triangleq (X_1, X_2, ..., X_W), Y \triangleq (Y_{W+1}, Y_{W+2}), U$ has an alphabet $\mathcal{U}$ of size $|\mathcal{U}| \leq 2^{W+1} - 2$.

In the case of DF, the achievable rate (using a combination of regular Markov encoding at the sources and the relay, as well as backward decoding) has been shown in [85]. In this case we have the following achievable rate:

$$\sum_{i\in G} R_i \leq \min\{I(X_G; Y_{W+1}|X_{G^c}, V_S, X_{W+1}), I(X_G, X_{W+1}; Y_{W+2}|X_{G^c}, V_{G^c})\} \tag{2.12}$$

for an input distribution: $\left(\prod_{i=1}^{W} p(v_i)p(x_i|v_i)\right) p(x_{W+1}|v_1, v_2, ., v_W)$ where V_i is an auxiliary random variable to help cooperation between the source and the relay and $V \triangleq (V_1, ..., V_W)$.

For CF strategy the achievable rate can be written as [87]:

$$\sum_{i\in G} R_i \leq I(X_G; \hat{Y}_{W+1}, Y_{W+2}|X_{G^c}, X_{W+1}) \tag{2.13}$$

Subject to constraint:

$$I(X_{W+1}; Y_{W+2}) \geq I(\hat{Y}_{W+1}; Y_{W+2}|X_{W+1}, Y_{W+2}) \tag{2.14}$$

for the joint distribution $\left(\prod_{i=1}^{W+1} p(x_i)\right) p(\hat{y}_{W+1}|y_{W+1}, x_{W+1})p(y_{W+1}, y_{W+2}|x_{W+2})$ where

Block 1	Block 2	Block 3	Block 4
$u_1(1)$	$u_1(w_{11})$	$u_1(w_{12})$	$u_1(w_{13})$
$x_1(1, w_{11})$	$x_1(w_{11}, w_{12})$	$x_1(w_{12}, w_{13})$	$x_1(w_{13}, 1)$
$u_2(1)$	$u_2(w_{21})$	$u_2(w_{22})$	$u_2(w_{23})$
$x_2(1, w_{21})$	$x_2(w_{21}, w_{22})$	$x_2(w_{22}, w_{23})$	$x_2(w_{23}, 1)$
$x_3(1, 1)$	$x_3(w_{11}, w_{21})$	$x_3(w_{12}, w_{22})$	$x_3(w_{13}, w_{23})$

Table 2.1: DF encoding strategy

$\hat{Y}_{W+1}$ is the compressed version of Y_{W+1}.

2.3.3 Two-Source One Relay MARC

It has been proved in [85] that for a two-source MARC as in Fig. 2.3, the achievable rate in DF is:

$$R_1 + R_2 \leq \min\{I(X_1X_2; Y_3|X_3), I(X_1X_2X_3; Y_4)\} \tag{2.15}$$

In DF the relay node decodes the source data and forwards it to destination node. A regular encoding strategy for this case can be Table 2.1 [85].

Here the message w_t from source t ($t = 1, 2$) is divided into B blocks $w_{t1}, w_{t2}, ..., w_{tB}$ of nR_t bits each (n is codeword length). Transmission is performed in $B + 1$ blocks by using codewords $u_1(i_1), x_1(i_1, j_1), u_2(i_2), x_2(i_2, j_2), x_3(i_1, i_2)$, where the first and second source transmit x_1 and x_2, respectively. The codeword of relay x_3 is statistically dependent on x_1 and x_2 through u_1 and u_2, which are auxiliary codewords at the respective sources. i_t and j_t range from 1 to 2^{nR_t}. Details can be found in Appendix A of [85];

For CF the MARC achievable rate for a two-source, one-relay case can be simplified to [85]:

$$R_1 + R_2 \leq I(X_1, X_2; \hat{Y}_3, Y_4|X_3) \tag{2.16}$$

Subject to constraint:

$$I(X_3, Y_4) \geq I(\hat{Y}_3; Y_3 | X_3, Y_4) \tag{2.17}$$

where $\hat{Y}_3$ is the compressed version of Y_3. This compression or quantization in general case is modeled by a compression noise: $\hat{Y}_3 = Y_3 + \hat{N}_3$

2.3.4 Gaussian Channel

The Gaussian scenario for a two source MARC is discussed in [86]. From proposition 2 in [85] one can simplify each of the mutual information terms of the DF capacity equation to:

$$I(X_G; Y'_U | X_{G^c}) = \int_h p(h) \log(\left|Q_{Y_U|X_{G^c}, H_{SU}=h}\right|) dh \tag{2.18}$$

where G and U are some subset of S and G^c is the complement of G in S. H_{ij} is complex fading variables between node i and j, $H_{GU} = \{H_{ij} : i \in G; j \in U\}$, $Y_U = \{Y_i : i \in U\}$ and $Y'_i = [Y_i, H_{1i}, ..., H_{(W-1)i}]$. Q is the covariance matrix.

Considering equal power for both sources, the corresponding source and relay signals are given as $X_i = \sqrt{(1-\mu)P_i}V_{0i} + \sqrt{\mu P_i}V_i$ for $i = 1, 2$ and $X_3 = \sqrt{0.5P_1}V_1 + \sqrt{0.5P_2}V_2$ where $\{V_{0i}, V_i\}_{i=1,2}$ are i.i.d complex Gaussian circularly symmetric random variables with zero mean and unit variance, P_i is the power for node i and μ is the fraction of power allocated by the source to sending new messages. So the MARC achievable rate under the DF strategy is:

$$R_1 + R_2 \leq \max_{\mu} \min\{C(\mu \frac{P_1}{d_{3,1}^{\gamma}} + \mu \frac{P_2}{d_{3,2}^{\gamma}}),$$
$$C(\frac{P_1}{d_{4,1}^{\gamma}} + \frac{P_2}{d_{4,2}^{\gamma}} + \frac{P_3}{d_{4,3}^{\gamma}} + 2\sqrt{\frac{(1-\mu)}{2}\frac{P_1}{d_{4,1}^{\gamma}}\frac{P_3}{d_{4,3}^{\gamma}}} + 2\sqrt{\frac{(1-\mu)}{2}\frac{P_2}{d_{4,2}^{\gamma}}\frac{P_3}{d_{4,3}^{\gamma}}})\} \tag{2.19}$$

where $C(x) = \log(1+x)$, $d_{i,j}$ is the distance between node i and j and γ is the path loss exponent.

2.4 Raptor Codes

Fountain codes are rateless channel codes with a non-constant rate; i.e. their rate is not known *a priori*. The destination will attempt to decode the codeword every time it receives a new symbol from the source, and this cycle continues until the destination is able to decode it successfully and send an acknowledgment signal to the source to terminate the transmission. The encoding is based on selecting and adding up some random source symbols and transmitting the resulting coded symbol.

Luby Transform (LT) codes were introduced by Luby [88]. He proposed the use of a degree distribution in order to define the number of source symbols participating in the generation of an output symbol. These degrees are derived from a probability distribution optimized for an LT code. LT code did not have a fixed encoding cost. Shokrollahi [89] addressed this problem by introducing Raptor codes, a new class of Fountain codes. In Raptor code, LT code is preceded by a high-rate code which is called a pre-code. The main purpose of this process is to fix the encoding cost. At decoding phase, only a fraction of the source symbols are decoded by the LT code layer and the remainder is decoded by the pre-code layer.

Following the use of LT code and Raptor code in the design of high-rate codes for the Binary Erasure Channel (BEC), these codes were adapted for noisy channels such as the Additive White Gaussian Noise (AWGN) channel, which required soft decoding. Nguyen et al. [90] used probabilistic decoding technique to soft decode LT codes. Jenkac et al. [91] generalized the BP decoding method for the soft decoding of LT code on binary symmetric channels (BSC). Etesami and Shokrollahi [92] used the same methods of Raptor coding on the BSC and optimized Raptor code for each AWGN channel realizations considering their characteristics.

Raptor codes consist of two layers, as shown in Fig. 2.4. The first layer is called a pre-code and is a high-rate block code. The second layer is an LT code with a degree distribution optimized for Raptor code.

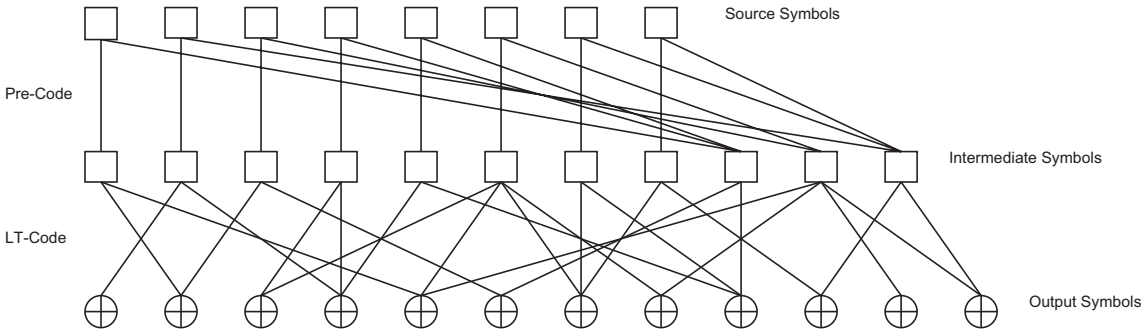

Figure 2.4: Raptor code layers

A Raptor code is characterized by parameters $(k, \mathscr{C}, \Omega(x))$ where k is the number of source symbols, $\mathscr{C}$ stands for the pre-code and $\Omega(x)$ is the degree distribution for the LT layer. Raptor code encodes k source symbols over $\mathbb{F}_2$ to a possibly infinite number of output symbols. The symbol generation is stopped when the destination can decode the transmitted message. Therefore, the length of the output codeword is not fixed *a priori*.

For Raptor codes on the soft output channels usually the pre-code is chosen to be LDPC code. This LDPC code can be a right-regular code with a high rate. The LDPC layer receives the source symbols and encodes them to generate intermediate symbols. The LT layer will generate the output symbols from these intermediate symbols. Let $\Omega_1, \Omega_2, ..., \Omega_t$ show the distribution on $1, 2, ..., t$ while Ω_i is the probability that i is chosen. This distribution is denoted by its generator polynomial $\Omega(x) = \sum_{i=1}^{t} \Omega_i x^i$. For encoding in the LT layer, degree d is sampled from the distribution $\Omega(x)$. Then d intermediate symbols are chosen according to a uniform distribution from all intermediate nodes and are combined together. The result of this combination is the value of the output symbol.

The number of the transmitted symbols for Raptor code can be written as:

$$n = \frac{k(1+\varepsilon)}{C} \tag{2.20}$$

where k is the number of source symbols, n is the number of output symbols (which is not fixed *a priori*), C is the channel capacity and ε is the overhead. From (2.20) the rate for Raptor code can be written as:

$$R = \frac{k}{n} = \frac{C}{1+\varepsilon} \tag{2.21}$$

For the LT layer, the classic BP decoder is modified. The only difference is where the channel output LLR are inserted. The LLR update rules for the decoding of LT code are as follows [92]:

$$\tanh(\tfrac{m_{o,i}^{(l)}}{2}) = \tanh(\tfrac{Z_o}{2}).\prod_{i' \neq i} \tanh(\tfrac{m_{o,i'}^{(l)}}{2}) \tag{2.22}$$

$$m_{i,o}^{(l+1)} = \sum_{o' \neq o} m_{o',i}^{(l)} \tag{2.23}$$

where i stands for the input/intermediate nodes, o for the output node, $m_{i,o}^{(l)}$ is the message passed from the input node i to the output node o at the iteration l and $m_{o,i}^{(l)}$ is vice versa. Z_0 is the channel output LLR for each bit. For BPSK case $Z_0 = L_c r$ while r is the received symbol and $L_c = 2E_s/\sigma^2$ where E_s is the symbol power and σ is the Gaussian noise standard deviation.

For MPSK, if we represent bits of each MPSK symbol b by $\{b_m ... b_1 b_0\}$, from [93] the soft inputs for Raptor decoder are calculated as:

$$\lambda_j = \ln\left[\frac{\sum_{b:b_j=0} \exp(\frac{\langle r, x(b)\rangle}{\sigma^2})}{\sum_{b:b_j=1} \exp(\frac{\langle r, x(b)\rangle}{\sigma^2})}\right] \tag{2.24}$$

where $\langle . \rangle$ represents inner product, $x(b)$ is the constellation point for b, and Z_0 for each node is equal to the corresponding λ_j.

After running BP algorithm for a sufficient number of iterations, the LLR for each intermediate node v is calculated as $LLR(v) = \sum_o m_{o,i}^{(l)}$. This *a posteriori* LLR is sent to LDPC decoder (another BP decoder) where it is used as *a priori* LLRs for

the respective symbols. The source symbol LLRs are the output of this stage of the decoding, which then pass through a hard decision stage to complete the decoding procedure.

2.5 Digital Video Broadcasting with Satellite

2.5.1 DVB-S2

Digital Video Broadcasting - Satellite - Second Generation (DVB-S2) [94] is the second generation of the popular DVB-S standard for satellite broadcasting. It outperforms the DVB-S in spectrum efficiency for constant carrier to noise ratios, while also being much more flexible than the DVB-S. The DVB-S2 can transmit a variety of streams, such as Moving Picture Experts Group (MPEG) streams, Internet Protocol (IP), and even Asynchronous Transfer Mode (ATM) packets. This flexibility is the result of Adaptive Coding and Modulation (ACM). ACM allows for the adaptation of transmission in terms of modulation and code rate for each user frame-by-frame.

The channel coding used is a concatenated code consisting of LDPC code and BCH code: LDPC code being the inner code and BCH code being the outer code. The BCH outer code is used in order to eliminate the error floor of the LDPC code. The channel code performs in different rates such as 1/4,1/3,2/5, etc. The LDPC code is decoded iteratively and its codeword length is typically 64800 bits (16200 bits for delay-sensitive systems). The modulation scheme is usually QPSK or 8PSK, although for special cases 16APSK and 32APSK are supported too.

For interactive point-to-point with the aid of ACM at each frame, optimum code rate and modulation is chosen. Each frame consists of one LDPC-BCH codeword preceded by a 90 bit header to the define code rate and the modulation of the following packet. The header itself is protected by a rate 7/64 code.

2.5.2 DVB-RCS

The DVB interactive satellite communications system, which is also known as Digital Video Broadcasting- Return Channel via Satellite (DVB-RCS) [95], is a standard for a very flexible, efficient, and low-cost two-way broadband satellite communication system. DVB-RCS can provide up to 20 Mbit/s to each terminal on the downlink, and up to 5 Mbit/s or more from each terminal on the uplink. The high capacity, flexibility and low cost of the DVB-RCS can be attributed to the following factors.

In the first place, there is the use of two different air interfaces on the outbound (downlink) and inbound (uplink). The choice of DVB-S (DVB-S2 in the next generation) on the downlink allows the use of a very mature widespread standard used for DVB. This not only enables the highly efficient coding and modulation schemes used for digital video delivery but also reduces the cost of the terminal, since the DVB-RCS terminals mostly use the circuitry used in low-cost digital TV set-top boxes for the downlink reception.

The uplink between Return Channel via Satellite Terminal (RCST) and its satellite is based on a DVB-RCS standard and uses MF-TDMA (Multiple Frequency - Time Division Multiple Access) with QPSK. There are two possible profiles for the standard: ATM and MPEG. Raptor coding performs more efficiently on large codewords. As a result, we have used the MPEG profile because of its larger packet size. Due to the choice of the MPEG profile each burst of data on MF-TDMA will contain 24 MPEG packets.

Each MPEG packet consists of a 184 byte payload and a 4 byte header. The FEC is either a (204,188) Reed-Solomon (RS) and convolutional code (CoC) combination with rates such as 1/2,2/3,3/4,... or a Turbo code with rates such as 1/3,2/5,1/2, etc.

Chapter 3

Collaborative Communication with Network Coding

3.1 Introduction

In cooperative networks, the source nodes or certain intermediate nodes (relays) help other source nodes to transmit data to its destination. As a result, the destination node receives multiple copies of each codeword, and therefore it is able to decode data more efficiently. In the cooperative networks, network coding-based schemes can be used to send redundancy to the destination node.

In this chapter, two cooperative schemes are proposed:

(i) Extended Iterative Decoding: In this scheme the source nodes send their complete codewords. These codewords are decoded in the relay and their original messages are combined to form a network code. Then the resulting codeword is channel coded again and transmitted by relay. At the destination node three iterative decoders decode the data received from the two source-destination channels and the relay-destination channel. Each decoder provides *a priori* data for the other two decoders. In contrast to previous works, the two decoders do not work independently under this

method. Instead they collaborate with each other and with the third decoder (the decoder for the relay-destination channel).

(ii) Two-Phase Collaborative Decoding: Here we add collaboration to transmission through puncturing. In this case, the source nodes send parts of their codeword; i.e. they shorten their codeword with a specific rate. The relay node decodes these codewords and the network codes the original messages, then encodes them again to get the original codewords. But instead of transmitting the whole codeword, it just sends the missing parts of the codewords. Instead of sending two missing parts, a combination of parts is transmitted, which results in a rate increase. We assume that one of the sources has more power than the other. At the destination, first, the codeword of the stronger source is decoded. Later, with help of this decoded codeword and the data transmitted from the relay, the missing part of the weaker source is extracted and decoded.

3.2 System Model

In our model we use a two-source, one-relay and one-destination cooperative network. Each source has a channel encoder and transmits its own data to both the relay and the destination. At first, using two decoders, the relay decodes the data gathered from each of the sources separately, and later the network codes and channel codes them, and finally transmits the resulting channel-network code to the destination. The destination node uses the data received from both the sources and the relay to decode both codewords. Fig. 3.1 depicts this model. The channels are considered to be interference-free and orthogonal. Multiple access strategies (such as TDMA, FDMA, and CDMA) can be used to satisfy this condition. The channel types examined here are fast-fading channels or AWGN.

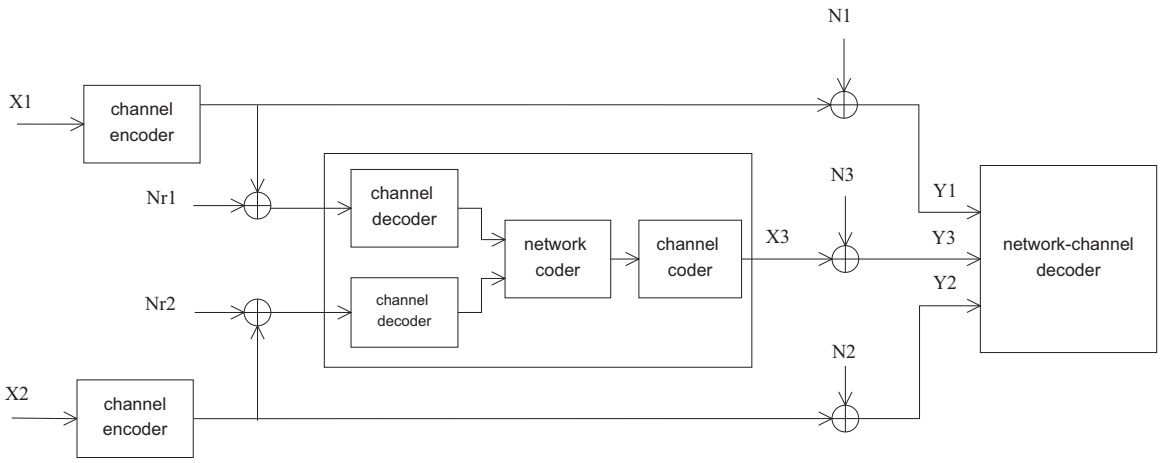

Figure 3.1: Network coding based MARC Model

3.3 Extended Iterative Decoding

In this scheme, each source encodes its data and transmits the whole codeword. The strategy that this relay uses is DF. The relay decodes the received signals to yield the original messages. These messages are then combined (in the binary case, XORed) in order to get the network code. This network code is channel coded again. The resulting code, due to the fact that LDPC is a block code, is itself a codeword and is forwarded to the destination node.

Therefore, the destination node receives three codewords: two from the sources and one from the relay. Here, the decoder uses three BP [96] decoders iteratively. BP decoders have soft output and each decoder sends its *a posteriori* LLRs to the other two decoders once its own decoding is finished. These LLRs serve as a basis for the calculation of *a priori* LLR for the other two decoders. Each decoder calculates its own *a priori* LLRs by considering *a posteriori* the LLRs of the other two decoders. The LLR can be written as:

$$L_y = \log \frac{p(y=1)}{p(y=0)} = \log \frac{p(y=1)}{1-p(y=1)} = \log \frac{1-p(y=0)}{p(y=0)} \tag{3.1}$$

$$p(y=1) = \frac{e^{L_y}}{1+e^{L_y}} \tag{3.2}$$

$$p(y=0) = \frac{1}{1+e^{L_y}} \tag{3.3}$$

Since the received signal of each decoder is the XOR of the other two signals, we can determine a given decoder's own LLR from LLRs of the other two using the following equation:

$$y_3 = y_1 \oplus y_2 \tag{3.4}$$

$$\begin{aligned} p(y_3=0) &= p(y_1 \oplus y_2 = 0) = p(y_1=0)p(y_2=0) + p(y_1=1)p(y_2=1) \\ &= \frac{1}{1+e^{L_{y_1}}}\frac{1}{1+e^{L_{y2}}} + \frac{e^{L_{y_1}}}{1+e^{L_{y_1}}}\frac{e^{L_{y2}}}{1+e^{L_{y2}}} = \frac{1+e^{L_{y_1}}e^{L_{y2}}}{(1+e^{L_{y_1}})(1+e^{L_{y2}})} \end{aligned} \tag{3.5}$$

$$p(y_3=1) = \frac{e^{L_{y_1}} + e^{L_{y_2}}}{(1+e^{L_{y_1}})(1+e^{L_{y2}})} \tag{3.6}$$

$$L_{y_3} = \log\frac{p(y_3=1)}{p(y_3=0)} = \log\frac{e^{L_{y_1}} + e^{L_{y_2}}}{1+e^{L_{y_1}}e^{L_{y_2}}} \tag{3.7}$$

where Y_1, Y_2, Y_3 are the received signals, and L_{y1}, L_{y2}, L_{y3} are the corresponding LLRs.

As mentioned before, each decoder has three inputs, two from the other decoders, carrying *a posteriori* LLRs which are used for the calculation of *a priori* LLR for the current decoder. The third input is $L_{c_y}Y$, the reliability value of the channel multiplied by its received signal. The output of the decoder is the *a posteriori* LLR, which is subtracted by $L_{c_y}Y$ to get an extrinsic value that is sent to the other two decoders. LLRs for the first decoder can be calculated as:

$$L_{y_{1(a\_prio)}} = \log(\frac{e^{L_{exy_2}} + e^{L_{exy_3}}}{1+e^{L_{exy_2}+L_{exy_3}}}) \tag{3.8}$$

$$L'_{y_1} = L_{c_{y_1}}Y_1 + L_{y_{1(a\_prio)}} \tag{3.9}$$

$$L_{ex_{y_1}} = L_{y_{1(a\_post)}} - L_{c_{y_1}}Y_1 \tag{3.10}$$

where $L_{y_{i(a\_prio)}}$, $L_{y_{i(a\_post)}}$, $L_{ex_{y_i}}$ and L'_{y_i} are the *a priori* LLR, *a posteriori* LLR, the extrinsic value and input for decoder i, respectively.

Fig. 3.2 is a schema for the proposed extended iterative decoding. Here, The

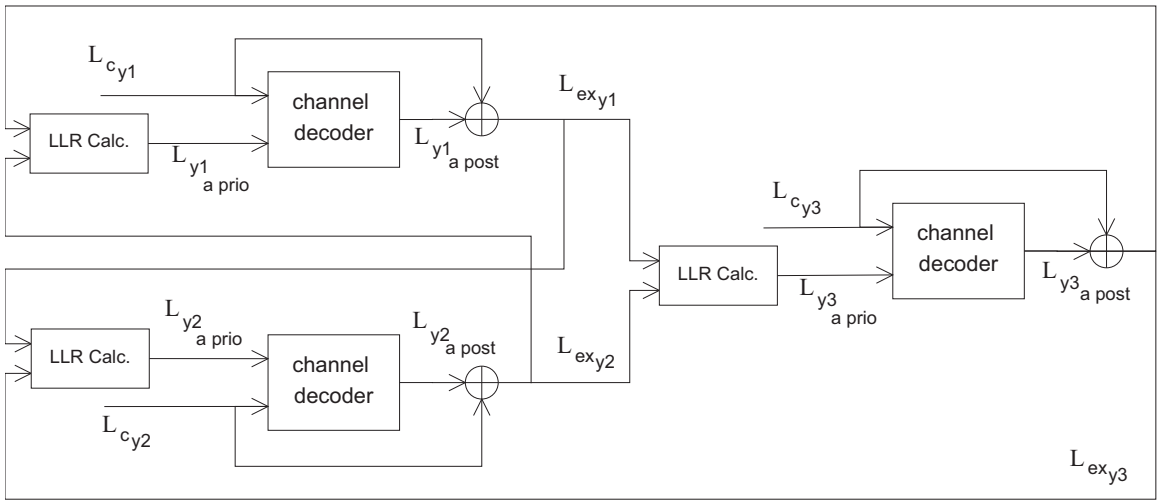

Figure 3.2: Extended iterative decoder model

LLR Calculation modules calculate $L_{y_{i(a\_prio)}}$ for decoder i with (3.8). This system continues to iteratively exchange LLRs between the decoders until their codewords correspond or a computational limit is achieved. Then a hard decision is taken regarding LLRs and the original messages of the sources can be acquired.

3.3.1 Main Parameters

There are four main parameters that influence the performance of this scheme:

BP Iterations

BP algorithm is the decoding method used to decode LDPC code. In this decoding method LLRs are passed between the variable nodes and check nodes. Variable and check nodes are parts of the structure of LDPC code generation. The n codeword bits are placed in n variable nodes. Each variable is connected to some of the $n-k$ check nodes so that if the value of the variable nodes connected to a specific check node is summed, the sum is zero. If a codeword can make all check nodes equal to zero, it is a valid codeword. In the decoding stage, LLRs are exchanged between the check and variable nodes in order to find and correct errors, until the codeword becomes

valid. Therefore, BP itself is an iterative decoding algorithm. As the number of BP iterations increases, the probability of decoding error is decreased.

Turbo Iterations

We call the number of iterations involving the three decoders in Fig. 3.2 Turbo iterations, due to similarities with the decoding of Turbo codes.

Relay power share

Relay power share demonstrates the percentage or amount of power assigned to the relay compared to the total power.

Power difference between the sources

It is possible to assign unequal shares of power to the sources. It will be shown that the unbalanced distribution of power between the two sources will increase the performance significantly. This parameter demonstrates the power difference between the two sources.

3.3.2 Simulations Setup

In our simulations, we considered the channel between the sources and the relay noiseless, so that the relay could decode the source codewords without error. In some practical scenarios this is an acceptable assumption because the relay is supposed to be near the sources or at least to have a high-quality channel. All other channels (source-to-destination channels and relay-to-destination channels) are AWGN or Reighley flat fast fading.

Right-regular LDPC codes with a node degree distribution of $\Lambda(x) = x^3$ [89] are used as channel codes. In this simulation LDPC(2000,1000) was chosen, i.e. $k = 1000$ and the rate is 1/2. The proposed scheme decreases the rate because of extra parity

bits transmitted by the relay. In this case, the relay receives two 2000 bit codewords from the sources, decodes them to the original 1000 bit messages, XORs them to get a 1000 bit network code, and finally re-encodes them to get a 2000 bit network-channel code. Hence the destination node receives 2000 bits from each of its three channels, which yields to a (3000,1000) network-channel code and drops the overall rate from 1/2 to 1/3. When comparing the proposed scheme with no-relay schemes, we used LDPC(3000,1000) to keep the rates the same and make a fair comparison.

Meanwhile, the complexity of decoding should be fixed too. The total number of iterations in the no-relay mode equals $2 * I_{LDPC}$. Because there are two decoders at the destination, each of them independently decodes its corresponding codeword in I_{LDPC}. However, in our case, the total number of the iterations is $3 * I_{Turbo} * I_{LDPC}$, since each of the three decoders has to repeat its I_{LDPC} cycle I_{Turbo} times. To have a fair comparison between the no-relay scenario and our scenario we kept the total number of iterations always fixed at 60. Thus, in the no-relay case $I_{LDPC} = 30$ and in our scheme $I_{Turbo} * I_{LDPC} = 20$. Note that this will also fix the intrinsic delay of the decoding, since the number of massage passings in BP decoders is the same and therefore the delay caused by decoding cycles will be equal.

3.3.3 Constant Total Power (CTP)

Optimization

In this case, we assumed that the total power of the whole network is constant. We developed optimization simulations to find the optimum amounts for the parameters described in the previous sections. Fortunately, the simulation results showed that these parameters were approximately independent, and there was no need to jointly optimize them. All simulations were run for two arbitrary dissimilar power differences between sources: $P_d = 5dB$ and $P_d = 10dB$. For the sake of simplicity, instead of working with the transmitter powers, we used channel SNRs. Total SNR is the

combination of the SNRs of the two source-destination channels and the one relay-destination channel. Here, Total SNR was fixed at 8.5dB and the channels were under the effect of Reighley flat fast fading.

BP Iterations

The relay share power was fixed at 40% and the number of Turbo iterations was set at three ($I_{Turbo} = 3$). Fig. 3.3 shows the achieved Frame Error Rate (FER) for different numbers of iterations. As the number of BP iterations increases, the performance gets better until saturation takes place.

Turbo Iterations

The relay share power was fixed at 40% and $I_{LDPC} = 10$. The results are shown in Fig. 3.3. FER curve saturates very fast at $I_{Turbo} = 2$. As mentioned earlier, the multiplication number for BP and Turbo iterations is fixed at 20. An increase in the number of BP iterations improves performance, while the increment of Turbo iterations saturates faster. For reaching the highest performance, we conclude that we should minimize the number of Turbo iterations; i.e. $I_{Turbo} = 2$. Since the total iteration number is limited to 20, we have $I_{LDPC} = 10$

Relay Power Share

Here the BP and Turbo iterations were fixed at 10 and 3. As shown in Fig. 3.4 there is an optimum point for relay sharing that varies as the power difference between sources changes. In this case, for $P_d = 5dB$ the optimum power share is 42%, and for $P_d = 10dB$ it is 48%. These numbers show how much power should be assigned to the relay, while the remaining power is divided between two sources with the appropriate power difference. For lower power shares than this, the relay cannot participate in the decoding stage and the system has a performance close to that of a no-relay case. For

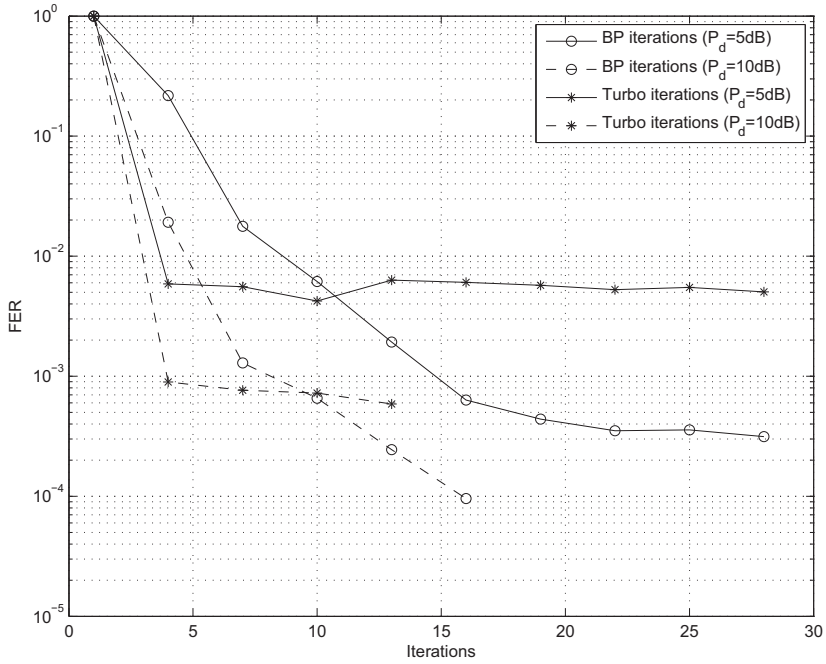

Figure 3.3: BP and Turbo Iterations Optimization (CTP)

higher powers, the main signals from the sources are too weak for efficient decoding.

Results

Considering the optimization results, we can compare the performance of the proposed scheme with the performance of the no-relay scheme. As mentioned before, for the no-relay case LDPC(3000,1000) was used as the channel code and the number of BP iterations was fixed at 30. For the network coding case, LDPC(2000,1000) was chosen. Eventually, this will result in a (3000,1000) network-channel code. BP and Turbo iterations were fixed at their optimum amounts, 10 and 2 respectively. For $P_d = 5dB$, power share was set at 42% and for $P_d = 10dB$ it was set at 48%. The simulation results for fast fading channels are reported in Fig. 3.5 and the results for AWGN in Fig. 3.6.

It can be seen that, as the power difference increases, the improvement in overall system performance is increased too. Using the same method, For the case of $P_d =$

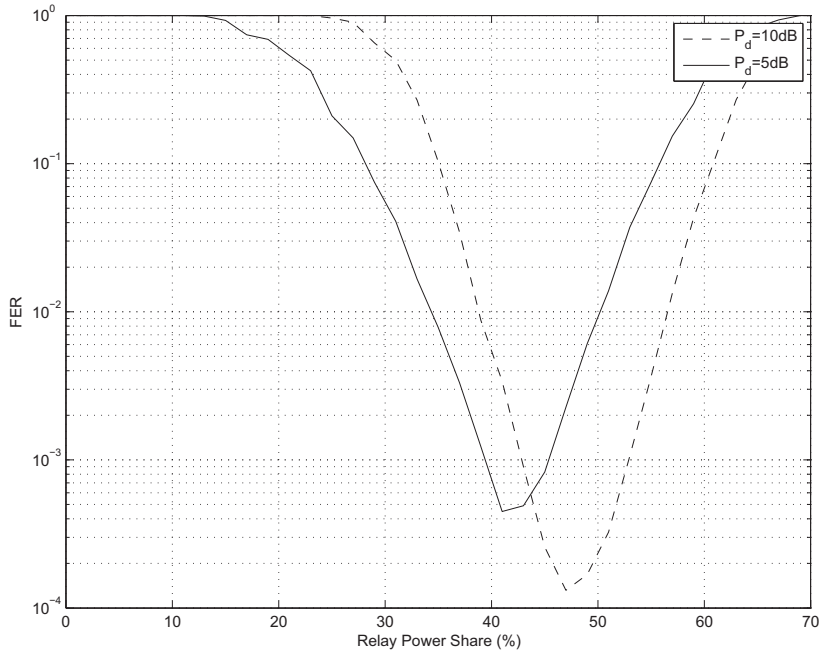

Figure 3.4: Relay Power Share Optimization (CTP)

$5dB$, the same error rate is achieved with 1dB less power, while for $P_d = 10dB$, the power reduction is 5dB. We can conclude that as power difference increases, the destination can decode more efficiently. This is due to the fact that with higher power difference, while the total power is fixed, we have an effective estimation of the stronger source at the destination. This helps to decode the weaker signals.

3.3.4 Separate Relay Power (SRP)

Optimization

In this case, we assume that the relay power is separated from the power of the source nodes. Hence, changing one will not affect the other. Here, the only difference is in the optimizations of BP iterations. For this simulation, the number of turbo iterations was fixed at 3, and power was allocated to both sources identically. Total SNR was fixed at 6dB while separately relay SNR was 5dB, and the channel is fast fading.

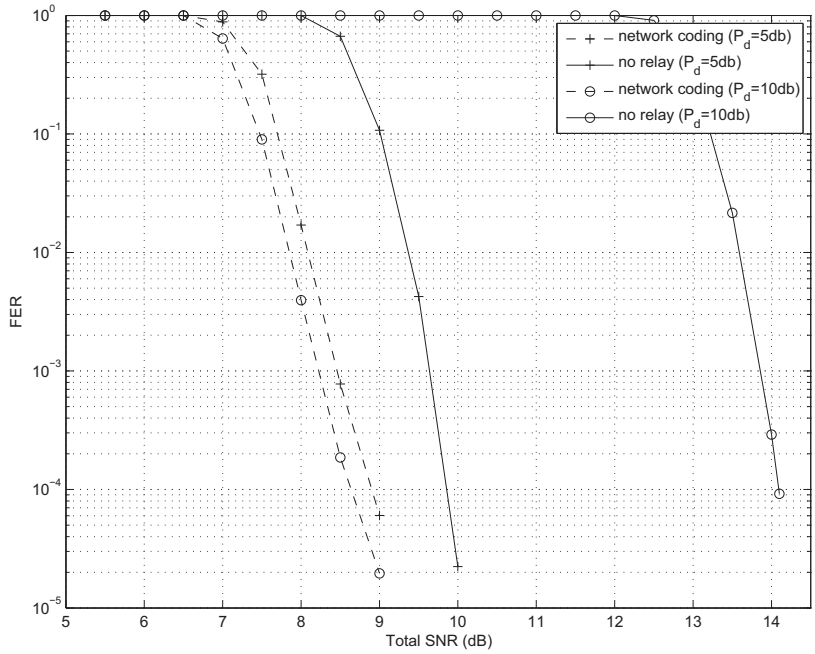

Figure 3.5: FER for fast fading channel (CTP)

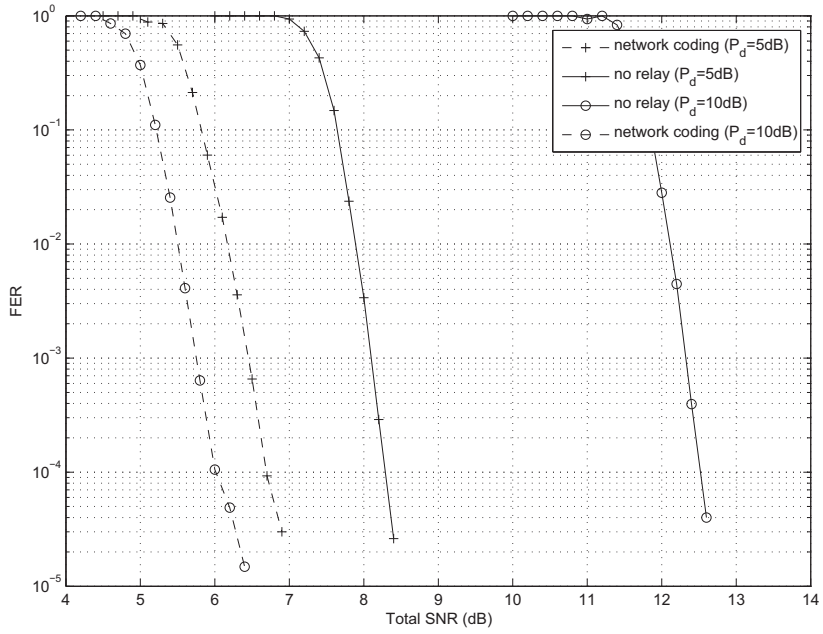

Figure 3.6: FER for AWGN channel (CTP)

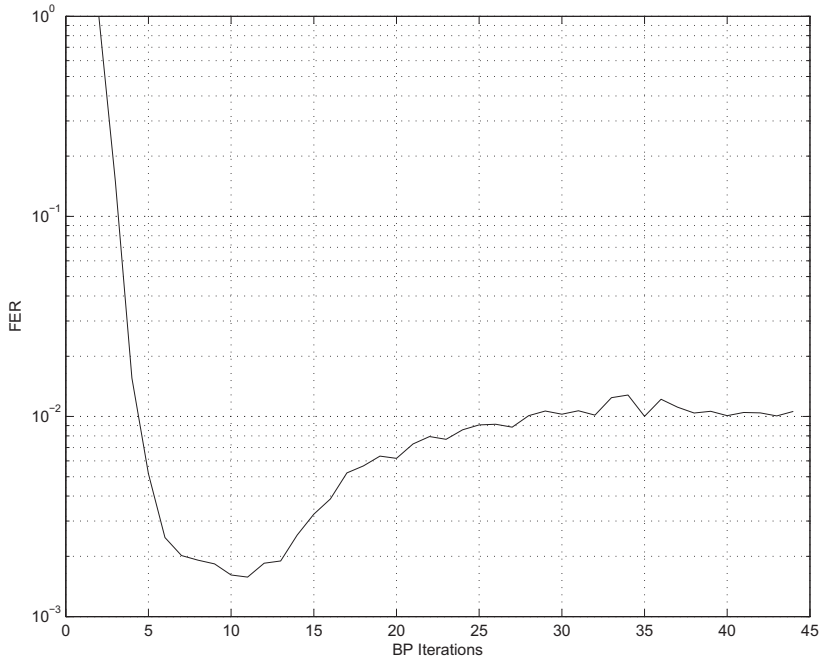

Figure 3.7: BP Iteration Optimization (SRP)

Fig. 3.7 demonstrates that there is an optimum point for BP iterations, which is 10 iterations. Therefore, again we fix $I_{Turbo} = 2$ and $I_{LDPC} = 10$.

Results

The setup is exactly the same as in the CTP case. For the no-relay case LDPC(3000,1000) was used as channel code and the number of BP iterations was fixed at 20. For the network coding case LDPC(2000,1000) was chosen, which eventually will result in a (3000,1000) network-channel code, and BP and turbo were respectively 10 and 2. 5dB power was allocated to the relay separately. The channels were under the effect of fast fading. The simulation results are reported in Fig. 3.8. It can be seen that the improvements are similar to those in the CTP case. However, the required power gap between two sources in this case is lower than in the CTP case, which is due to the fact that here the relay power is allocated separately.

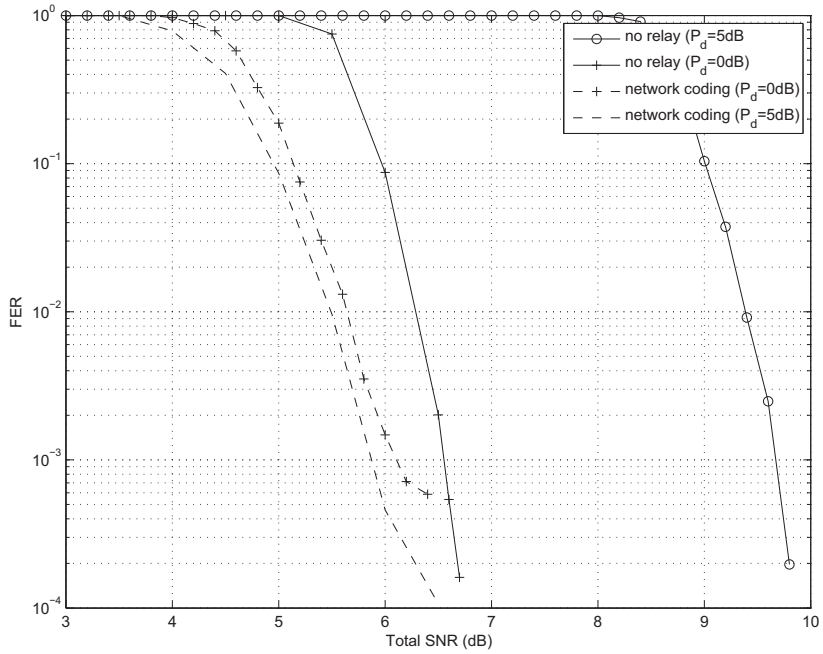

Figure 3.8: FER for fast fading (SRP)

3.4 Two-Phase Collaborative Decoding

This method is based on dividing the codewords between the sources and the relay so each node sends a part of the codeword and the destination node combines them to get the whole codeword. We call this shortening the codewords. The source nodes shorten their corresponding codewords using a specific cutting rate and transmit them. The strategy that the relay uses is again DF. The relay node receives these codes and considers the missed parts as erasures and it decodes the codewords to get the original messages. Next, the decoded messages are XORed and re-encoded. But the whole codeword is not transmitted. Only the missing parts of the original source transmissions are sent to the destination. Fig. 3.9 illustrates this procedure.

Instead of transmitting two missing parts separately, the relay sends XORed versions of them. Therefore, the number of bits transmitted by the relay is reduced, resulting in a total rate increase. As in the previous section, we consider that the

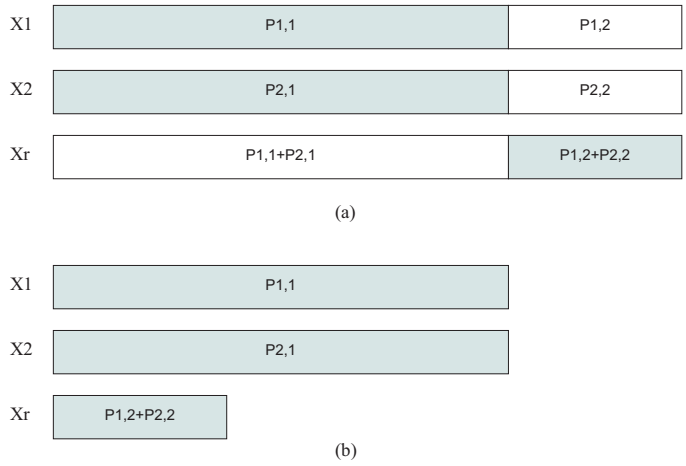

Figure 3.9: (a) Shortening codewords at the sources and the relay (b) Transmitted codeword parts

power has not been allocated equally between the sources. Thus, one of the received signals is of better quality. At the destination node (since it has access to CSI and can determine the more reliable received codeword), the signal with better quality is first decoded separately. This means that, as in the decoding at the relay, the missing parts appear to be erased during the decoding. Now that the stronger source has been decoded, it can be used to help with the decoding of weaker source. We can XOR a missing part of the stronger codeword (which is now available after its successful decoding) with data received from the relay and get the missing part of weaker codeword. We can then attach it to weaker codeword and start decoding it separately. Fig. 3.10 shows the decoding procedure (assuming that X_2 is the stronger source).

3.4.1 Main Parameters

These are three parameters that affect the performance of this scheme:

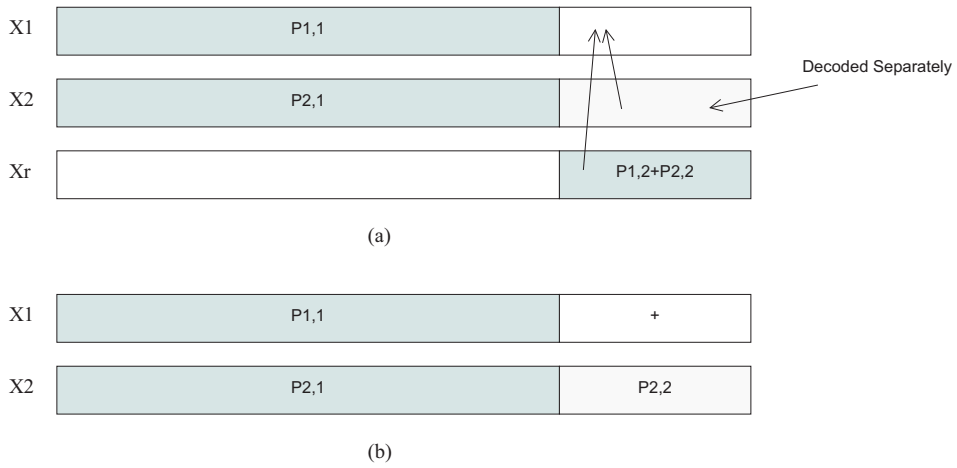

Figure 3.10: (a) Decoding stronger codeword and XORing it with relay code (b) Decoding weaker codeword

Cutting Rate

As discussed before, the proposed scheme increases the rate. Consider r' to be the cutting rate (i.e. the ratio of the bit number of the shortened codeword to that of the original codeword). Thus, nr' would be the number of bits transmitted by each source, and $n(1-r')$ is the number of bits transmitted by the relay. Therefore we can calculate the shortening rate r at follows:

$$r = \frac{nr' + nr' + n(1-r')}{2n} = \frac{1+r'}{2} \tag{3.11}$$

$$R = \frac{R_c}{r} \tag{3.12}$$

where R is the total channel-network rate and R_c is the channel code rate. In the comparison of the proposed scheme with the no-relay scenario, R is considered to be fixed.

Relay power share

In this scheme the total power of the whole network is constant and will be divided between the two sources and the relay. Relay power share involves the percentage of

power assigned to the relay compared to the total power.

Power difference between sources

It is possible to assign unequal shares of power to the sources. It will be shown that an unbalanced distribution of power between the two sources will increase performance significantly. This parameter demonstrates the power difference between the two sources.

3.4.2 Optimization and Simulation

LDPC Coding

To find the optimum shortening rate and the relay power share we developed simulation tests. Again there was almost no correlation between optimization parameters. In these simulations, we assumed that the relay can decode data from the sources error-free; i.e., that the source-relay channel is noiseless. LDPC(2000,1000) is used as the channel code and $P_d = 10dB$. The channels operate under fast fading. The number of BP iterations is 30.

Cutting Rate We establish that the total SNR is fixed at 13dB and that the relay share power varies from 10% to 30%. From Fig. 3.11 it can be determined that the optimum cutting rate, r', is 0.71. Therefore the optimum shortening rate, r, would be:

$$r'_{opt} = 0.71 \tag{3.13}$$

$$r_{opt} = \frac{1 + r'_{opt}}{2} = 0.855 \tag{3.14}$$

$$n' = nr_{opt} = 1710bits \tag{3.15}$$

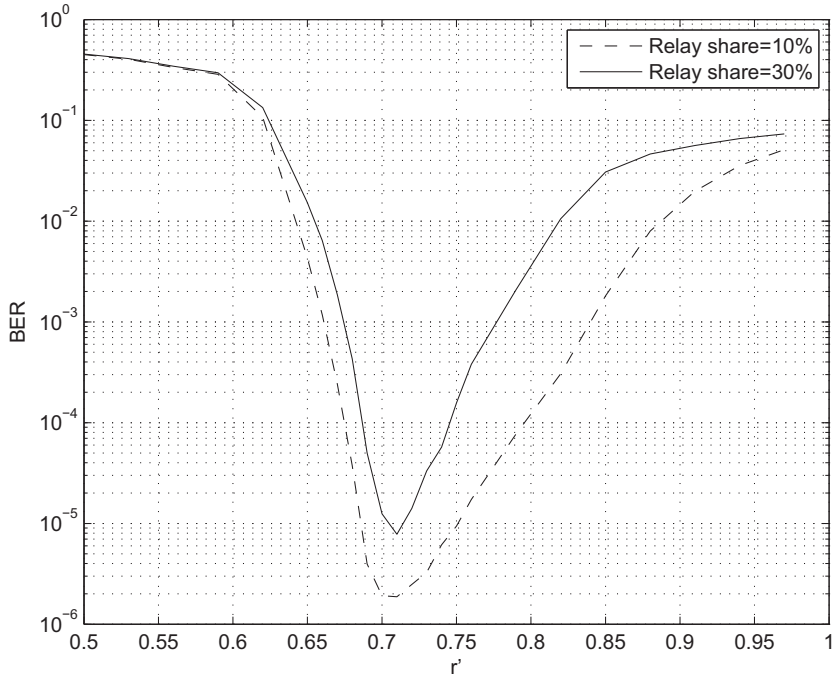

Figure 3.11: Cutting Rate Optimization (LDPC code)

where n' indicates the number of the bits transmitted for each source by the source and the relay combined. Overall, this means that for each 1000-source bit, we have transmitted 1710 bits. Hence, the overall code would be LDPC(1710,1000). R_c for the standard channel code LDPC(2000,1000) is equal to 0.5. We can find the overall rate as follows:

$$R = \frac{R_c}{r_{opt}} = 0.58 \tag{3.16}$$

which is exactly the rate for the overall network channel code LDPC(1710,1000).

Relay Power Share Here, the cutting rate is fixed at 0.71. From the simulation results which are presented in Fig. 3.12, it can be concluded that the optimal relay share power is 17%. As mentioned before, for lower powers than this optimum relay power share, the relay cannot participate in the decoding stage and the system has a

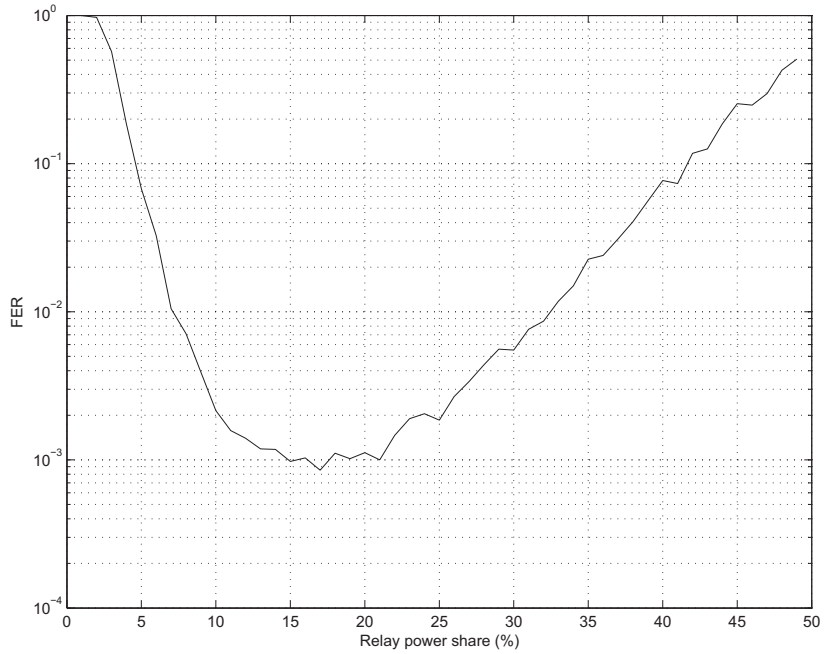

Figure 3.12: Relay Power share Optimization (LDPC code)

performance close to that of no-relay case. For higher powers, the main signals from the sources are too weak for efficient decoding.

Results As was suggested by the optimization results, the cutting rate is fixed at 0.71 and the relay share power at 17%. $P_d = 10dB$. LDPC(2000,1000) channel code is used which finally gives us a LDPC(1710,1000) network-channel code. For the no-relay scenario LDPC(1710,1000) was used so that the rates of both scenarios were the same. In addition, the channels were affected by fast fading. The number of BP iterations in the decoders for both cases is 30. Fig. 3.13 shows the results. It can be observed that the proposed scheme improves the performance by more than 3dB.

Read Solomon Coding

We have also simulated the same system in the previous section with RS codes. Here, RS(255,171) code was used as the channel code and the total SNR was fixed at 23dB.

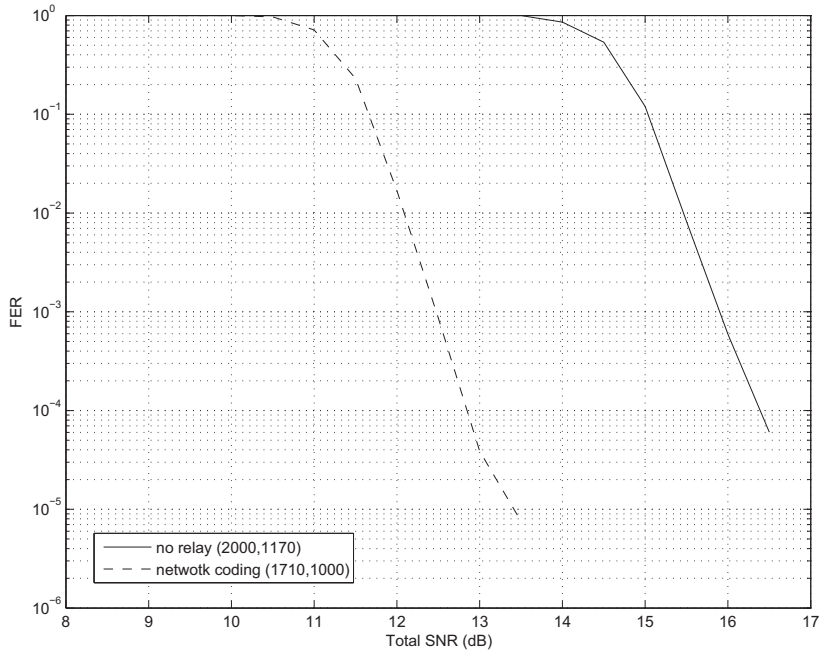

Figure 3.13: FER for fast fading (LDPC code)

$P_d = 10dB$ and channels were under fast fading effect.

Cutting Rate The relay share power range was 10% to 30%. Fig. 3.14 shows the optimization results. It is observed that the optimum cutting rate is 0.76. Consequently, we can calculate the optimum shortening rate:

$$r'_{opt} = 0.76 \tag{3.17}$$

$$r_{opt} = \frac{1 + r'_{opt}}{2} = 0.88 \tag{3.18}$$

$$n' = nr = 224bits \tag{3.19}$$

where n' shows number of bits transmitted for each source by itself and the relay.

Thus the resulted rate increased code is a RS(224,171) code. We find the overall

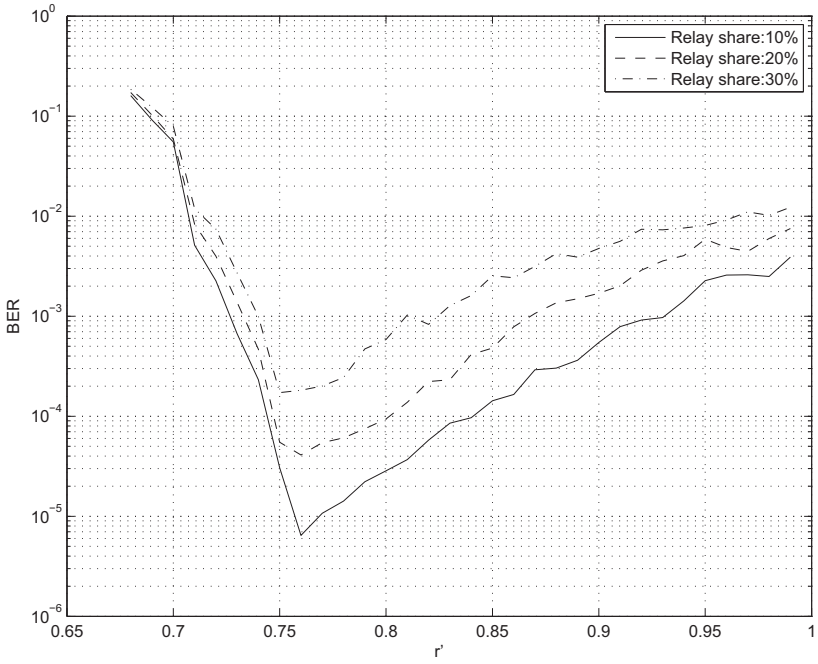

Figure 3.14: Cutting Rate Optimization (RS code)

rate to be:

$$R_c = 171/255 = 0.67 \tag{3.20}$$

$$R = \frac{R_c}{r_{opt}} = 0.76 \tag{3.21}$$

Which is the same rate as the rate of RS(224,171). However, since RS(224,171) cannot be generated, we will use RS(255,195) for the no-relay case which has the same rate.

Relay Power Share In this optimization the cutting rate is fixed at 0.76. From the simulation results which are presented in Fig. 3.15, it can be concluded that the optimal relay share power is 10%.

Results Considering the optimization results, the cutting rate is found to be 0.76, while the relay share power is fixed at 10%. The RS(255,171) is used which will

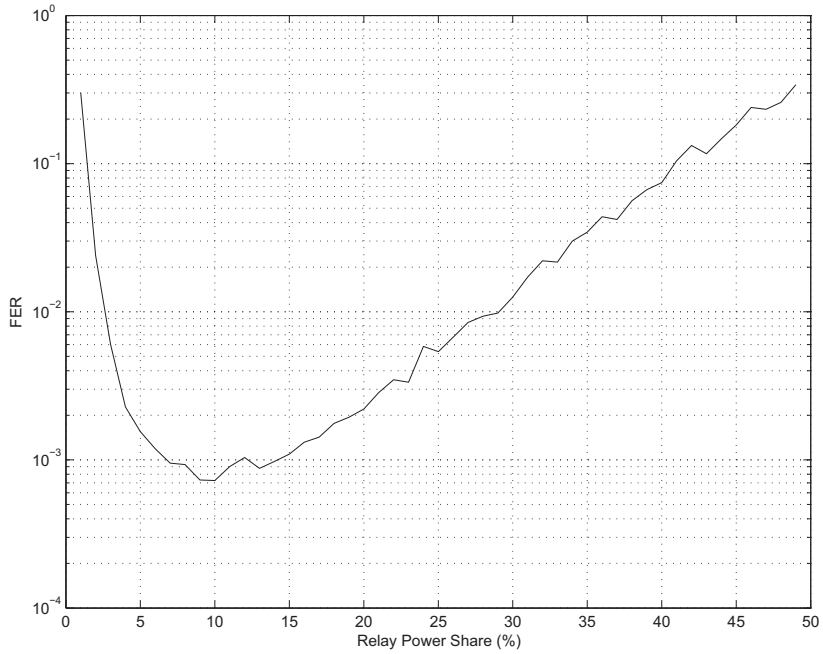

Figure 3.15: Relay Power share Optimization (RS code)

result in RS(224,171) as discussed before. For no-relay case, RS(255,195) is used, which yields the same rate as RS(224,171). $P_d = 10dB$ and channels are under fast fading. The results are reported in Fig. 3.16. It can be seen that the proposed network coding based scheme improves the performance around 3dB.

3.5 Discussion

It can be concluded that, in both of the proposed schemes, the relay blindly helps the weaker source through network coding while transmitting parity bits for both sources. The destination (in the first scheme unintentionally and in the second one intentionally) uses the signal from the more powerful source in the iterative process to retrieve the signal from the weaker source. In both schemes, it is the destination node that determines how to use this parity. That is why we say that the relay is

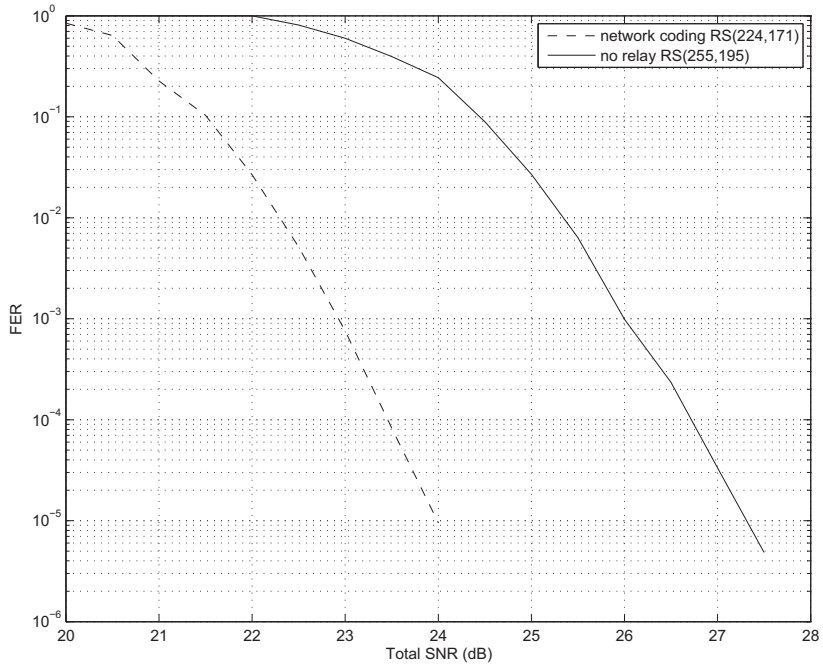

Figure 3.16: FER for fast fading (RS code)

helping both parties blindly and without having any information about CSI.

The power difference between the sources is significant too. When this gap is small or, in the worst case, when both of sources have the same power, the relay cannot increase the performance. As mentioned above, it can help only one source (i.e. the weak source), and if both sources are weak it cannot help either of them. This is the reason behind poor performance when the power difference between the sources is low.

3.6 Complexity Order

The complexity order of the above schemes can be determined for the source, relay, and destination nodes. Both here and in later chapters, we will find the complexity order in terms of the codeword length. For both of the schemes mentioned in this

chapter, we assume that LDPC was used as the channel code.

The computational complexity order of the source in both of the schemes is the same. This is also the case for the relays, since the only difference between the relays of the two schemes is the number of the transmitted symbols. In case of the two-phase collaboration, the relay computes the whole network coded codeword, but sends only a fraction of it. This does not change the complexity order.

In [97] it was shown that the complexity order of LDPC encoding is linear and of $O(n)$ where n is the codeword length. Hence, the complexity order of source nodes in Fig. 3.1 is $O(n)$.

The relay node has three components, two LDPC decoders, one network coding block, and another LDPC encoder. Chen *et al.* [98] showed that the decoding of LDPC code, like its encoding, has a linear order of computational complexity.

The network coding block is just XOR of two codewords with length n, and therefore its order is $O(n)$. The encoding block has again order of $O(n)$. Since all of the components have linear complexity orders, we can conclude that the complexity order of the computations in the relay node is also $O(n)$.

3.6.1 Complexity Order of Extended Iterative Decoding

As it can be seen in Fig. 3.2, there are three iterative LDPC decoders, each with an LLR calculation block. The LLR calculation block calculates one LLR value from two other LLRs using (3.8). Thus it performs a fixed number of calculations per the LLR value of each node, say K, and the total computations will be nK. Hence its order is $O(n)$. Therefore, each of the three decoders and three LLR calculation blocks has $O(n)$ order, and the total order will be also $O(n)$.

3.6.2 Complexity Order of Two-Phase Collaborative Decoding

In Fig. 3.9 we have two decoders with a network coding section in between. As mentioned above, both the LDPC decoders and the network coding block have linear complexity and hence the overall computational complexity order of this decoders is again $O(n)$.

Chapter 4

Successive Decoding with Raptor Codes

4.1 Introduction

In recent years, there has been increasing interest in MAC. Numerous methods have been proposed to achieve MAC capacity, such as multi-antenna techniques, orthogonal sequences, cooperation, etc. All of these methods use fixed rate channel codes and are unable to respond to channel state changes through coding alone, since fixed rate channel codes are not adaptive. Moreover, in the case of non-orthogonal channels, all of the transmitting sources in MAC require detailed CSI of all channels to adjust their power levels. This imposes high overhead on the feedback channels. However, if the fixed rate codes are replaced with rateless codes (fountain codes), the system will not need complete CSI. Furthermore, the system can adapt to changing channel quality. This can be especially useful when an interfering source is introduced to an existing source-destination link - hence forming a MAC - and the interfering source does not have access to CSI or channel variations of the pre-existing link.

Rateless codes are channel codes with a non-constant rate; i.e., their rate is not

known *a priori*. The destination will attempt to decode the codeword every time it receives a new symbol from the source, and this cycle continues until the destination is able to decode it successfully and send an acknowledgment signal to the source to terminate the transmission. This encoding is based on selecting and adding up a random subset of source symbols and transmitting the resulting coded symbol.

In this chapter, we propose the addition of an interfering channel with rateless code to an existing main DVB-RCS channel, in an arrangement where these two channels are not orthogonal. This may be the case when we want to use an existing link between a DVB-RCS terminal and the hub to serve two terminals (one main and one interfering). In such a case, the two terminals will share the same channel and consequently increase the throughput of the system. An interesting aspect of our proposed scheme is that no modification should be made to the main terminal. We will demonstrate that the added interfering channel does not have any effect on the performance of the main channel while the interfering signal itself can be decoded successfully.

In a departure from the previously-mentioned works on Multiple Access Channel (MAC) coding schemes, we do not use cooperation to achieve near-capacity rates, and thus we avoid the complexity of cooperative encoding and decoding. Instead, we will use rateless code for a physical layer, instead of packet correction (which is the usual approach in rateless FEC on DVB scenarios). In all other works, rateless codes are used in higher levels of DVB standard and therefore rateless codes work with data packets. However, we have used rateless codes in the physical layer for bit correction and we even suggest their use as the sole channel code in the standard.

For decoding, we use successive decoding, which performs more efficiently when there is a transmit power level difference between the two sources. We will demonstrate that there is a tradeoff between achieved rate and power efficiency, and we will find the optimum power allocation scenario for this tradeoff. When power adaptation

is not feasible, it is possible for the two sources to have the same power level. In such a case, since the symbols from the two sources may cancel each other out, we suggest a hard decoding stage before the decoders in order to eliminate misleading data.

Finally, we propose a power adaptation scheme which uses the feedback channel of the rateless code to estimate the channel state of the main channel and chooses the optimum power level for the interfering source accordingly. Therefore, the interfering source can transmit its data to the destination efficiently, and without having access to CSI data of the main channel. Furthermore, it can adapt itself to changing quality in the main source and destination channel.

Raptor codes can be easily adapted to DVB-S2 protocols. The long codeword length of the channel codes of DVB-S2 makes Raptor code an ideal match for LDPC-BCH code. In our proposed scheme, though the main source can use the original DVB-S2 scheme with LDPC-BCH channel code, the interfering source can use Raptor code with the same frame length (including the header) of the DVB-S2. It is possible to use Raptor code alone or as an extra coding layer (on top of LDPC and BCH codes). The latter approach allows the re-use of the already existing DVB-S2 hardware in the interfering channel. Raptor coding circuitry can even be appended to a generic DVB-S2 board as a daughter board.

Note that in this chapter and the following chapter we always assumed that we are extending and DVB-RCS link, hence all of the channels are AWGN. However, this is not a restriction of the method. The proposed methods (power adaptation in this chapter and constellation rotation in the next chapter) can be also implemented in wireless links with block fading, if there is good channel estimation at beginning of each fading block, so that the interfering source can take into account the phase rotation and amplitude change.

4.2 System Model

Consider a main source S_1 transmitting data stream T_1 to the destination D over an AWGN channel. We chose QPSK modulations since it is used in DVB-RCS standard. Source S_2, the interfering source, starts transmitting data stream T_2 on the same channel with the same modulation. We assume that the transmitters are symbol synchronized. For each source i we define codeword x_i where $E\left[{x_i}^2\right] = 1$ and $x_i(j)$ is the constellation point for symbol j in source i where $j = 0, 1, ..., M-1$ (M is the constellation size) and $i = 1, 2$. The destination receives signal y as follows:

$$y = \sqrt{E_{S_1}}x_1 + \sqrt{E_{S_2}}x_2 + z \tag{4.1}$$

where E_{S_i} is the energy per symbol for each source i and z is a circular symmetric complex Gaussian random vector, $z \sim CSCG(0, \sigma^2)$ where $\frac{\sigma^2}{2}$ is the variance of AWGN in each dimension. We assume that the interfering source power level is higher than or equal to that of the main source power and hence there can be a power level difference between the two sources. We denote the ratio of the two power levels as β ($\beta \geq 1$):

$$\beta = \sqrt{E_{S_2}/E_{S_1}} \tag{4.2}$$

Fig. 4.1 shows the system model. Through successive decoding, the destination will decode both data streams. It will first decode the interfering source data (since its power level is higher than or equal to that of the main source power) and later, by subtracting it from the received data, it will get the original data from the main source.

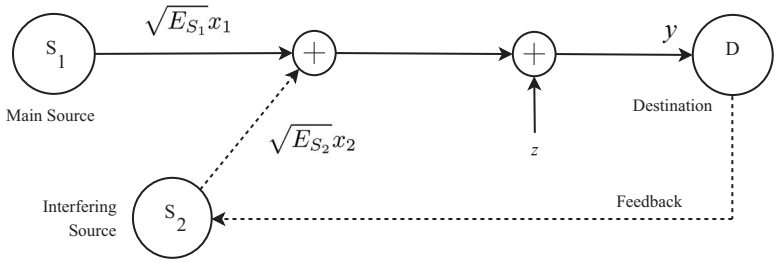

Figure 4.1: System Model

4.3 Constellation Constrained Capacity for Two Source MAC

In general, the capacity for a two source MAC can be written as [99]:

$$R_1 \leq I(x_1; y|x_2) \tag{4.3}$$

$$R_2 \leq I(x_2; y|x_1) \tag{4.4}$$

$$R_1 + R_2 \leq I(x_1, x_2; y) = I(x_2; y) + I(x_1, y|x_2) \tag{4.5}$$

But since the modulation we are using here is QPSK, we need to find the above equations for the Constellation Constrained (CC) case which is explained in detail in [100]. Fig. 4.2 shows the capacity regions for Gaussian and constellation constrained cases where $C(x) = \frac{1}{2}log(1+x)$. At points A and B, one of the sources is transmitting at its maximum rate while the other source is working below its capacity limit.

In our proposed scheme, we will transmit at point A in order to keep the performance of the main channel untouched. Therefore, although the interfering channel rate (R_2) is below its maximum achievable rate, the rate of the main channel (R_1) remains the same, i.e.,

$$R_1 \leq I(x_1; y|x_2) \tag{4.6}$$

$$R_2 \leq I(x_2; y) \tag{4.7}$$

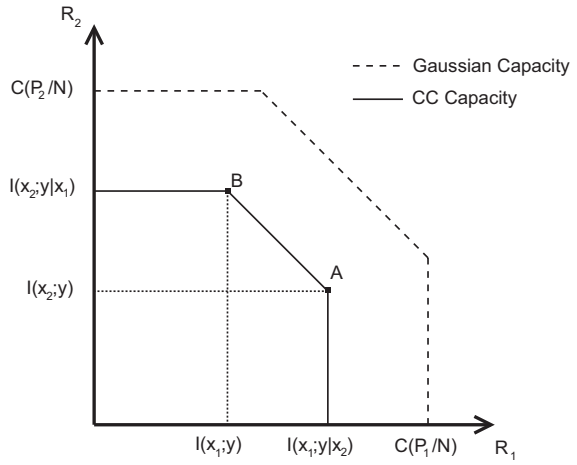

Figure 4.2: Capacity regions for two source MAC for Gaussian and CC cases

As it can be seen from (4.6) and (4.7), to compute CC capacities we just need to find $I(x_2; y)$ and $I(x_1, y|x_2)$. Here, we briefly present these results from [100]. In order to find $I(x_2; y)$, we assume that $\sqrt{E_{S_1}}x_1 + z$ is the undesired signal in (4.1) and therefore:

$$\begin{aligned} I(x_2, y) &= H(y) - H(y|x_2) \\ &= H(y) - \frac{1}{M}\sum_{i=0}^{M-1} H(y|x_2 = x_2(i)) \end{aligned} \tag{4.8}$$

To find $H(y)$ and $H(y|x_2 = x_2(i))$ we will need $p(y)$ and $p(y|x_2 = x_2(i))$:

$$p(y) = \frac{1}{M^2}\sum_{k=0}^{M-1}\sum_{i=0}^{M-1} p(y|x_1 = x_1(k), x_2 = x_2(i)) \tag{4.9}$$

$$p(y|x_2 = x_2(i)) = \frac{1}{M}\sum_{k=0}^{M-1} p(y|x_1 = x_1(k), x_2 = x_2(i)) \tag{4.10}$$

where $p(y|x_1 = x_1(k), x_2 = x_2(i))$ can be written as:

$$p(y|x_1 = x_1(k), x_2 = x_2(i)) = \frac{1}{\pi\sigma^2} e^{-\frac{|y - x_1(k) - x_2(i)|^2}{\sigma^2}} \tag{4.11}$$

By replacing (4.9), (4.10) and (4.11) in (4.8) we can find $I(x_2, y)$ as follows:

$$R_2 \leq I(x_2; y) = \log_2 M - \frac{1}{M^2} \sum_{k_1=0}^{M-1} \sum_{k_2=0}^{M-1} E\left[\log_2(\psi_2)\right] \tag{4.12}$$

where:

$$\psi_2 = \frac{\sum_{i_1=0}^{M-1} \sum_{i_2=0}^{M-1} e^{-\frac{\left|\sqrt{E_{S_1}}(x_1(k_1)-x_1(i_1))+\sqrt{E_{S_2}}(x_2(k_2)-x_2(i_2))+z\right|^2}{\sigma^2}}}{\sum_{i_1=0}^{M-1} e^{-\frac{\left|\sqrt{E_{S_1}}(x_1(k_1)-x_1(i_1))+z\right|^2}{\sigma^2}}}$$

where $E[x]$ is the expectation with respect to the distribution of z.

Similarly we can find $I(x_1; y|x_2)$:

$$R_1 \leq I(x_1; y|x_2) = \log_2 M - \frac{1}{M} \sum_{k_1=0}^{M-1} E\left[\log_2(\psi_1)\right] \tag{4.13}$$

where:

$$\psi_1 = \frac{\sum_{i_1=0}^{M-1} e^{-\frac{\left|\sqrt{E_{S_1}}(x_1(k_1)-x_1(i_1))+z\right|^2}{\sigma^2}}}{e^{-\frac{|z|^2}{\sigma^2}}}$$

Therefore, the CC capacities for MPSK modulation can be computed using (4.12) and (4.13). For simplicity, in this chapter we will refer to them as follows:

$$R_1 \leq C_1(E_{S_1}, \sigma^2) \tag{4.14}$$

$$R_2 \leq C_2(E_{S_1}, E_{S_2}, \sigma^2) \tag{4.15}$$

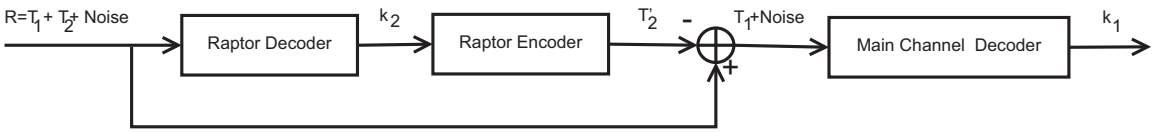

Figure 4.3: Successive decoding scheme

4.4 Proposed Scheme

Assume that the main source S_1 is transmitting data stream T_1 with k_1 source symbols to the destination. Source S_2 (the interfering source) encodes k_2 symbols using Raptor code and starts transmitting data stream T_2 on the same channel.

As mentioned before, we use successive decoding. First, the received signal is sent to the Raptor decoder, since the power level of the Raptor coded signal is higher than or equal to the power level of the main source data. After the successful decoding of the interfering signal into its k_2 source symbols, we re-encode them to get T_2' ($T_2' \equiv T_2$). Then we subtract T_2' from the received signal to get T_1 plus noise, as if there were no interfering channel at all. This signal is then sent to the main channel decoder to finalize the decoding. Fig. 4.3 shows this procedure. Since the transmitted signals from both sources are combined in the wireless channel, the required bandwidth is not increased.

4.4.1 Sources with Equal Transmit Power Levels

In this case, both sources have equal transmit power levels and the capacity of both channels can be calculated from (4.14) and (4.15) with $E_{S_1} = E_{S_2} = E_S$. In general, the two QPSK constellation maps from the two sources with different symbol powers would add up and result in 16 merged constellation points. However, when both sources have equal powers ($\beta = 1$), we will have a multiple access binary erasure channel [101], with just 9 merged constellation points as shown in Fig. 4.4. Here the four outer constellation points each have a probability of 1/16, the middle points

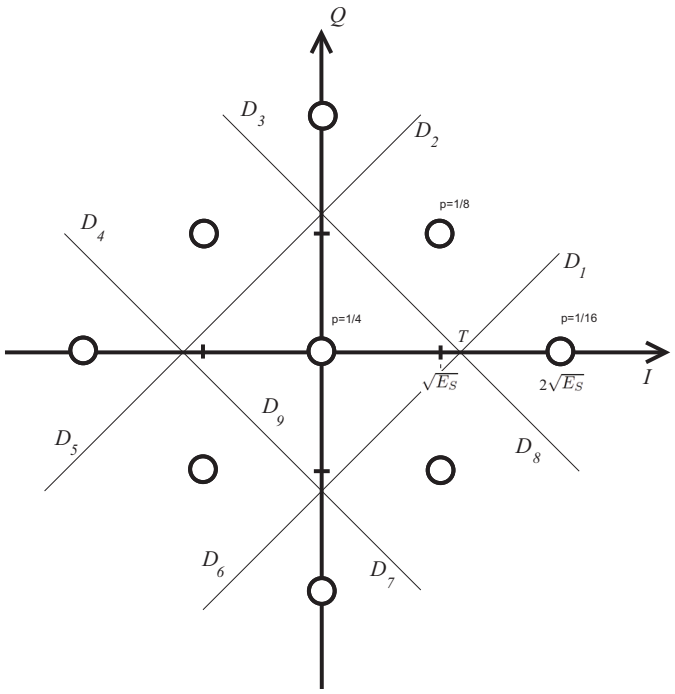

Figure 4.4: Received symbols and their probability

have a probability of 1/8, and the constellation point at the origin (which represents the completely erased symbols) has a probability of 1/4.

This is due to the fact that, in this case, some of the symbol pairs transmitted by the sources cancel each other out. For example, without considering the effect of the noise, if we receive a zero at the destination, the two sources may have transmitted $[\sqrt{E_S}, -\sqrt{E_S}]$, $[-\sqrt{E_S}, \sqrt{E_S}]$, $[j\sqrt{E_S}, -j\sqrt{E_S}]$ or $[-j\sqrt{E_S}, j\sqrt{E_S}]$. Therefore we have lost 4 entire bits of data (2 bits from each source).

Now consider that at the destination (again without considering the effect of the noise) we have received $[\sqrt{E_S} + j\sqrt{E_S}]$ (a middle point). This means that the sources have transmitted either $[\sqrt{E_S}, j\sqrt{E_S}]$ or $[j\sqrt{E_S}, \sqrt{E_S}]$. If gray code is used in the modulation, we can at least recover one bit from each source. Therefore we have lost 2 bits in total.

To calculate the erasure probability (P_{er}), we use the following sum:

$$P_{er} = \sum_{i=1}^{N_c} p(i)e(i) \tag{4.16}$$

where N_c is the number of the merged constellation points ($N_c = 9$), $p(i)$ stands for the probability of symbol i $(1/4, 1/8, 1/16)$, and $e(i)$ defines the ratio of the bits lost in that symbol which is 1 for the symbol at origin, 0.5 for the middle symbols and 0 for the outer symbols.

In this case, it can be demonstrated that $P_{er} = 0.5$. Therefore, the scheme can be modeled as an erasure channel with $P_{er} = 0.5$. Since the channel capacity is reduced to 1/2 of its original capacity, we need to send twice the number of output symbols. From (2.20) this can be explained as:

$$n_{eq} = \frac{k(1+\varepsilon)}{C_{eq}} = \frac{k(1+\varepsilon)}{(0.5)C} = 2\frac{k(1+\varepsilon)}{C} = 2n \tag{4.17}$$

where the subscript *eq* signifies the fact that a case with equal power is being considered.

The erased bits do not contain any information and even create inaccuracies in the decoding process due to the effect of the noise. Therefore, in order to increase the performance of the decoding, this misleading data should be eliminated before the decoding stage. This elimination can be done by means of a hard decision.

In optimal detection, upon observing y, the detector looks for the constellation point that maximizes $p(x(m)|y)$ which is the probability that the constellation point $x(m)$ was transmitted (observing the fact that y is received). The optimal detection rule [102] can be written as:

$$\hat{m} \quad = \quad \underset{1\leq m\leq N_c}{\arg\max}[p(x(m)|y)] = \underset{1\leq m\leq N_c}{\arg\max}[p(x(m))p(y|x(m))]$$

$$= \arg\max_{1\leq m\leq N_c}[p(x(m))\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{\|y-x(m)\|^2}{2\sigma^2}}] \tag{4.18}$$

where $\hat{m}$ is the detected symbol.

The above equation can be simplified to:

$$\hat{m} = \arg\max_{1\leq m\leq N_c}[\eta_m + y.x(m)] \tag{4.19}$$
$$\eta_m = \sigma^2 \ln p(x(m)) - \frac{1}{2}\|x(m)\|^2$$

The decision regions in this scheme can be calculated as:

$$D_m = \{y : y.x(m) + \eta_m > y.x(m') + \eta_{m'}, \forall m' \neq m\} \tag{4.20}$$

where $1 \leq m \leq N_c, 1 \leq m' \leq N_c$.

In our case, in order to find the borderline between decision regions (Fig. 4.4), we will find the line where the above inequality becomes an equality. If we assume received symbol $y = I + jQ$, $x(1) = \sqrt{E_s}(1+j)$, $x(2) = 2\sqrt{E_s}$ and $\sigma = 0.5$. From Fig. 4.4 we will have $p(x(1)) = 1/8$ and $p(x(2)) = 1/16$. Hence:

$$\eta_1 = \frac{1}{4}\ln\frac{1}{8} - \frac{1}{2}\left\|\sqrt{E_s}(1+j)\right\|^2 = -\frac{3}{4}\ln 2 - E_s \tag{4.21}$$
$$\eta_2 = \frac{1}{4}\ln\frac{1}{16} - \frac{1}{2}\left\|2\sqrt{E_s}\right\|^2 = -\ln 2 - 2E_s \tag{4.22}$$
$$(I+jQ).\left(\sqrt{E_s}(1+j)\right) - \frac{3}{4}\ln 2 - E_s = (I+jQ).\left(2\sqrt{E_s}\right) - \ln 2 - 2E_s \tag{4.23}$$
$$I = Q - \frac{1}{4\sqrt{E_s}}\ln 2 - \sqrt{E_s} \tag{4.24}$$

We can summarize (4.24) as:

$$I = Q - T \tag{4.25}$$

where $T = \frac{1}{4\sqrt{E_s}}\ln 2 + \sqrt{E_s}$. If we find all other decision region borderlines, they all

can be shown to be:

$$I \pm Q = \pm T \tag{4.26}$$

If the received symbol is in D_9, then because all the data is erased the received data will not be considered in the decoding. If it is in D_2, D_4, D_6 or D_8 then only one bit per source is lost, and the other bit can be used in the decoding. Therefore, if we represent bits of each QPSK symbol by $b_1 b_0$, from (2.24) the soft inputs for the Raptor decoder are calculated as:

$$\lambda_0 = \begin{cases} 0 & y \in D_9, D_2, D_6 \\ \ln\left[\frac{\sum_{b:b_0=0} \exp(\frac{\langle y, x(b)\rangle}{\sigma^2})}{\sum_{b:b_0=1} \exp(\frac{\langle y, x(b)\rangle}{\sigma^2})}\right] & o.w. \end{cases} \tag{4.27}$$

and

$$\lambda_1 = \begin{cases} 0 & y \in D_9, D_4, D_8 \\ \ln\left[\frac{\sum_{b:b_1=0} \exp(\frac{\langle y, x(b)\rangle}{\sigma^2})}{\sum_{b:b_1=1} \exp(\frac{\langle y, x(b)\rangle}{\sigma^2})}\right] & o.w. \end{cases} \tag{4.28}$$

where λ_j is the LLR for b_j in each QPSK symbol.

The value of T in (4.26) affects the performance of the system by changing the size of the decision regions, as will be demonstrated in the next section through simulations.

4.4.2 Sources with Unequal Transmit Power Levels

Here, in contrast to the previous case, the interfering source can have a higher power level than the main source ($\beta > 1$); hence the destination can decode both codewords more efficiently and higher rates are achieved. The value of β has a considerable effect on the achieved rates, as will be shown with simulations. Furthermore, there is an optimum value for β from a power efficiency-rate tradeoff point of view.

To evaluate this tradeoff we need to find the required power for any specific achieved rate. For the interfering channel the corresponding E_b/N_0 for an achieved rate R_2 can be calculated as follows:

$$\frac{E_{b_2}}{N_0} = \frac{E_{c_2}}{N_0 R_2} = \frac{E_{S_2}}{N_0 R_2 \log_2 M} \tag{4.29}$$

To find out how far we are from the theoretical limits, we calculate the required power E'_{S_2} to achieve the same R_2 with (4.15), i.e., the same amount of power that a capacity-achieving code would require to get R_2:

$$R_2 = C_2(E_{S_1}, E'_{S_2}, \sigma^2) \tag{4.30}$$

$$\frac{E'_{b_2}}{N_0} = \frac{E'_{S_2}}{N_0 R_2 \log_2 M} \tag{4.31}$$

We can compare the actual E_{b_2}/N_0 with its corresponding E'_{b_2}/N_0 (derived from E'_{S_2}) for each β. We call the gap between them Δ:

$$\Delta[dB] = \frac{E_{b_2}}{N_0}[dB] - \frac{E'_{b_2}}{N_0}[dB] \tag{4.32}$$

Δ shows the difference between power levels that our scheme and a capacity-achieving code would need to achieve a specific rate: i.e., the power wasted by our scheme compared to an ideal capacity-achieving code. Now we can find out which power allocation scenario (which β value) minimizes this gap. Unfortunately, the achievable rates of Raptor codes cannot be calculated analytically, and due to this it is not possible to analytically optimize (4.32) against β. Therefore, we will use simulations to analyze this gap. In the simulations section it will be shown that Δ has a concave curve against β, so that there is an optimum β (β_{opt}) which minimizes this gap and indicates a power scenario with the least waste of power.

Power differences below β_{opt} are not sufficient for the efficient successive decoding

of the two sources. For β above this value, although higher rates can be achieved, the power is used less efficiently. In fact, with a capacity-achieving code, this rate could have been achieved with much less power. This is a case of the previously-mentioned power efficiency-rate tradeoff. Note that we assume that E_{S_1} is known, and therefore we can find the optimum transmit power for the interfering source from (4.2): $E_{S_2(opt)} = E_{S_1}(\beta_{opt})^2$.

4.4.3 Power Adaptation

One of the exceptional advantages of rateless codes is their ability to transmit data over channels with different qualities. This arises from the fact that they continue to generate and transmit symbols until the destination sends an acknowledgment signal confirming successful decoding of the codeword. This advantage comes at the price of the need for a feedback channel between destination and source nodes. Since this feedback can be as simple as an acknowledgment, it can be highly coded, and we assume that it is error-free.

Nevertheless, this feedback can also be used for channel estimation purposes. In Section 4.4.2 we assumed that the interfering channel has a precise estimation of the channel between the main source and the destination, and that through this estimation it could optimize β and its transmitting power level. However, in practical scenarios this is not the case, and the interfering channel does not have any estimation regarding the main channel. Therefore, an estimation of the power level of the main source is necessary. In this section we propose the use of the already-mentioned feedback data to address this problem.

Power Estimation and Adaptation

The main idea here is to use the feedback from the destination to estimate the main channel's power, and with that estimation to find the optimum power level for the

interfering source through the optimization of (4.32). We assume that we are aware of demodulator sensitivity in the destination and hence can estimate the channel noise. For estimating the main source's power, we use tables or curves simulated *a priori*. Assume that, in a controlled environment where we have control of the power levels of both sources, we can fix the power per symbol of the interfering channel E_{S_2} at some specific test level. Then, for the different symbol powers of the main source (E_{S_1}, or β consequently) we simulate the achievable rate at an specific noise power for the interfering channel. Therefore we have the achievable rates for the interfering source at a specific noise power and different main source powers.

Having these simulation results in hand, at the beginning of the transmission phase of a real-world case (when we do not have any estimate of the main source's power) the transmitter chooses from one of the test powers that it has its achievable rate curves *a priori*. Then it starts transmission with that test power $E_{S_2(test)}$ until it gets the feedback from the destination. This feedback can be simply an acknowledgment bit which shows how many symbols were sufficient for successful decoding of the codeword. Therefore, the transmitter can calculate the achieved interfering channel rate, i.e., R_2. Now, using the curves that it has *a priori*, it can calculate β and with (4.2) get an estimation of E_{S_1}. This estimation will be used to find the β_{opt} through optimization of (4.32) and from there to calculate $E_{S_2(opt)}$.

Overall Algorithm

The following algorithm explains the whole procedure:

1: Choose $E_{S_2(test)}$ for the interfering source.

2: Transmit one (or more) codeword with the test power level.

3: Receive feedback from the destination.

4: Calculate R_2.

5: Find β from the rate curves or tables simulated *a priori*.

6: Calculate $E_{S_1} = E_{S_2(test)}/\beta^2$.

7: Find the β_{opt} for the estimated E_{S_1} through optimization of (4.32).

8: Calculate $E_{S_2(opt)} = E_{S_1}(\beta_{opt})^2$.

9: Transmit next codewords with $E_{S_2(opt)}$.

Robustness

The above channel estimation method can also be used to make the interfering channel robust to power level changes in the main source. If after the first adaptation the achieved rate of the interfering channels changes significantly, this means that the main channel power has either increased or decreased. Therefore, the optimum power level that is currently used for the interfering source is not optimum anymore. In this case, the interfering source can easily adapt itself to the new channel conditions the same way it adapted to the initial conditions. The only difference is that its current power level is the test power level in the above algorithm. Therefore, since it has its current power and rate, simply by a search in the corresponding table, it can find the optimum power for the current channel states and adapt itself to the new conditions. In other words, in the above algorithm, adaptation begins at step 5 with $E_{S_2(test)} = E_{S_2(current)}$ and $R_2 = R_{2(current)}$. This is why we call this method robust to channel changes.

4.5 Simulation Results

In our simulations a rate 0.98 right-regular LDPC code with $k_2 = 1472$ has been chosen as the pre-code for the Raptor code. The rate and type of the LDPC code is identical to Shokrollahi's original simulations in [89]. Like LT codes, LDPC codes are decoded by the BP algorithm and therefore, to simplify the simulations, LDPC code is chosen as the pre-code. As mentioned before, The pre-code must have a high rate

(hence the rate 0.98) and the codeword length of 1472 comes from the MPEG packet length which, as mentioned in section 2.5, has a payload of 184 bytes (or 1472 bits). For the LT layer the distribution $\Omega(x)$ is the optimized distribution for $k = 65536$ case in Table I at [89]:

$$\begin{aligned}\Omega(x) &= 0.008x + 0.49x^2 + 0.166x^3 + 0.072x^4 + 0.083x^5 + 0.056x^8 + 0.037x^9 \\ &+ 0.056x^{19} + 0.025x^{66} + 0.003x^{67} \end{aligned} \tag{4.33}$$

The above weight distribution is optimized for the erasure channels. In [92] it was shown that in AWGN channels, Raptor codes lose their generality and for each value of σ a specific Raptor code should be designed. However, it was demonstrated that although the weight distribution for the erasure channels is not optimized for AWGN channels (leaving room for improvement) it performs very well and its achieved rate is acceptable. Therefore, for the sake of simplicity we used the BEC weight distribution here.

At the destination, both of the decoders employ the BP decoding with BP iterations of 50 and 300 for the LDPC layer and LT layer respectively. Our empirical results show that at these numbers of iterations the corresponding decoders saturate and no more error correction is possible. The standard deviation for the noise per dimension in the AWGN channel is chosen to be $\sigma = \sqrt{0.5}$. The modulation is QPSK and hence $M = 4$ in (4.12) and (4.13). Without loss of generality and for the sake of simplicity, the main channel is considered to be transmitting an uncoded data stream.

We have used the Monte Carlo method for the simulations. In all of the rate simulations, the achieved rate of a Raptor code is the rate at which the channel code guarantees a BER not exceeding 10^{-4}.

4.5.1 Confidence Interval

To calculate the confidence level for BERs (which is a series of error tests $\mathscr{X}$) we calculate that around BER = 10^{-4}, the average number of the transmitted bits in each simulation point is about $n = 2.2 \times 10^8$. The sample mean is $\bar{x} = 10^{-4}$. For variable x, the sample variance (when it is unknown) can be estimated from the mean as follows [103]:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{4.34}$$

In our case, since the bit errors are either zero or one, the above equation can be simplified as [104]:

$$\begin{aligned} s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 &= \frac{1}{n-1} \left\{ \sum_{i=1}^{n} {x_i}^2 - 2\bar{x} \sum_{i=1}^{n} x_i + n\bar{x}^2 \right\} \\ &= \frac{1}{n-1} \left\{ \sum_{i=1}^{n} {x_i}^2 - n\bar{x}^2 \right\} \end{aligned} \tag{4.35}$$

Since $x_i = \{0, 1\}$:

$$\sum_{i=1}^{n} {x_i}^2 = \sum_{i=1}^{n} x_i = n\bar{x} \tag{4.36}$$

Therefore:

$$s^2 = \frac{1}{n-1} \left\{ n\bar{x} - n\bar{x}^2 \right\} = \frac{n}{n-1} \bar{x}(1 - \bar{x}) \tag{4.37}$$

Hence, the sample variance is $s^2 = 9.99 \times 10^{-5}$.

For confidence coefficient ω the confidence interval of $\mathscr{X}$ is shown to be [103]:

$$P \left\{ \bar{x} - t_{1-\delta/2}(n) \frac{s}{\sqrt{n}} < \mathscr{X} < \bar{x} + t_{1-\delta/2}(n) \frac{s}{\sqrt{n}} \right\} = 1 - \delta = \omega \tag{4.38}$$

where $t_u(n)$ is student's t percentile.

Here, if we assume $\omega = 99\%$, then $1 - \delta/2 = u = 0.995$ and from Table 8.2 in

[103] $t_u(n) = 2.75$. The confidence interval of our simulations is

$$P\left\{10^{-4} - 1.85 \times 10^{-6} < \mathscr{X} < 10^{-4} + 1.85 \times 10^{-6}\right\} = 99\% \tag{4.39}$$

$$P\left\{9.81 \times 10^{-5} < \mathscr{X} < 1.02 \times 10^{-4}\right\} = 99\% \tag{4.40}$$

In other words, any rate achieved by Raptor code has a BER in the above interval with a probability of 99%.

4.5.2 Sources with Equal Transmit Power Levels

Here we assumed that $E_S = 2$, and from (4.20) it can be shown that for optimal decision regions in (4.26) $T = 1.53$. In the optimal decoding each erroneous detection results in one error, and therefore if an erased symbol is detected as a non-erased one it will have only a minor effect on the total decoding process. Yet in the case of Raptor decoding, due to its iterative decoding nature, errors propagate and even a small number of erroneous detections may result in decoding failure. Therefore, it is to be expected that in our case the erasure zone should expand beyond the above threshold.

We have simulated this scenario with different values for T. Fig. 4.5 shows the achievable rates with different T values. As mentioned before, the achieved rate is the rate at which the channel code guarantees a BER not exceeding 10^{-4}. As expected, there should be one optimum point for T. Fig. 4.6 shows that this optimum T is around 1.8. As predicted above,this number exceeds the threshold acquired from the optimal detection regions.

The achievable rate from (4.14) and (4.15) for $\beta = 1$ can be written as:

$$R_1 \leq C_1(2, 0.5) = 1.9677 \tag{4.41}$$

$$R_2 \leq C_2(2, 2, 0.5) = 0.9860 \tag{4.42}$$

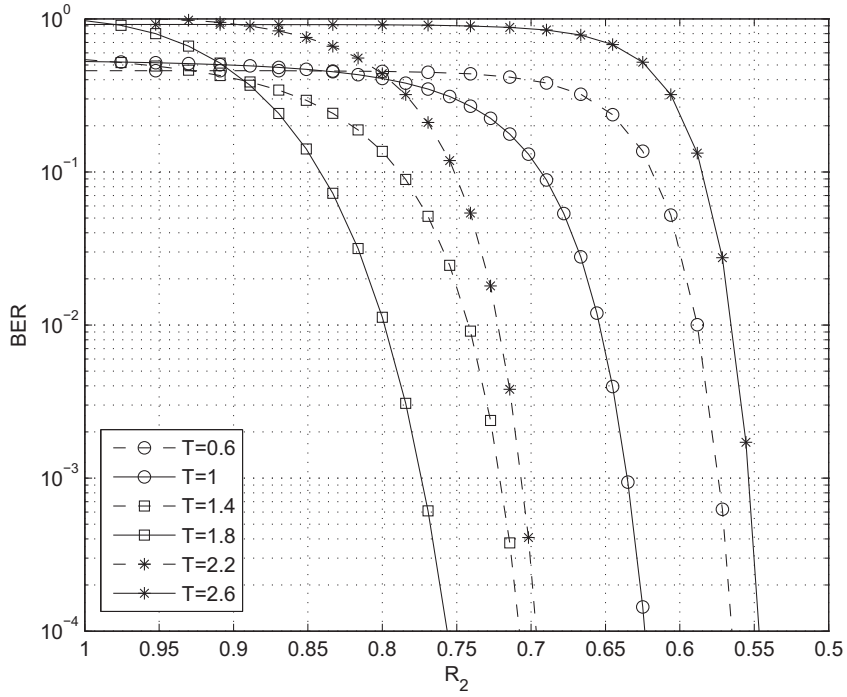

Figure 4.5: Performance for equal source power scenario with a hard decision stage

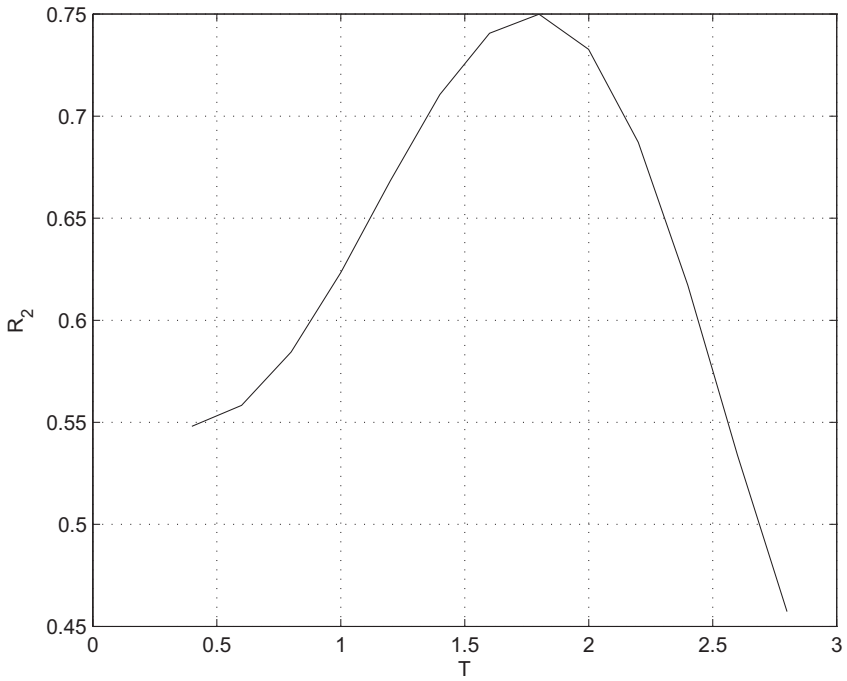

Figure 4.6: Achievable rates for the interfering source with different T values

From Fig. 4.5, for the case of $T = 1.8$, we derive $R'_2 \simeq 0.756$. It can be concluded that this scheme has achieved 76% of the interfering channel capacity. Furthermore, we have increased the total throughput. Without the interfering source, the total rate would be $R = R'_1 = 1.38$ (if the main DVB-RCS channel was using RS(204,188) and rate 3/4 CoC with QPSK), while with our scheme it will increase to $R = R'_1 + R'_2 = 1.38 + 0.756 = 2.138$ which shows 55% increase of the throughput. Note that this increase does not affect the performance of the main channel, which will be shown later with simulation results.

4.5.3 Sources with Unequal Transmit Power Levels

Here, we present simulation results for a situation with sources that have unequal power levels ($\beta > 1$). Fig. 4.7 shows BER against the rate of the interfering channel with various β while $E_{S_1} = 4.5$. As β increases, the achievable rate is increasing too. We can compare these achieved rates with the capacity derived from (4.15) for the same power scenario. Fig. 4.8 shows that the difference between the achieved rate and the capacity decreases as β increases.

As discussed in the previous section, there is a tradeoff between power efficiency and rate, and the optimum point of this tradeoff can be calculated through the optimization of Δ in (4.32). Fig. 4.9 shows Δ for our simulation settings. It can be observed that the curve is convex, and increasing the difference between the source powers after a certain point enlarges the gap between these two parameters. As an example, the optimum β in our simulation for $E_{S_1} = 4.5$ is $\beta_{opt} = 1.75$, hence from (4.2) $E_{S_{2(opt)}} = 13.78$.

Furthermore, Fig. 4.9 also shows that as E_{S_1} increases, β_{opt} is increased too. Fig. 4.10 demonstrates the value of β_{opt} for different values of E_{S_1} and shows this increasing trend of β_{opt}. This curve shows the optimization results of (4.32); i.e., for each E_{S_1} it gives the optimum power allocation scenario.

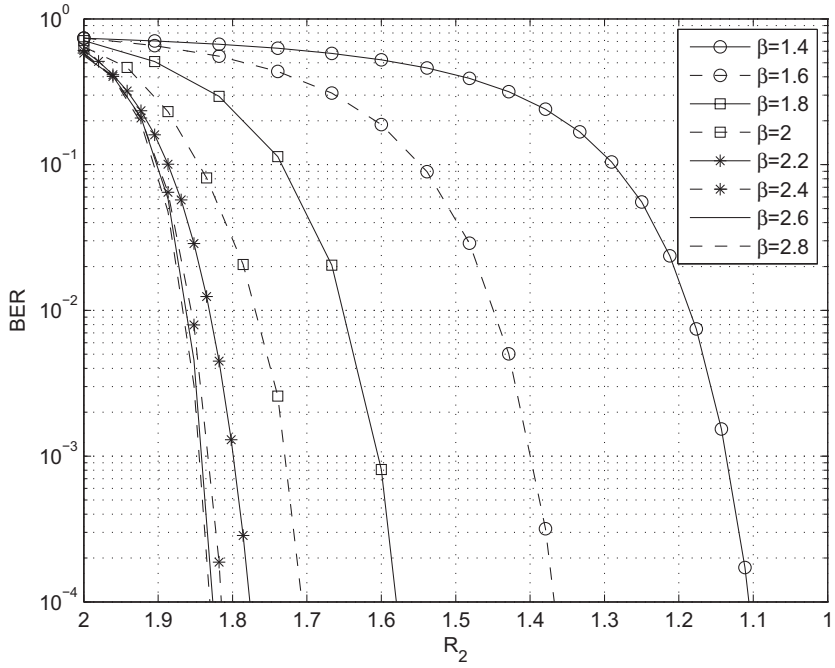

Figure 4.7: Interfering channel performance for unequal power scenario for different β with $E_{S_1} = 4.5$

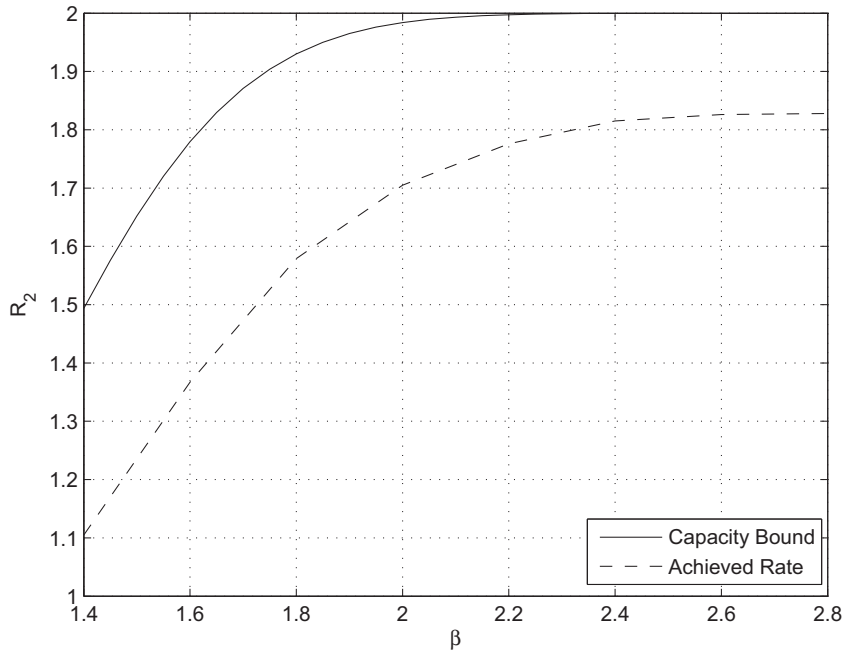

Figure 4.8: Interfering channel achievable Rates for unequal power scenario for different β with $E_{S_1} = 4.5$

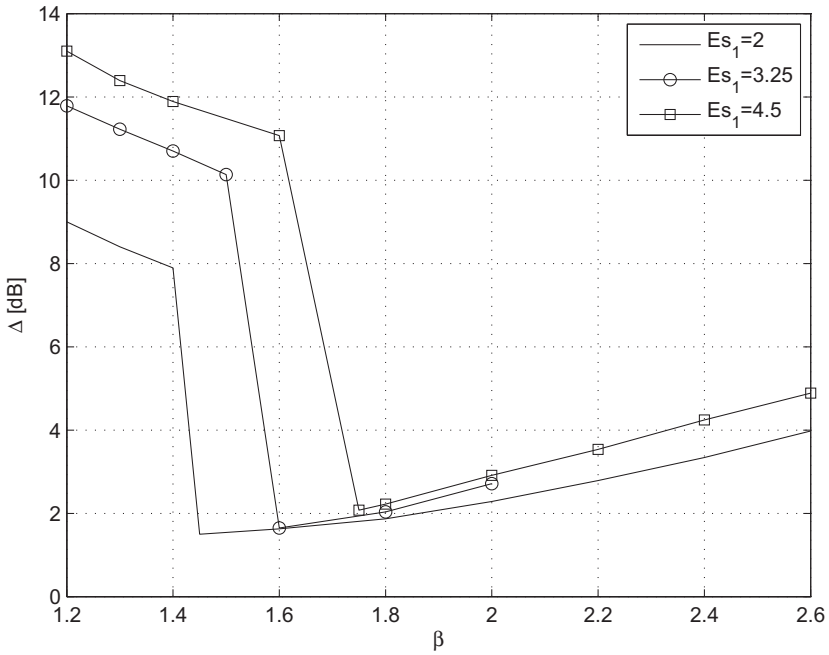

Figure 4.9: Difference of E_b/N_0 and E_b'/N_0 corresponding to capacity

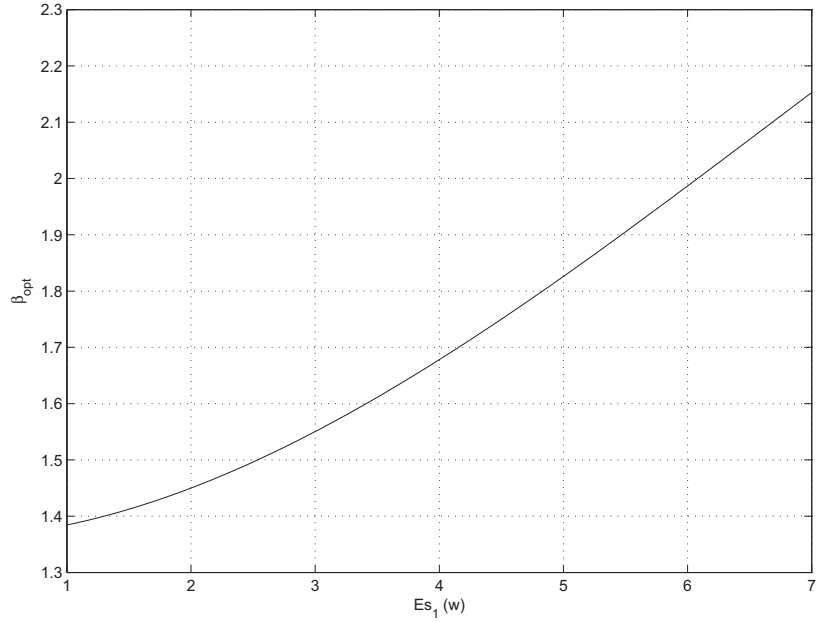

Figure 4.10: Optimum power difference between the two sources for different values of E_{S_1}

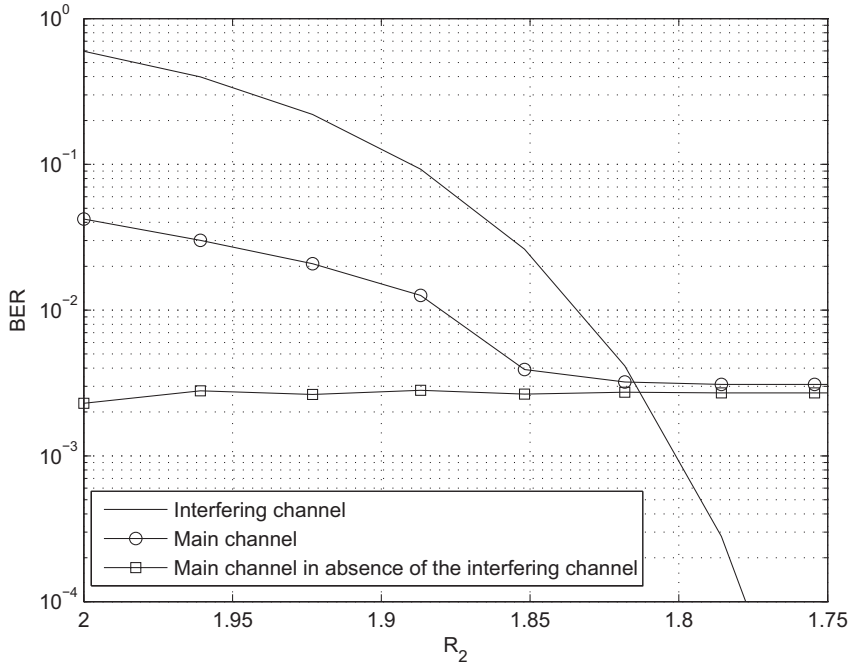

Figure 4.11: Main and interfering channel performances for $\beta = 1.75$ and $E_{S_1} = 4.5$

Fig. 4.11 shows the BER of the main and the interfering channels for $\beta = 1.75$ and $E_{S_1} = 4.5$. Note that since the main channel is transmitting an uncoded data stream its performance is so poor. After some point the adverse effect of the interfering channel on the main channel vanishes and the main channel performs as though there were no interfering channel. We can conclude that the interfering source does not have any effect on the performance of the main source.

The achievable rate from (4.14) and (4.15) for $\beta = 1.75$ and $E_{S_1} = 4.5$ can be written as:

$$R_1 \leq C_1(4.5, 0.5) = 1.9998 \tag{4.43}$$

$$R_2 \leq C_2(4.5, 13.78, 0.5) = 1.9945 \tag{4.44}$$

Compared to achieved rate of this scheme $R_2' \simeq 1.778$ (from Fig. 4.11) it can be concluded that this scheme has achieved 89% of the interfering channel capacity.

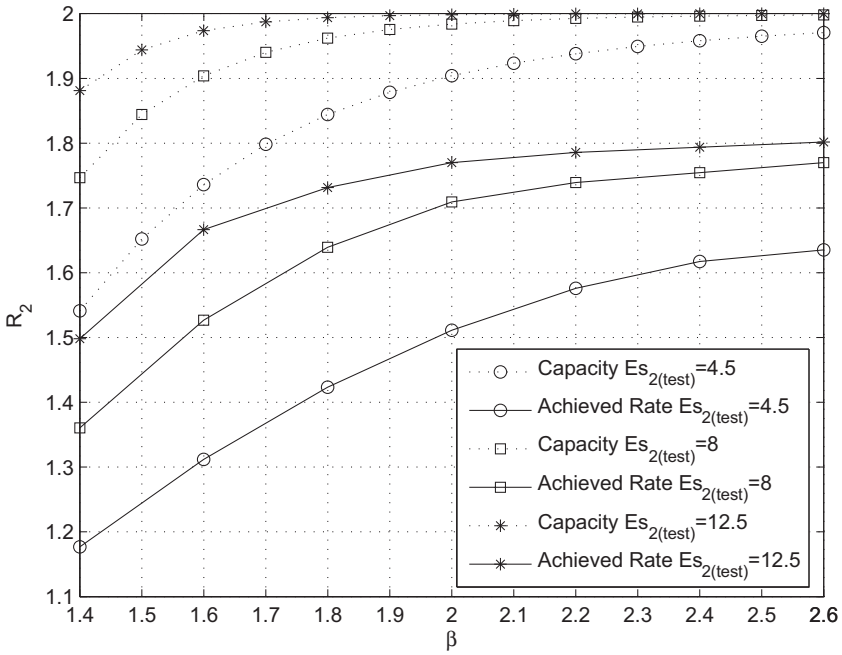

Figure 4.12: Interfering channel achievable rates for different $E_{S_2(test)}$ levels

Furthermore, we have increased the total throughput. Without the interfering source, again the total rate would be $R = R_1' = 1.38$, while with our scheme it will increase to $R = R_1' + R_2' = 1.38 + 1.778 = 3.158$ which shows 128% increase of the throughput.

4.5.4 Power Adaptation

As mentioned earlier, for power adaptation we use achievable rate curves for the interfering source at a specific noise power and under different power allocation scenarios, simulated *a priori*. Fig. 4.12 shows these simulated curves for different E_{S_2} levels at $\sigma = \sqrt{0.5}$.

For example, consider that the interfering source starts transmitting with a test power of $E_{S_2(test)} = 8$. After receiving feedback for the first codeword, let's say it calculates that the transmission rate is $R_2 = 1.6$. From Fig. 4.12, it is evident that for $E_{S_2(test)} = 8$ this means $\beta = 1.73$. From (4.2) it can be calculated that $E_{S_1} = 2.67$.

Now that we have an estimation of E_{S_1} we can find the optimum power difference

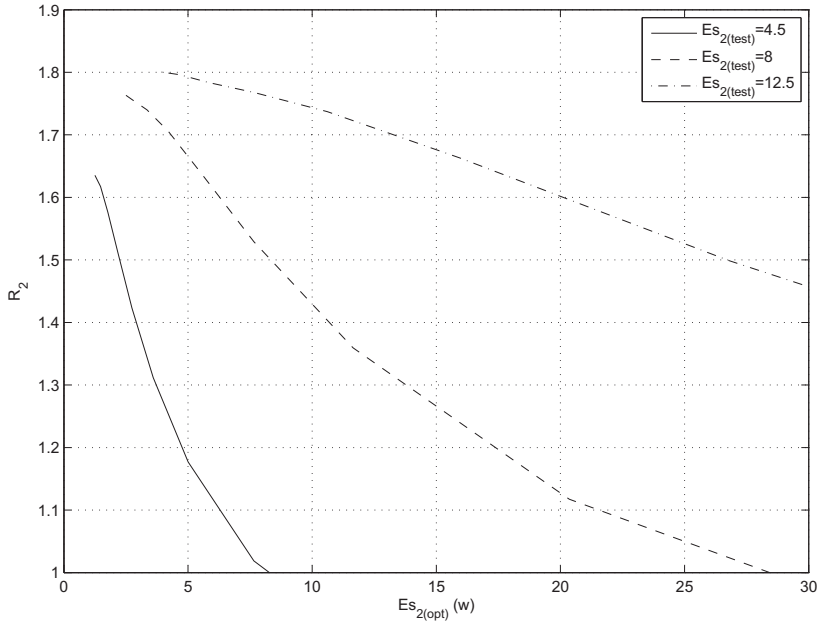

Figure 4.13: Optimum E_{S_2} for different interfering channel rates and test powers

between the two sources (β_{opt}) and adjust the transmission power for the interfering source. For this calculation we use Fig. 4.10. From this curve, which displays the optimization results of (4.32), we can find the β_{opt}, and using (4.2) we can find the new value for $E_{S_2(opt)}$. Here, for $E_{S_1} = 2.67$, from Fig. 4.10 it can be seen that $\beta_{opt} = 1.53$ and from (4.2) we derive the optimum value of $E_{S_2(opt)} = 6.28$. Therefore, instead of using $E_{S_2(test)} = 8$, from now on we will use $E_{S_2(opt)} = 6.28$, which according to the criteria (4.32), is the optimum power for this scenario.

For the sake of simplicity, steps 5-8 of the algorithm in section 4.4.3 can be merged into one step. If we draw R_2 against $E_{S_2(opt)}$ we can skip between steps. Fig. 4.13 shows that for each test power level and any given R_2, what is the optimum value for E_{S_2}. In the previous example where R_2 was 1.6 and the test power was $E_{S_2(test)} = 8$, Fig. 4.13 can be used to show that $E_{S_2(opt)} = 6.28$, which is the same value that was calculated previously. Thus the calculation of $E_{S_2(opt)}$ from R_2 has been reduced to one step. These curves can be expressed with tables that show for each $E_{S_2(test)}$ and

any given R_2 what the value of $E_{S_{2(opt)}}$ is.

4.6 Complexity Order

It has been shown the encoding of Raptor codes on BEC have linear complexity [89]. This is the case also for AWGN since the encoding process does not depend on the channel type. However, for AWGN channels, in [105] the decoding complexity of LDPC and LT layers of Raptor code and the overall complexity was found to be:

$$C^p_{LDPC} = I_{LDPC} N_e (c_{\tanh} + c_{\text{atanh}}) + I_{LDPC}(4N_e - 2k + N_i)c \tag{4.45}$$

$$C^p_{LT} = I_{LT} n_p (c_{\tanh} + c_{\text{atanh}}) + I_{LT} n_p (4a - 2\frac{N_i}{n_p} - 1)c \tag{4.46}$$

$$C_{Raptor} = \sum_{p=1}^{f_{de}} C^p_{LDPC} + C^p_{LT} \tag{4.47}$$

where I_{LT} and I_{LDPC} are the number of BP iterations for LT and LDPC layers, respectively, k is the number of source symbols, N_i is the number of intermediate symbols, n is the codeword length, N_e is the number of LDPC tanner graph edges, a is the average degree of each check node and constant, p is the number of the decoding attempt, n_p is the codeword length at decoding attempt p, f_{de} is the total number of decoding attempts and finally, c_{tanh}, c_{atanh} and c are the computational complexity for hyperbolic tangent, inverse hyperbolic tangent, and basic operations, respectively.

The total computational complexity order of Raptor code per decoding attempt can be summarized as $O(I_{LT}(n_p + N_i) + I_{LDPC}(N_e + k + N_i))$. In our work all of the above variables are fixed, except for n_p. Hence the overall complexity can be written as $C_{Raptor} = \sum_{p=1}^{f_{de}} L n_p$ where L is a constant. n_p can be written as $n_p = k + pI$ where I is the number of extra bits received between two decoding attempts. From (2.20) we have $n = n_{f_{de}} = k(1+\epsilon) = k + k\epsilon$. Hence I can be written $I = k\epsilon / f_{de}$. The overall

complexity can be written as:

$$C_R = \sum_{p=1}^{f_{de}} Ln_p = L\sum_{p=1}^{f_{de}} k + pI = L(kf_{de} + I\sum_{p=1}^{f_{de}} p) = L(kf_{de} + \frac{If_{de}(1+f_{de})}{2}) \quad (4.48)$$

Hence, the complexity order can be written as $O(I{f_{de}}^2)) = O(I(k\epsilon/I)^2) = O(\frac{(n-k)^2}{I})$. Therefore we can conclude that the decoding of Raptor codes over AWGN in terms of its codeword has a complexity order of $O(n^2)$.

At the destination node, as shown in Fig. 4.3, there is a decoding/encoding pair for Raptor code and the decoder for the main source. Since the latter already existed in the link, we will not consider it here. The encoder and decoder have polynomial complexity $O(n)$ and $O(n^2)$, as discussed above. At the interfering source node, the encoder again has linear complexity. The power adaptations depends on tables that have been simulated beforehand and, the interfering source will just look up the table for the optimum power level, and this will not add to the complexity order.

Chapter 5

Successive Decoding with Constellation Rotation

5.1 Introduction

In this chapter, we will investigate a scheme to increase the capacity of an existing DVB-RCS channel with the addition of an interfering source. Consider an existing DVB-RCS link between a main source and a destination. As mentioned in the previous chapter, the idea is to transmit data from another source, an interfering source, on the same channel to the same destination. This interfering source is symbol synchronized with the main source and, since it is using the same channel, the channels between the two sources and the destination are non-orthogonal. Thus, at the destination, their mixed codewords need to be separated and decoded. This is done through successive decoding. The interfering channel uses Raptor coding.

We can extend the work of the previous chapter and discuss successive decoding in a more general manner. In a departure from the previous chapter, where we assumed that in successive decoding the interfering source data is decoded first, here we propose a scheme for decoding the main source data first. Decoding the interfering

source data first results in some delay for the main source data, since the interfering data source has to be decoded, subtracted from the received stream, and then fed into the main decoder so that the main source data can finally be decoded.

In this chapter, we investigate a case where the main source data is decoded first and the delay is transferred to the interfering source data. However, we demonstrate that although decoding the main source data first it eliminates the delay for the main source, this procedure diminishes decoding performance. There is a tradeoff between delay and performance for the main source. We will demonstrate that in specific power allocation scenarios, this deterioration in the performance of the main source is negligible, while the corresponding delay is eliminated completely.

Furthermore, in some power allocation schemes, the symbols of the sources cancel one another. To address this problem we can use constellation rotation for the interfering source. We will calculate the average distance between the points in the constellation resulting from the superposition of the main and interfering sources' constellations for different power scenarios. We will also find the optimum rotation angle for each case. An average optimum rotation angle will also be found for the general case.

5.2 System Model

Consider a main source S_1 transmitting to destination D with QPSK modulation over an AWGN channel. Source S_2, the interfering source, starts transmitting on the same channel with the same modulation. The main source uses DVB-RCS, MPEG profile with RS(204,188), and CoC with rate 3/4, 2/3 or 1/2. The interfering source can use the same FEC as the main source, followed by an extra Raptor encoding layer, but for simplicity we will assume that it directly encodes its data with Raptor code.

We can assume that the transmitters are symbol synchronized. For both of the

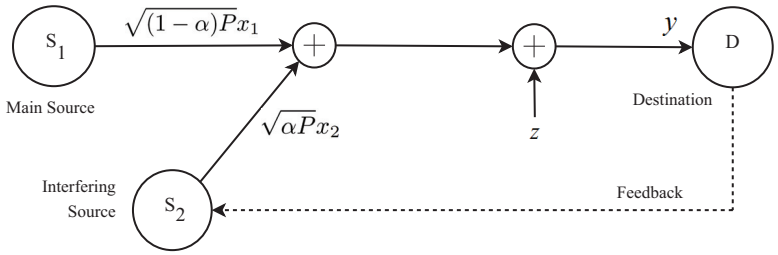

Figure 5.1: System Model

sources we define codewords x_i where $E\left[{x_i}^2\right] = 1$. We assume that the total power P is constant and divided between two sources with the power allocation ratio α $(0 < \alpha < 1)$. The destination receives the following:

$$y = \sqrt{(1-\alpha)P}x_1 + \sqrt{\alpha P}x_2 + z \tag{5.1}$$

where z is a circular symmetric complex Gaussian random vector, $z \sim CSCG(0, \sigma^2)$. $\frac{\sigma^2}{2}$ is the variance of AWGN in each dimension. Fig. 5.1 shows the system model. Note that in the previous chapter, in contrast to this stage of our investigation, the total power is not fixed.

Through successive decoding, the destination tries to decode both data streams. The decoder can either decode the main or the interfering source data first. In both cases, after decoding the data and regenerating its original transmitted codeword, it subtracts the regenerated data from the received data stream to get the channel data of the other source. This data is sent to its corresponding decoder.

There is a feedback channel between the destination and the interfering source. The destination sends an acknowledgment every time it decodes a codeword of the interfering source successfully, and the interfering source then proceeds to the next codeword.

5.3 Constellation Constrained Capacity for Two Source MAC

We already calculated the constellation constrained capacity for a similar MAC in Section 4.3. Here, the only change in the system model is the power allocation, which was discussed in previous section. From (5.1) and (4.12), the achievable rate of the interfering channel with MPSK modulation is as follows:

$$R_2 \leq I(x_2; y) = \log_2 M - \frac{1}{M^2} \sum_{k_1=0}^{M-1} \sum_{k_2=0}^{M-1} E\left[\log_2\left(\psi_2\right)\right] \tag{5.2}$$

where:

$$\psi_2 = \frac{\sum\limits_{i_1=0}^{M-1} \sum\limits_{i_2=0}^{M-1} e^{-\frac{\left|\sqrt{(1-\alpha)P}(x_1(k_1)-x_1(i_1))+\sqrt{\alpha P}(x_2(k_2)-x_2(i_2))+z\right|^2}{\sigma^2}}}{\sum\limits_{i_1=0}^{M-1} e^{-\frac{\left|\sqrt{(1-\alpha)P}(x_1(k_1)-x_1(i_1))+z\right|^2}{\sigma^2}}}$$

where $E[x]$ is the expectation with respect to the distribution of z. For the main channel from (4.13) it follows:

$$R_1 \leq I(x_1; y|x_2) = \log_2 M - \frac{1}{M} \sum_{k_1=0}^{M-1} E\left[\log_2\left(\psi_1\right)\right] \tag{5.3}$$

where:

$$\psi_1 = \frac{\sum\limits_{i_1=0}^{M-1} e^{-\frac{\left|\sqrt{(1-\alpha)P}(x_1(k_1)-x_1(i_1))+z\right|^2}{\sigma^2}}}{e^{-\frac{|z|^2}{\sigma^2}}}$$

The constellation constrained capacities for MPSK modulation can be computed

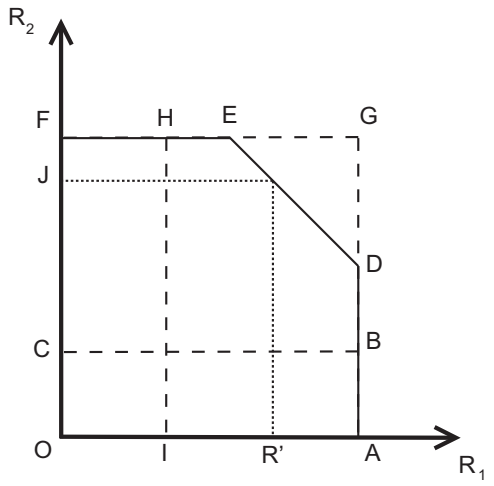

Figure 5.2: Constellation constrained capacity regions for different power allocations

using (5.2) and (5.3). It can be seen that both equations are a function of the power allocation ratio, α.

Depending on α, the constellation constrained region changes as shown in Fig. 5.2. Here, as α increases, the capacity region goes from the line OA (for $\alpha = 0$) to $OABC$. This rectangle expands vertically and reaches its maximum in $OAGF$. Later it shrinks diagonally to $OADEF$ (for $\alpha = 0.5$) and again expands to $OAGF$. Finally it start to shrink again, but this time horizontally to $OIHF$ until it becomes the line OF (for $\alpha = 1$).

If we assume that the rate of the main channel is fixed at $R_1 = R'$, then, as α increases, the maximum value of R_2 is increased from 0 to the level corresponding to F, falls back to the level J and again rises and stays at F, until R' falls out of the capacity region (outage). The solid line in Fig. 5.3 shows R_2 as α goes from zero to one. Here the main channel rate is fixed at $R_1 = 1.2$ while the total power is $P = 10$ and $\sigma^2 = 1$. As mentioned before, the interfering source should not have any effect on the main source. Hence, for $\alpha > 0.95$ - since the main channel rate is in outage - we do not consider the achieved rate R_2 to be practical.

From Fig. 5.3 it can be seen that around $\alpha = 0.5$, where the power of both channels is close, R_2 declines. This is due to the fact that with equal powers, symbols

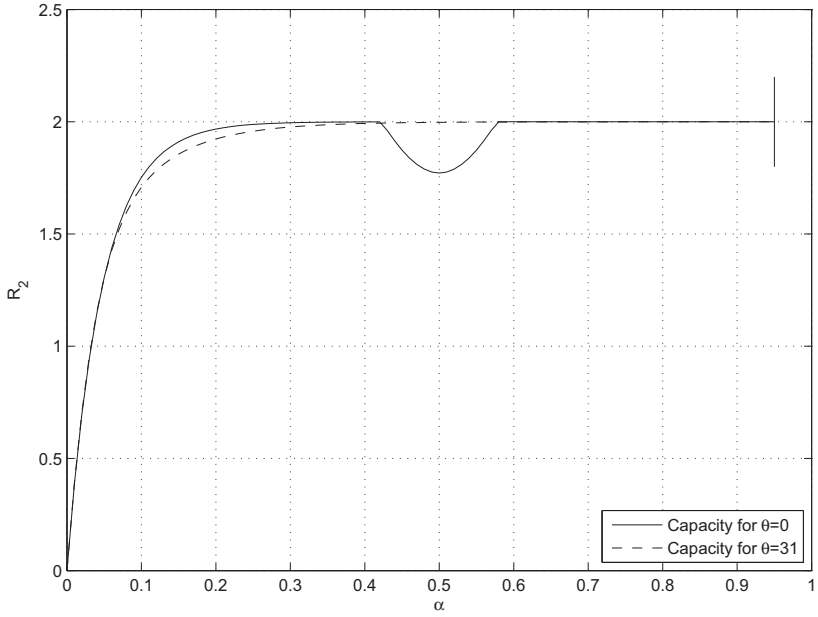

Figure 5.3: Capacity of the interfering channel for fixed $R_1 = 1.228$

from two sources may cancel each other. For example, in the QPSK constellation, if one source transmits symbol $\sqrt{P/2}$ and the other transmits $-\sqrt{P/2}$, the destination only receives channel noise and all four bits (two from each source) are erased.

It is clear that if one of the sources rotates its constellation appropriately this loss of rate can be compensated for. Without any loss of generality, we can assume that the interfering source constellation is rotated. If we define the symbols of rotated constellation x_2' then: $x_2' = x_2 e^{j\theta}$.

For capacity analysis, (5.3) does not change since it does not have any terms with x_2. However, (5.2) is updated with x_2' instead of x_2:

$$R_2 \leq I(x_2'; y) = \log_2 M - \frac{1}{M^2} \sum_{k_1=0}^{M-1} \sum_{k_2=0}^{M-1} E\left[\log_2(\psi_2)\right] \tag{5.4}$$

$$\psi_2 = \frac{\sum_{i_1=0}^{M-1} \sum_{i_2=0}^{M-1} e^{-\frac{\left|\sqrt{(1-\alpha)P}(x_1(k_1)-x_1(i_1))+e^{j\theta}\sqrt{\alpha P}(x_2(k_2)-x_2(i_2))+z\right|^2}{\sigma^2}}}{\sum_{i_1=0}^{M-1} e^{-\frac{\left|\sqrt{(1-\alpha)P}(x_1(k_1)-x_1(i_1))+z\right|^2}{\sigma^2}}} \tag{5.5}$$

In order to find maximum capacity for R_2, we have to find the analytical optimum value of θ such that:

$$\theta^* = \arg \max_{\theta \in (o, 2\pi)} R_2 = \arg \max_{\theta \in (0, 2\pi)} I(x_2'; y) \tag{5.6}$$

In [100] it was shown that for high SNR values, (5.6) can be approximated as follows:

$$\theta^* = \arg \min_{\theta \in (0, 2\pi)} M(\theta) \tag{5.7}$$

where:

$$M(\theta) = \sum_{k_1=0}^{M-1} \sum_{k_2=0}^{M-1} \log_2 \left(\sum_{i_1=0}^{M-1} \sum_{i_2=0}^{M-1} \psi_M \right) \tag{5.8}$$

$$\psi_M = e^{-\frac{\left| \sqrt{(1-\alpha)P}(x_1(k_1) - x_1(i_1)) + e^{j\theta}\sqrt{\alpha P}(x_2(k_2) - x_2(i_2)) \right|^2}{2\sigma^2}}$$

Fig. 5.4 shows (5.8) for $\theta \in (0, \pi/2)$ while $P = 10$, $\sigma^2 = 1$ and $\alpha = 0.5$. This curve repeats itself for the next quarters. It can be seen that in this case θ^* has two optimum values at 31 and 59 degrees. The dashed line in Fig. 5.3 shows the maximum capacity for R_2 (5.4) while $\theta = 31°$. It can be seen that with constellation rotation, the drop around $\alpha = 0.5$ is compensated for, while there is a small decline in capacity for low values of α.

Therefore, the capacity for both channels with or without rotation can be found from (5.3) and (5.4). For simplicity, in this chapter we refer to them as follows:

$$R_1 \leq C_1(\alpha, P, \sigma^2) \tag{5.9}$$

$$R_2 \leq C_2(\alpha, P, \theta, \sigma^2) \tag{5.10}$$

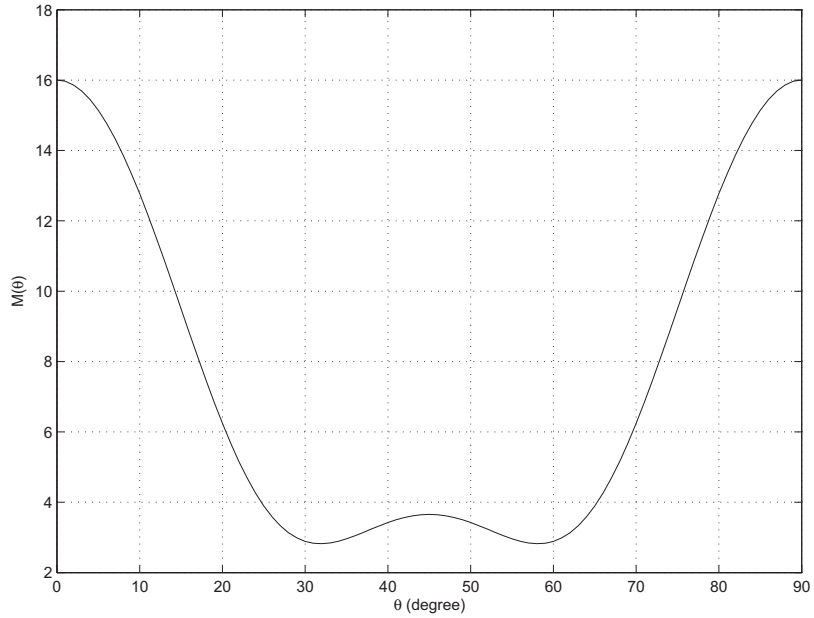

Figure 5.4: Optimum rotation angle for achieving maximum capacity for R_2

5.4 Successive Decoding

Since the main and the interfering sources are transmitting over the same channel, the signals received by the destination consist of the codewords transmitted by both sources. Successive decoding is used to separate these two signals. In the previous chapter, we assumed that the interfering source had higher power and was decoded first. Later, its codeword was regenerated and subtracted from the original received signal to obtain the data of the main source, plus noise. We call this an Interfering Source First (ISF) scheme.

As mentioned earlier, the interfering source uses Raptor code which is a rateless code. When the interfering data is first decoded, the decoding of the main codeword is delayed until after the successful decoding of the interfering source.

As shown in Fig. 5.5, due to channel conditions it is possible that the interfering source codeword will become decodable only after receiving more symbols than the length of the main source codeword. This results in a cumulative delay. Eventually

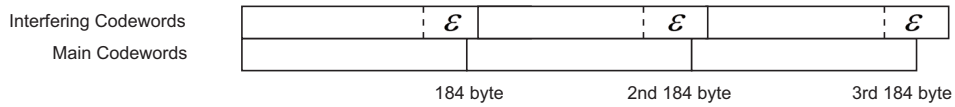

Figure 5.5: Codewords from the main and interfering sources

the time gap between the received codeword from the source and the just-decoded codeword will increase. If the main source codeword is decoded first, the problem with the delay of the main source data is solved and the delay is transferred to the interfering source data.

Therefore, we can extend this work to a situation where the data from the main source can be decoded first and the same successive procedure is carried out for the interfering source. We call this a Main Source First (MSF) scheme. The main advantage of this method is its shorter delay for the main source. However, when $\alpha > 0.5$ (i.e., the main source power is less than the power of the interfering source) this method diminishes the decoding performance of the main source, since the data from the interfering source - although more powerful - has not been removed from data stream, and thus has an adverse effect on the decoding process of the main source. Nevertheless, although there is a reduction in the achievable rate of the main channel, the achieved rate of the interfering channel increases. Overall, we can show that, for some specific power allocation scenarios, the total rate achieved by both sources in MSF is close to ISF, though it introduces less of a delay.

In both of the above scenarios, the achievable rate for both channels is highly dependent on how the total power is divided between the two sources, i.e. on the value of the α in (5.1). As α increases, more power is allocated to the interfering source and its achieved rate is therefore increased. Around $\alpha = 0.5$, symbols from both sources cancel out each other and the rate decreases, but when the power difference is increased again the rate increases too. We can demonstrate this with simulations.

In our simulations, a rate 0.98 right-regular LDPC code [89] with $k = 1472$ is chosen as the pre-code. The codeword length of 1472 comes from the MPEG packet

length in DVB-RCS, which has a payload of 184 bytes (or 1472 bits). For the LT layer the distribution $\Omega(x)$ is the distribution for $k = 65536$ case in Table I at [89]:

$$\begin{aligned}\Omega(x) &= 0.008x + 0.49x^2 + 0.166x^3 + 0.072x^4 + 0.083x^5 + 0.056x^8 + 0.037x^9 \\ &+ 0.056x^{19} + 0.025x^{66} + 0.003x^{67} \end{aligned} \tag{5.11}$$

The above degree distribution is optimized for the erasure channels. In [92] it was shown that in AWGN channels, Raptor codes lose their generality and for each value of σ a specific Raptor code should be designed. However, it was demonstrated that although the degree distribution for the erasure channels is not optimized for AWGN channels (leaving room for improvement) it performs very well and its achieved rate is acceptable. Therefore, for the sake of simplicity, we used the BEC degree distribution here.

In the destination decoder, both of the decoders employ belief propagation decoding with iteration numbers of 50 and 300 for the LDPC layer and LT layer, respectively. The modulation is QPSK (as it is defined in DVB-RCS standard) and $P = 10$, $\sigma^2 = 1$. The main source uses a concatenated code of RS(204,188) and CoC(3/4), hence $R_1 = 1.382$. Fig. 5.6 shows the interfering channel achieved rate and BER for different α value with MSF scheme. It can be seen that there is a drop in the achieved rate around $\alpha = 0.5$, as expected. Besides that drop, as α increases, R_2 is increased too.

Fig. 5.7 shows the achievable rates for both schemes more clearly. In this figure, as well as in all other simulations, the achieved rate is the rate at which the channel code guarantees a BER not exceeding 10^{-4}. Here we have shown both the interfering channel's achievable rate (R_2) and the total achievable rate ($R_1 + R_2$) for ISF and MSF.

In the case of ISF, the interfering source is decoded first. The destination requests

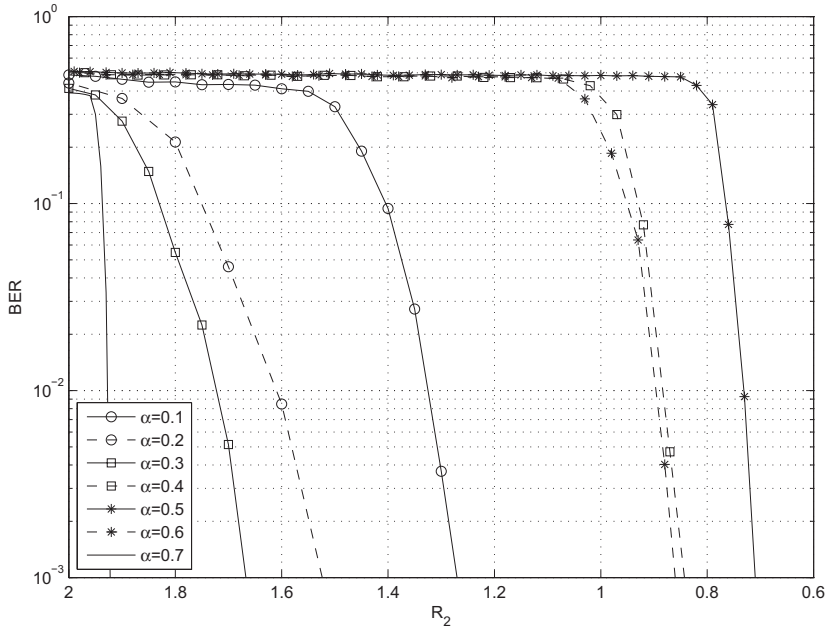

Figure 5.6: BER for the interfering source for different power allocation scenarios with MSF scheme

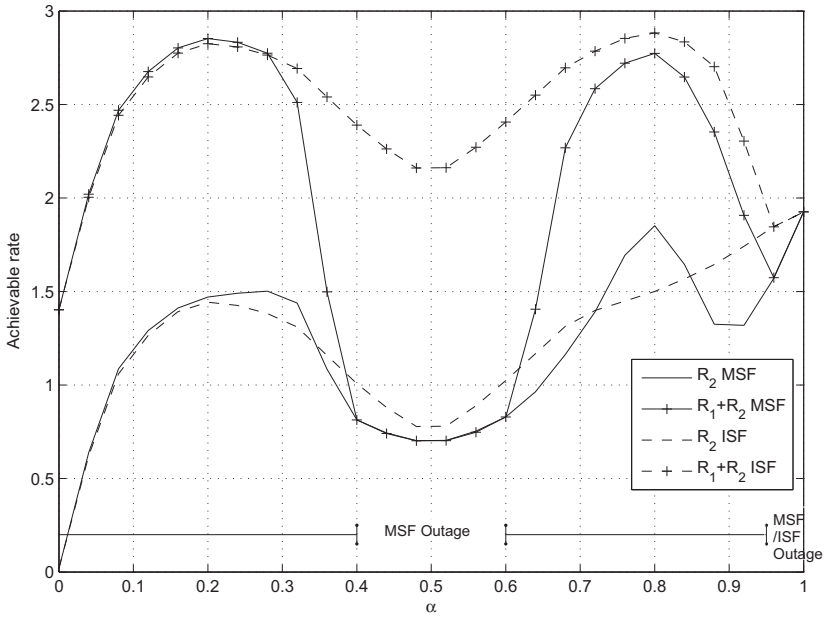

Figure 5.7: Total and interfering channel achievable rates with MSF and ISF schemes

R_1	RS code	CoC rate	α range
1.382	(204,188)	3/4	$\alpha < 0.2$
1.229	(204,188)	2/3	$0.2 \leq \alpha < 0.4, 0.6 \leq \alpha < 0.7$
1.092	(204,188)	1/2	$0.7 \leq \alpha < 0.9$
0	outage	outage	$0.4 \leq \alpha < 0.6, \alpha > 0.9$

Table 5.1: The changes of the main channel rate in MSF

as many symbols as it needs from the interfering source for the correct decoding of its data. Therefore, the interference can be removed perfectly, and this does not have any effect on the main source rate R_1. For the total achievable rate, the curve for R_2 is added with $R_1 = 1.382$, except for $\alpha > 0.95$ where the allocated power for the main source is too low and it is in outage.

However, for the MSF this is not the case. As mentioned before, when $\alpha > 0.5$ the decoding performance of the main source decreases. In order to compensate for this, we have used stronger (lower rate) codes for the main source to guarantee the same error-free transmission. Thus, the CoC rate can be reduced from 3/4 to 2/3 or 1/2 which corresponds to rate changing from $R_1 = 1.382$ to $R_1 = 1.229$ or $R_1 = 0.921$, respectively. If even with CoC(1/2) error-free transmission is not possible, an outage has occurred and R_1 is set to zero. Table 5.1 shows the changes of R_1 for the MSF in Fig. 5.7.

Therefore, although compared to ISF, MSF has achieved a higher rate for the interfering channel around $\alpha = 0.8$, because of decrease in R_1, the total achieved rate is still less than ISF.

It should also be noted that for $\alpha < 0.4$ and $0.7 < \alpha < 0.8$ the performance of MSF is close to ISF, though no delays are caused. Hence, in these power ranges, ISF can be replaced by MSF when a delay effect is not desired.

5.5 Constellation Rotation

We have shown that when the power level of the sources is close (i.e. α is close to 0.5), the capacity (as seen is Fig. 5.3) and the achievable rate (as seen in Fig. 5.7) both decrease. As mentioned before, this is due to the fact that some symbols from each source cancel the corresponding symbols from the other source. For example, consider a case where $\alpha = 0.5$. Fig. 5.8 shows the constellation map of the received signal (without considering noise) for a QPSK system when the sources have equal powers ($\alpha = 0.5$).

Fig. 5.9 shows the demodulation decision regions of the received signal for $\alpha = 0.5$ and $P = 2$. Here, each color defines the corresponding regions for each of the main source symbols. In some regions, these decision regions overlap. If the source symbols are equiprobable, a quarter of the symbols received at the destination are zero. For these symbols, one of the sources has transmitted $\pm\sqrt{0.5P}$ or $\pm j\sqrt{0.5P}$ and the other source has sent its negative. Therefore, both symbols are lost. In some cases the destination looses one bit from each symbol (for example, when the destination receives $\sqrt{0.5P} + j\sqrt{0.5P}$).

The erasure occurs when two symbols are too close to each other or overlap in the constellation map. By rotation, we can put some distance between these symbols and eliminate the decision region overlaps. Fig. 5.10 and 5.11 show the received signal constellation map with $\alpha = 0.5$, $P = 2$ and rotation angles of $\theta = 35°$ and $\theta = 45°$, respectively. Again, each color defines the corresponding regions for each of the main source symbols. It can be seen that in both rotations signal erasure is avoided and the decision regions do not overlap anymore.

Without considering noise interference, (5.1) can be expanded as follows:

$$x_1(m) = e^{j\frac{m\pi}{2}} \tag{5.12}$$

$$x_2'(n) = e^{j(\frac{n\pi}{2}+\theta)} \tag{5.13}$$

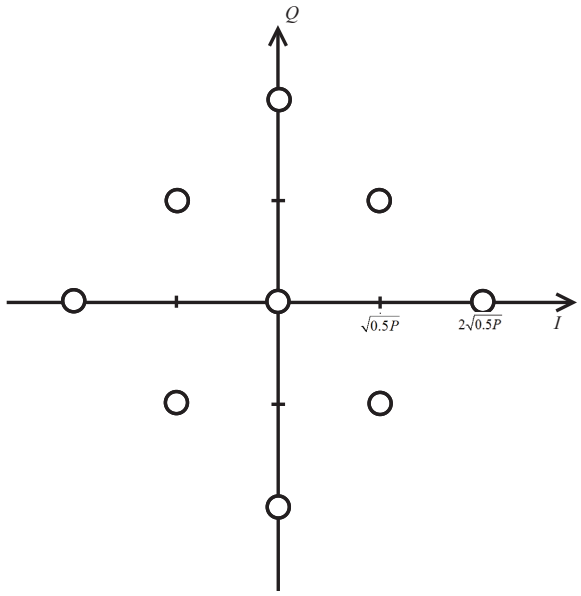

Figure 5.8: Symbol erasure for sources with equal power and non-orthogonal channels

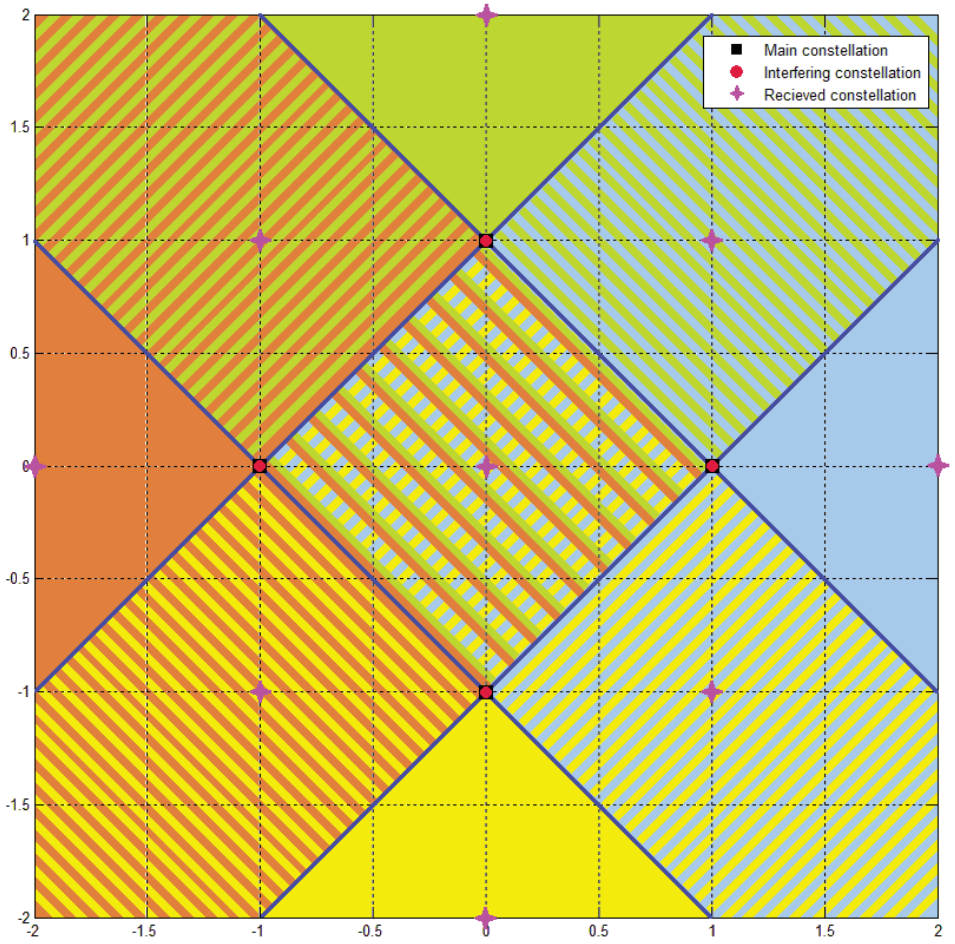

Figure 5.9: Decision regions of the received signal for $\alpha = 0.5$ and $P = 2$

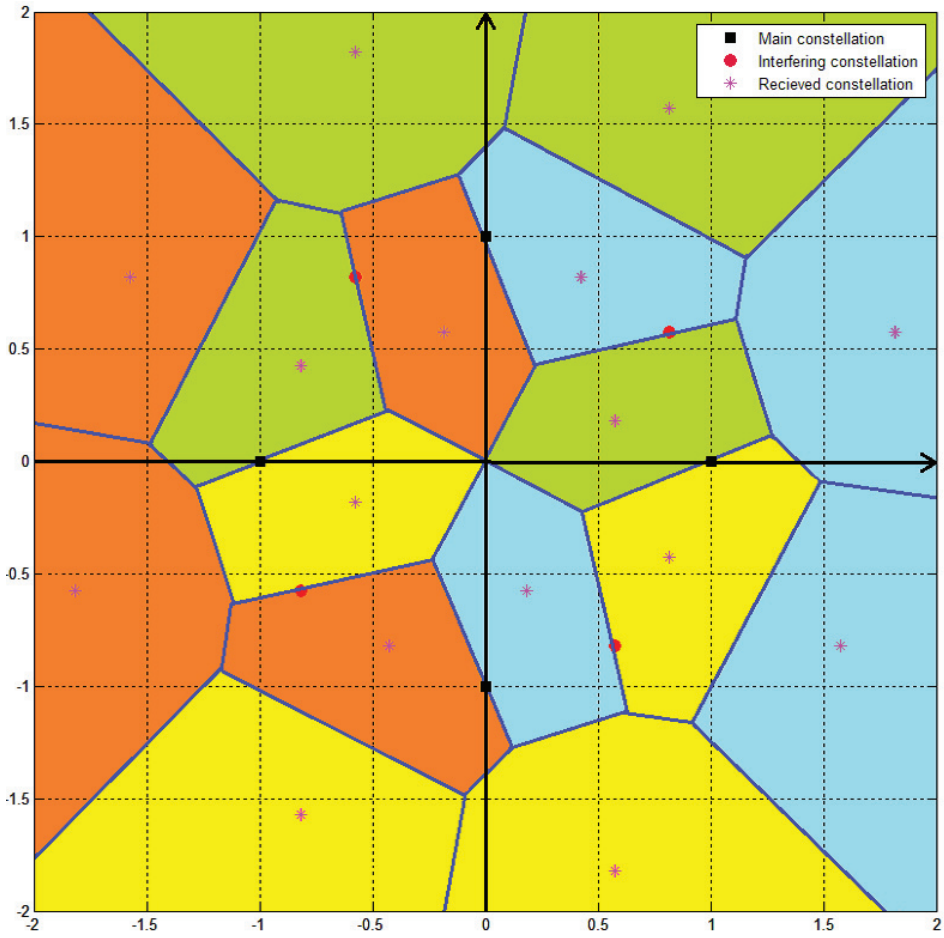

Figure 5.10: Decision regions of the received signal for $\alpha = 0.5$, $P = 2$ and $\theta = 35^{\circ}$

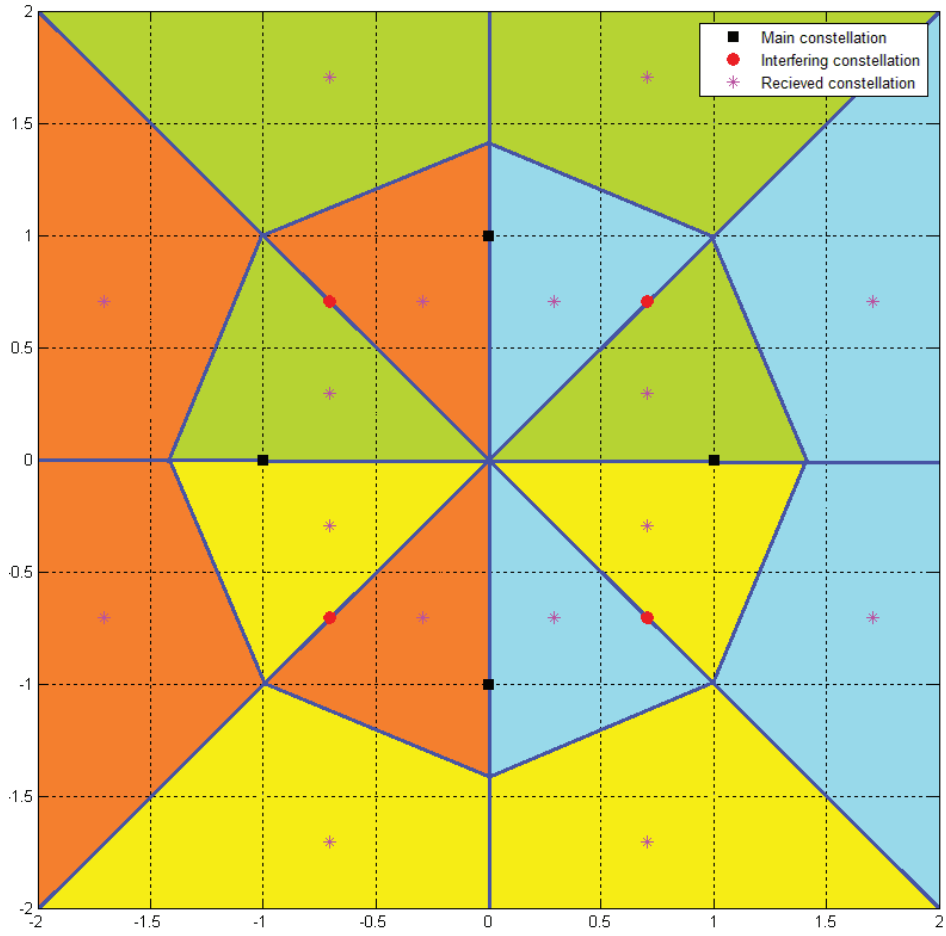

Figure 5.11: Decision regions of the received signal for $\alpha = 0.5$, $P = 2$ and $\theta = 45°$

$$
\begin{aligned}
y'_{m,n}(\alpha,\theta) &= \sqrt{(1-\alpha)P}x_1(m) + \sqrt{\alpha P}x_2(n) \\
&= \sqrt{(1-\alpha)P}e^{j\frac{m\pi}{2}} + \sqrt{\alpha P}e^{j(\frac{n\pi}{2}+\theta)} \qquad (5.14)
\end{aligned}
$$

where θ is the rotation angle of the interfering source constellation, y' is the combination of the symbols transmitted by the sources and $m, n = 0, 1, 2, 3$.

The distance between any two received symbols can be written as:

$$
\begin{aligned}
d_{m,n,m',n'}(\alpha,\theta) &= \left|{y'}_{m,n}(\alpha,\theta) - {y'}_{m',n'}(\alpha,\theta)\right| \qquad (5.15) \\
&= \left|\sqrt{(1-\alpha)P}(e^{j\frac{m\pi}{2}} - e^{j\frac{m'\pi}{2}}) + \sqrt{\alpha P}e^{j\theta}(e^{j\frac{n\pi}{2}} - e^{j\frac{n'\pi}{2}})\right|
\end{aligned}
$$

where $m, n, m', n' = 0, 1, 2, 3$.

For any pair α and θ, there are 16 possible values for y' and therefore 256 possible values for $d(\alpha,\theta)$. But we are just interested in the pair with the smallest distance that may eventually cause detection error. Therefore, we find the pair with the minimum distance:

$$
D(\alpha,\theta) = \min_{m,n,m',n'}\{d_{m,n,m',n'}(\alpha,\theta)|m,n,m',n' = 0,1,2,3,(m,n) \neq (m',n')\} \quad (5.16)
$$

$D(\alpha,\theta)$ gives the minimum distance for any α and θ. We can determine what is the optimum rotation angle that maximizes the minimum distance for each power allocation scenario. For any given value of α we can find the optimum rotation angle θ_{opt}:

$$
\theta_{opt}(\alpha) = \arg\max_{\theta\in(0,2\pi)} D(\alpha,\theta) \qquad (5.17)
$$

Fig. 5.12 shows this optimum rotation angle as a function of α. For $\alpha < 0.21$ and $\alpha > 0.79$ the rotation angle does not have any significant effect on the minimum distance and performance of the system. It can be seen that as the power levels of

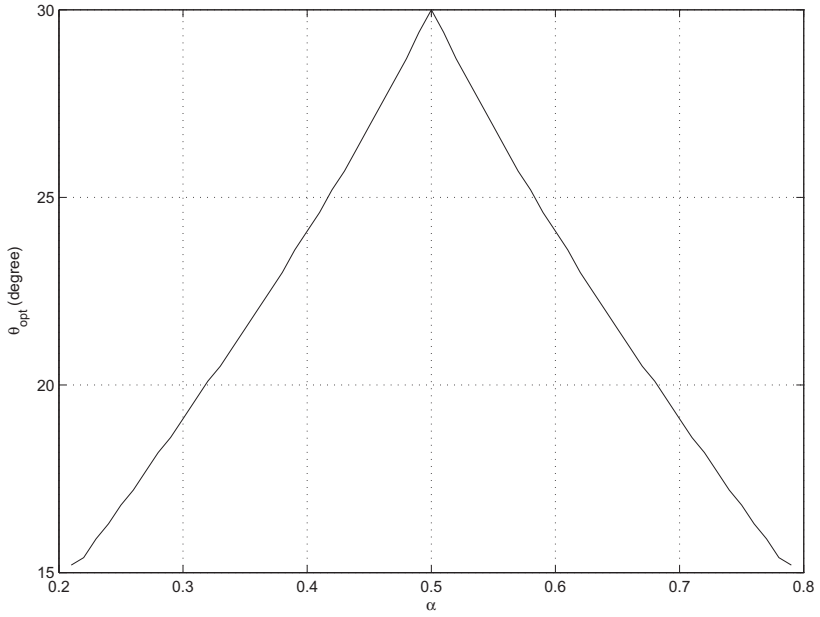

Figure 5.12: Optimum rotation angle for each α maximizing the minimum distance

the two sources get closer, i.e. α gets closer to 0.5, the required rotation angle is increased.

Fig. 5.12 also shows that in the case of $\alpha = 0.5$ we have $\theta_{opt} = 30°$. This is close to the optimal rotation angle from capacity calculations in (5.7), which for the case of $\alpha = 0.5$ in Fig. 5.4 was shown to be $\theta^* = 31°$.

It may not be practical to have a specific rotation angle for each power allocation scenario. It would be easier if we could decide on one rotation angle that would increase the average performance of the system over the whole range of α. For this we have to average the minimum distance on α and find the optimum rotation angle. We average D in (5.16): $(i = 0, 1, ..., t)$

$$\alpha_i = \frac{i}{t} \tag{5.18}$$

$$\bar{D}(\theta) = \frac{1}{n} \sum_{i=0}^{t} D(\alpha_i, \theta) \tag{5.19}$$

$$\bar{\theta}_{opt} = \arg \max_{\theta \in (0,2\pi)} \bar{D}(\theta) \tag{5.20}$$

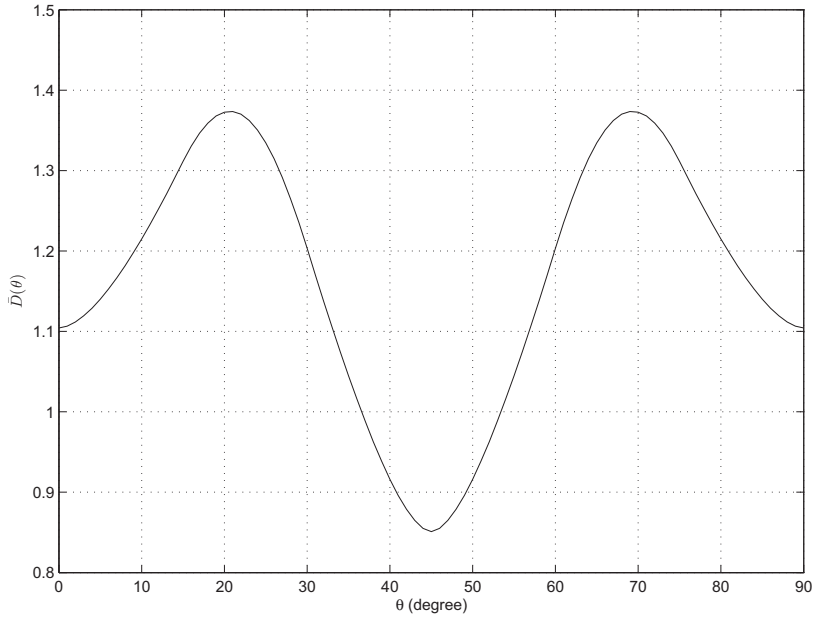

Figure 5.13: Minimum distance between received symbols averaged on α values

where t is the number of values used in taking the average. $\bar{D}(\theta)$ shows what is the average minimum distance for a specific rotation angle and $\bar{\theta}_{opt}$ is the average optimum rotation angle. Fig. 5.13 shows the average minimum distances for $P = 10$. It is evident that in this setup $\bar{\theta}_{opt}$ has two optima in 21° and 69° that maximize the minimum distance and give the highest performance on average.

We have simulated the effect of rotation on both the MSF and ISF schemes. Fig. 5.14 shows this effect when the ISF scheme is chosen and Fig. 5.15 shows the results when the decoding scheme is switched to MSF with the same settings. As before, here $P = 10$ and $\sigma^2 = 1$. In both schemes, the rotation eliminates the rate drop around $\alpha = 0.5$ where the difference between power levels is small. The achievable rates for $\theta = 21°$ and $\theta = 30°$ are in agreement, and the results are shown in Fig. 5.12. The rotation angle $\theta = 30°$ outperforms other rotation angles around $\alpha = 0.5$ while $\theta = 21°$ gives higher achievable rates around $\alpha = 0.3$ and $\alpha = 0.7$.

Note that the bold curves in these figures demonstrate the optimum rotation

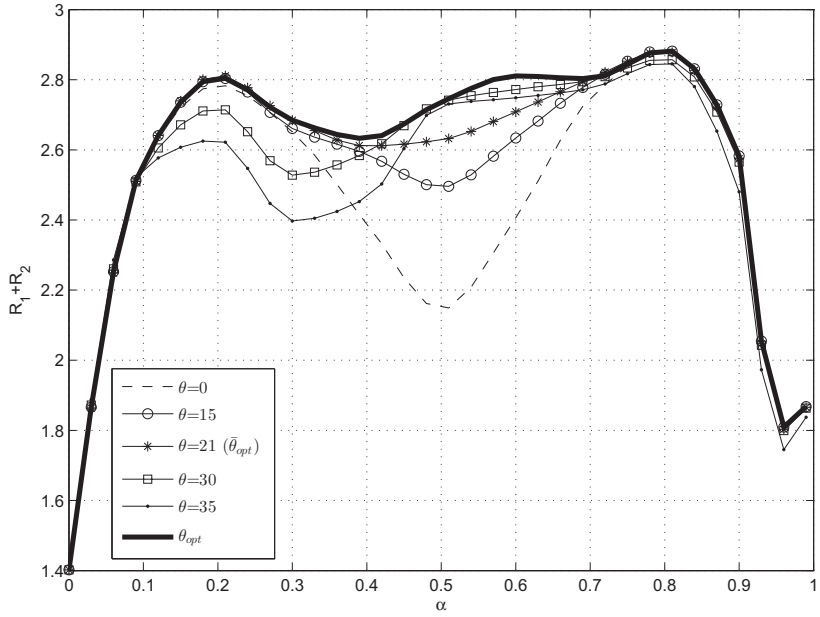

Figure 5.14: Effect of rotation on achievable rates for the interfering source with ISF scheme

degree θ_{opt} from (5.17) and, as expected, deliver the highest achievable rates. It can be seen that these curves are the envelope of all other curves which each represent a fixed rotation angle. In other words, at any particular α, the achievable rate by θ_{opt} is always equal to or higher than the rates achieved by any of the other fixed rotation angles, as if θ_{opt} is jumping from one fixed rotation angle to another as α increases.

Furthermore, if the destination can switch from MSF to ISF when $\alpha = 0.5$, the average optimum rotation degree $\bar{\theta}_{opt} = 21^{\circ}$ gives the highest achievable rate on average, as anticipated by Fig. 5.13. Although the achievable rate by θ_{opt} is higher than $\bar{\theta}_{opt}$, if it is not possible or practical to adjust the rotation angle for each specific power allocation scheme with (5.17), we can use the average method (5.20) and find the average optimum rotation angle without considering α values.

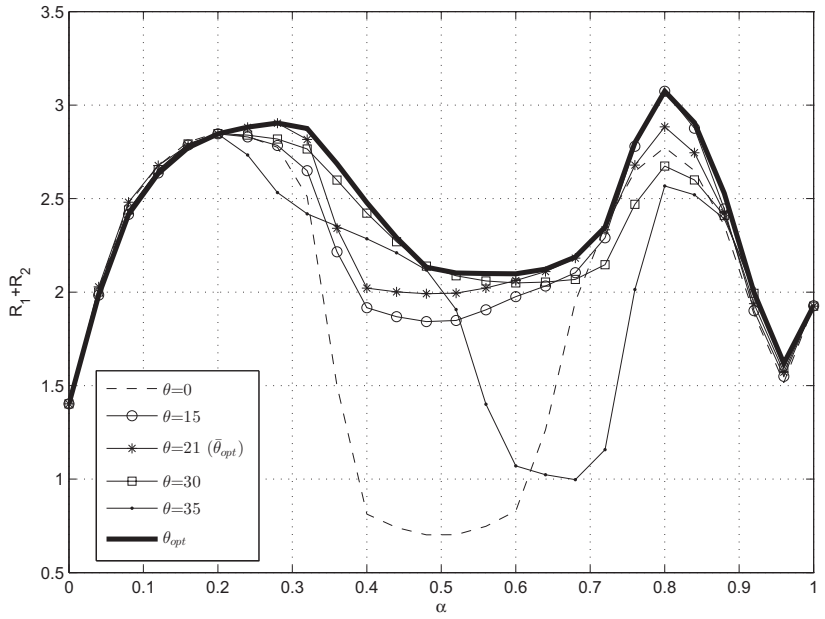

Figure 5.15: Effect of rotation on achievable rates for the interfering source with MSF scheme

5.6 Complexity Order

The complexity order of the schemes proposed in this chapter is similar to the complexity orders presented in the previous chapter. Here we have introduced MSF besides ISF (which its complexity order was discussed in Section 4.6) and constellation rotation. The decoding model of MSF is similar to ISF with the decoding/encoding of the main source first, and later the decoding of the interfering source data.

The Raptor decoder complexity order, as has already been discussed in Section 4.6, is $O(n^2)$ in terms of codeword length. We will not discuss the main source decoder, since it was present in the link before the addition of the interfering source. However, the re-encoding block of the main source channel code is new, and hence the computational complexity is increased. The complexity order changes based on what kind of channel code is used for the main channel. In case of DVB-RCS, the decoding and encoding complexity of Turbo code has been shown to be linear [106].

As for the constellation rotation, although it increases the complexity to some extent, it does not have any effect on the complexity order since it is just multiplication of each modulated symbol by a fixed phase rotation before transmission. This adds only a fixed computational complexity to each symbol of the whole message.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we investigated various methods for improving the bandwidth efficiency of MAC networks. These methods vary in terms of bandwidth utilization, complexity, and channel orthogonality.

In Chapter 3, two schemes were proposed. One relies on iterative decoding where the destination node iteratively decodes data received from the two sources and the relay. This scheme was based on LLRs that were exchanged iteratively by the three decoders at the destination. The second scheme uses the relay to increase the rate and sends parity bits which were missed in the original broadcasts. In other words, each source shortens its codeword and sends it to the relay and the destination. The relay sends the combination of missing parts to the destination so that it can have the whole codeword and decode it.

Both schemes reduce the error probability of decoding when one of the sources is weaker than the other one, and performance increases as this gap increases. Relay does not need to know the CSI between sources and destination, and blindly helps the weaker one, i.e. although the relay is sending parity bits for both of the sources, it is

helping the destination to decide which source needs parity to extracts its codeword from the data transmitted by the relay.

In Chapter 4, we proposed the addition of an interfering channel to an already-existing main channel without affecting the decoding performance of the main source while the channels are non-orthogonal. The interfering source uses Raptor codes. We assume that the power level of the interfering source is higher than or equal to the power level of the main source, and at the destination successive decoding is used. In successive decoding, first the interfering channel signal and later the signal from the main channel are decoded. In the case of equal powers for both of the sources, a hard decision stage is proposed prior to the decoding phase in order to eliminate the misleading erased data.

In the case of sources with unequal power levels, we demonstrated that there is an optimum power allocation scenario from a power efficiency-rate point of view. We also proposed a power adaptation scheme which uses the feedback channel of the Raptor code, estimates the channel state of the main channel, and chooses the optimum power level for the interfering source accordingly. With this scheme, the interfering source can adapt itself to the existing link efficiently without having any direct access to the CSI of the main channel. Furthermore, this scheme is robust to any quality change in either channel and adapts itself accordingly.

Our proposed method can be used in many wireless and satellite broadcasting and communication systems, including DVB-S2 and DVB-RCS. An interesting aspect of our proposed scheme is that it can be added to the already existing DVB-RCS hardware without requiring any modification to the internal circuitry. We have simulated both equal and unequal powers scenarios for DVB-RCS with MPEG profile. In both schemes, the simulation results showed acceptable achievable rates around 89% and 76% of the constellation constrained capacity for the equal and unequal power scenarios, respectively.

Finally, in Chapter 5 we investigated the addition of an extra interfering source to an existing DVB channel. The destination uses successive decoding to decode both sources and can decode either the main source or the interfering source first. As discussed in previous chapter, by decoding the interfering source first, we achieve near-capacity rates for the interfering source without affecting the decoding performance of the main source. However, this introduces a delay in decoding the main source. Here, we proposed decoding the main source data first and demonstrated that, although decoding the main channel first eliminates this delay, it also diminishes the decoding performance of the main source. Nevertheless, in specific power allocation scenarios, the performances of the two schemes are close and the main source can be decoded first, instead of the interfering source. This eliminates the delay without any major loss in performance.

Meanwhile, when the difference between the power levels of the two sources is small, symbols from the sources may cancel each other and reduce the achievable rate in both schemes. To address this problem we have suggested constellation rotation. There is an optimum rotation angle for any power allocation scenario, and if it is not practical to adjust the rotation angle for each particular power allocation scenario, an average optimal rotation angle that on averages, delivers the highest achievable rate can be found.

6.2 Future Work

Both of the proposed ideas, the collaborative scheme in Chapter 3 and the interfering source scheme in Chapters 4 and 5, can be developed beyond this thesis.

In the collaborative scheme, one of the ideas that can be pursued further is finding the upper or lower bounds for the proposed schemes (much like the bounds calculated in [22]). Finding these bounds gives the proposed scheme an analytic background and

can be useful in demonstrating each scheme's distance from optimum performance.

Furthermore, as mentioned before, the source-relay channels were considered to be noise-free. An extension of this could be the consideration of noisy source-relay channels. This noise which will result in SNR reduction can be modeled by distance too. Therefore the performance of the system can be tested against different relay positions. Meanwhile, in light of recent achievements in PNC [10], this idea can be used in the relays and with less decoding complexity.

As for the scheme of introducing an interfering source to an existing scheme, one possibility is extending the work to higher modulations. The current scheme uses QPSK and the received constellation at the destination, if no symbol cancellation has happened, is a 16-point constellation. However, this 16-point constellation is not a standard 16PSK or 16QAM. The transmitting constellations can be chosen so that the final received constellation is a standard map.

For example, assume that the main source is transmitting with 4QAM and the interfering source chooses a 16QAM constellation, so that at the destination a 64QAM map is received. To achieve this constellation map, the interfering source should adjust its power according to the power level of the main source. The benefit of this method is additional routine demodulation at the destination node.

Furthermore, in this work, we assumed that the main and interfering sources data pass an AWGN channel and that there is no fading. An interesting topic for further research is the investigation of the fading effect on the proposed scheme. Fading effects have both advantages and disadvantages for this system. Since the fading always changes the phase of the symbols, the constellation rotation which was proposed will happen automatically in a fading channel, and will lower the encoding complexity at the interfering source. However, besides the phase, the fading channel alters the power level of the symbols too.

This means that if the power levels of the two sources are not significantly higher

or lower that one another, the weak signal may become the stronger one at the destination; i.e., if the main source was transmitting with higher power, due to the fading effect, we receive interfering source data with higher power at the destination. This will change the whole decoding scheme.

Finally, another appealing idea would be to study the use of distributed data transmission under this scheme. This means that both sources have access to the same data, but each encodes parts of it with a different redundancy. The main source encodes the higher-priority data and uses higher redundancy, and the interfering source encodes the non-crucial details of data with lower redundancy. At the destination, the data from main source is decoded first and gives the most significant pieces of data. Later, if the data from interfering source can be decoded, the destination will have a better image of the original data as a whole.

6.3 Publications

The following is a list of publications that are based on the results presented in this thesis:

- M. J. Hagh and M. R. Soleymani, "Constellation Rotation for DVB Multiple Access Channels with Raptor coding,"IEEE Transactions on Broadcasting, March 2013.

- M. J. Hagh and M. R. Soleymani, "Application of Raptor Coding with Power Adaptation to DVB Multiple Access Channels," IEEE Transactions on Broadcasting, vol.58, no.3, pp.379-389, Sept. 2012.

- M. J. Hagh and M. R. Soleymani, "Power Adaptation for DVB Multiple Access Channel with Raptor Code," In Proc. IEEE Global Telecomm. Conf. (GLOBECOM), pages 16, Houston, TX, December 2011.

– M. J. Hagh and M. R. Soleymani, "Raptor coding for Non-Orthogonal multiple access channels," In Proc. IEEE Int. Conf. Commun. (ICC), pages 16, Kyoto, Japan, June 2011.

– M. J. Hagh and M. R. Soleymani, "Novel Techniques in Cooperative Wireless Networks using Network Coding," in Proc. IEEE Canadian Conf. on Electrical and Computer Engineering (CCECE), pages 557-562, Niagara Falls, Ontario, Canada, May 2011.

Bibliography

[1] T. Ho S. Ray D. R. Karger M. Effros, M. Medard and R. Koetter. Linear network codes: A unified framework for source, channel and network coding. In *Proc. DIMACS Workshop on Network Information Theory*, March 2003.

[2] C. Hausl and J. Hagenauer. Iterative network and channel decoding for the two-way relay channel. In *Proc. IEEE Int. Conf. Commun. (ICC)*, volume 4, pages 1568–1573, June 2006.

[3] C. Hausl, F. Schreckenbach, I. Oikonomidis, and Bauch G. Iterative network and channel decoding on a tanner graph. In *Proc. 43rd Allerton Conf. on Communication, Control, and Computing*, September 2005.

[4] N. Sarshar and Xiaolin Wu. A practical approach to joint network-source coding. In *Proc. Data Compression Conf.*, pages 93–102, March 2006.

[5] J. Kliewer, T. Dikaliotis, and T. Ho. On the performance of joint and separate channel and network coding in wireless fading networks. In *Proc. IEEE Inf. Theory Workshop on Inf. Theory for Wireless Networks*, pages 1–5, July 2007.

[6] Xingkai Bao and Jing Li. Matching code-on-graph with network-on-graph: Adaptive network coding for wireless relay networks. In *Proc. 43rd Allerton Conf. on Communication, Control, and Computing*, September 2005.

[7] Xingkai Bao and Jing Li. An information theoretic analysis for adaptive-network-coded-cooperation (ancc) in wireless relay networks. In *Proc. IEEE Int. Symp. Info. Theory (ISIT)*, pages 2719–2723, July 2006.

[8] Xingkai Bao and Jing Li. A unified channel-network coding treatment for user cooperation in wireless ad-hoc networks. In *Proc. IEEE Int. Symp. Info. Theory (ISIT)*, pages 202–206, July 2006.

[9] Xingkai Bao and Jing Li. On the outage properties of adaptive network coded cooperation (ancc) in large wireless networks. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages IV–IV, May 2006.

[10] Shengli Zhang and Soung-Chang Liew. Channel coding and decoding in a relay system operated with physical-layer network coding. *IEEE Journal Selected Areas in Commun.*, 27(5):788–796, June 2009.

[11] Shengli Zhang, Yu Zhu, Soung-Chang Liew, and Khaled Ben Letaief. Joint design of network coding and channel decoding for wireless networks. In *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC)*, pages 779–784, March 2007.

[12] Shengli Zhang, Soung Chang Liew, and Patrick P. Lam. Hot topic: physical-layer network coding. In *Proc. 12th Int. Conf. on Mobile computing and networking (MobiCom)*, pages 358–365, New York, NY, USA, 2006.

[13] Shengli Zhang, Soung Chang Liew, and Lu Lu. Physical layer network coding schemes over finite and infinite fields. In *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, pages 1–6, Dec. 2008.

[14] Chun-Hung Liu and A. Arapostathis. Joint network coding and superposition coding for multi-user information exchange in wireless relaying networks. In *Proc. IEEE Global Telecommunications Conf.*, pages 1–6, Dec. 2008.

[15] Moonseo Park and Seong-Lyun Kim. A minimum mean-squared error relay for the two-way relay channel with network coding. *IEEE Communications Letters*, 13(3):196–198, March 2009.

[16] Lei Xiao, T. Fuja, J. Kliewer, and D. Costello. A network coding approach to cooperative diversity. *IEEE Trans. Inf. Theory*, 53(10):3714–3722, Oct. 2007.

[17] J. Del Ser, P.M. Crespo, B.H. Khalaj, and J. Gutierrez-Gutierrez. On combining distributed joint source-channel-network coding and turbo equalization in multiple access relay networks. In *Proc. Third IEEE Int. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMOB)*, pages 18–18, Oct. 2007.

[18] Suhua Tang, Jun Cheng, Chen Sun, R. Suzuki, and S. Obana. Turbo network coding for efficient and reliable relay. In *Proc. IEEE Singapour Int. Conf. on Communication Systems*, pages 1603–1608, Nov. 2008.

[19] P. Popovski and H. Yomo. Bi-directional amplification of throughput in a wireless multi-hop network. In *Proc. IEEE Vehicular Technology Conference (VTC)*, volume 2, pages 588–593, May 2006.

[20] D.H. Woldegebreal and H. Karl. Multiple-access relay channel with network coding and non-ideal source-relay channels. In *Proc. 4th Int. Symp. on Wireless Commun. Systems (ISWCS)*, pages 732–736, Oct. 2007.

[21] J.N. Laneman. Network coding gain of cooperative diversity. In *Proc. IEEE Military Communications Conference (MILCOM)*, volume 1, pages 106–112, Nov. 2004.

[22] P. Popovski and H. Yomo. Physical network coding in two-way wireless relay channels. In *Proc. IEEE Int. Conf. Commun. (ICC)*, pages 707–712, June 2007.

[23] M.J. Hagh and M.R. Soleymani. Novel techniques in cooperative wireless networks using network coding. In *Proc. 24th Canadian Conf. on Electrical and Computer Engineering (CCECE)*, pages 557 –562, May 2011.

[24] Sang Wu Kim. Concatenated random parity forwarding in multi-source multihop wireless networks. In *Proc. IEEE Inf. Theory Workshop (ITW)*, pages 102–107, Sept. 2007.

[25] M. Janani, A. Hedayat, T.E. Hunter, and A. Nosratinia. Coded cooperation in wireless communications: space-time transmission and iterative decoding. *IEEE Trans. Signal Processing*, 52(2):362–371, Feb. 2004.

[26] Xingkai Bao and Jing Li. Progressive network coding for message-forwarding in ad-hoc wireless networks. In *Proc. 3rd IEEE Commun. Society Conf. on Sensor and Ad Hoc Commun. and Networks (SECON)*, volume 1, pages 207–215, Sept. 2006.

[27] Su Kiang Kuek, Chau Yuen, and Woon Hau Chin. Four-node relay network with bi-directional traffic employing wireless network coding with pre-cancellation. In *Proc. IEEE Vehicular Technology Conference (VTC)*, pages 1201–1205, May 2008.

[28] Cong Peng, Qian Zhang, Ming Zhao, and Yan Yao. Opportunistic network-coded cooperation in wireless networks. In *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC)*, pages 3358–3363, March 2007.

[29] Cong Peng, Qian Zhang, Ming Zhao, and Yan Yao. Sncc: A selective network-coded cooperation scheme in wireless networks. In *Proc. IEEE Int. Conf. Commun. (ICC)*, pages 4219–4224, June 2007.

[30] H. Yomo and P. Popovski. Opportunistic scheduling for wireless network coding. In *Proc. IEEE Int. Conf. Commun. (ICC)*, pages 5610–5615, June 2007.

[31] Sang Joon Kim, P. Mitran, and V. Tarokh. Performance bounds for bi-directional coded cooperation protocols. In *Proc. 27th IEEE Int. Conf. on Distributed Computing Systems Workshops (ICDCSW)*, pages 83–83, June 2007.

[32] T. Koike-Akino, P. Popovski, and V. Tarokh. Two-way relaying with network coding for frequency-selective fading channels. In *Proc. 42nd Asilomar Conf. on Signals, Systems and Computers*, pages 2221–2225, Oct. 2008.

[33] T. Koike-Akino, P. Popovski, and V. Tarokh. Denoising maps and constellations for wireless network coding in two-way relaying systems. In *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, pages 1–5, Dec. 2008.

[34] Bo Yang, Hongyi Yu, Ling Lv, and Qiang Feng. Spectral efficient cooperative relaying techniques via physical network coding. In *Proc. 9th Int. Conf. on Signal Processing (ICSP)*, pages 2812–2815, Oct. 2008.

[35] Ling Lv and Hongyi Yu. Spectrally-efficient successive wireless relaying based on network coding. In *Proc. 4th Int. Conf. on Wireless Communications, Networking and Mobile Computing (WiCOM)*, pages 1–4, Oct. 2008.

[36] Yong Ho Kim, Nan Sol Seo, and Young Yong Kim. A network coding based multicasting (netcom) over ieee 802.11 multi-hop. In *Proc. IEEE Vehicular Technology Conference (VTC)*, pages 845–848, April 2007.

[37] Jia-Qi Jin, T. Ho, and H. Viswanathan. Comparison of network coding and non-network coding schemes for multi-hop wireless networks. In *Proc. IEEE Int. Symp. Info. Theory (ISIT)*, pages 197–201, July 2006.

[38] Y.E. Sagduyu and A. Ephremides. Crosslayer design for distributed mac and network coding in wireless ad hoc networks. In *Proc. IEEE Int. Symp. Info. Theory (ISIT)*, pages 1863–1867, Sept. 2005.

[39] Y.E. Sagduyu and A. Ephremides. On joint mac and network coding in wireless ad hoc networks. *IEEE Trans. Inf. Theory*, 53(10):3697–3713, Oct. 2007.

[40] A.A. Hamra, C. Barakat, and T. Turletti. Network coding for wireless mesh networks: a case study. In *Proc. Int. Symp. on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 9–114, 2006.

[41] S. Katti, H. Rahul, Wenjun Hu, D. Katabi, M. Medard, and J. Crowcroft. Xors in the air: Practical wireless network coding. *IEEE/ACM Trans. Networking*, 16(3):497–510, June 2008.

[42] C. Fragouli, J. Widmer, and J.-Y. le Boudec. On the benefits of network coding for wireless applications. In *Proc. 4th Int. Symp. on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 1–6, April 2006.

[43] Jingyao Zhang and Pingyi Fan. On network coding in wireless ad-hoc networks. In *Proc. 2nd Int. Conf. on Mobile Technology, Applications and Systems*, pages 8 pp.–8, Nov. 2005.

[44] Wei Pu, Chong Luo, Shipeng Li, and Chang Wen Chen. Continuous network coding in wireless relay networks. In *Proc. 27th IEEE Int. Conf. on Computer Communications (INFOCOM)*, pages 1526–1534, April 2008.

[45] Sichao Yang and Ralf Koetter. Network coding over a noisy relay : a belief propagation approach. In *Proc. IEEE Int. Symp. Info. Theory (ISIT)*, pages 801–804, June 2007.

[46] Tairan Wang and G.B. Giannakis. High-throughput cooperative communications with complex field network coding. In *Proc. 41st Conf. on Information Sciences and Systems (CISS)*, pages 253–258, March 2007.

[47] E. Fasolo, F. Rossetto, and M. Zorzi. Network coding meets mimo. In *Proc. Fourth Workshop on Network Coding, Theory and Applications (NetCod)*, pages 1–6, Jan. 2008.

[48] H. Wicaksana, See Ho Ting, and Yong Liang Guan. Spectral efficient half duplex relaying for fountain code with wireless network coding. In *Proc. IEEE Int. Conf. on Communications Workshops (ICC Workshops)*, pages 295–299, May 2008.

[49] Yingda Chen, S. Kishore, and Jing Li. Wireless diversity through network coding. In *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC)*, volume 3, pages 1681–1686, April 2006.

[50] E. Kurniawan, S. Sun, K. Yen, and K.F.E. Chong. Application of network coding in rateless transmission over wireless relay networks. *IEEE Trans. Commun.*, 59(2):507–517, February 2011.

[51] B.N. Vellambi, N. Rahnavard, and F. Fekri. Fts: A distributed energy-efficient broadcasting scheme using fountain codes for multihop wireless networks. *IEEE Trans. Commun.*, 58(12):3561–3572, December 2010.

[52] Momin Uppal, Anders Host-Madsen, and Zixiang Xiong. Practical rateless cooperation in multiple access channels using multiplexed raptor codes. In *Proc. IEEE Int. Symp. Info. Theory (ISIT)*, pages 671–675, Nice, France, June 2007.

[53] Zigui Yang and A. Host-Madsen. Rateless coded cooperation for multiple-access channels in the low power regime. In *Proc. IEEE Int. Symp. Info. Theory (ISIT)*, pages 967 –971, Seattle, WA, July 2006.

[54] M. Uppal, Zigui Yang, A. Host-Madsen, and Zixiang Xiong. Cooperation in the low power regime for the mac using multiplexed rateless codes. *IEEE Trans. Signal Processing*, 58(9):4720–4734, Sept. 2010.

[55] M. Uppal, Guosen Yue, Xiaodong Wang, and Zixiang Xiong. A rateless coded protocol for half-duplex wireless relay channels. *IEEE Trans. Signal Processing*, 59(1):209 –222, Jan. 2011.

[56] O.Y. Bursalioglu, M. Fresia, G. Caire, and H.V. Poor. Lossy multicasting over binary symmetric broadcast channels. *IEEE Trans. Signal Processing*, 59(8):3915 –3929, Aug. 2011.

[57] D. Sejdinovic, R. Piechocki, A. Doufexi, and M. Ismail. Fountain code design for data multicast with side information. *IEEE Tran. Wireless Commun.*, 8(10):5155 –5165, Oct. 2009.

[58] C. Gong, G. Yue, and X. Wang. Analysis and optimization of a rateless coded joint relay system. *IEEE Tran. Wireless Commun.*, 9(3):1175 –1185, March 2010.

[59] A. Ravanshid, L. Lampe, and J. Huber. Signal combining for relay transmission with rateless codes. In *Proc. IEEE Int. Symp. Info. Theory (ISIT)*, pages 508 –512, July 2009.

[60] M.J. Hagh and M.R. Soleymani. Raptor coding for non-orthogonal multiple access channels. In *Proc. IEEE Int. Conf. Commun. (ICC)*, pages 1–6, Kyoto, Japan, June 2011.

[61] M.J. Hagh and M.R. Soleymani. Power adaptation for dvb multiple access channel with raptor code. In *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, pages 1–6, Houston, TX, December 2011.

[62] M. J. Hagh and M. R. Soleymani. Application of raptor coding with power adaptation to dvb multiple access channels. *IEEE Trans. Broadcasting*, 58(3):379 –389, Sept. 2012.

[63] C. Gong, A. Tajer, and X. Wang. Interference channel with constrained partial group decoding. *IEEE Trans. Commun.*, 59(11):3059–3071, November 2011.

[64] S. Papaharalabos, D. Benmayor, P.T. Mathiopoulos, and Pingzhi Fan. Performance comparisons and improvements of channel coding techniques for digital satellite broadcasting to mobile users. *IEEE Trans. Broadcasting*, 57(1):94–102, March 2011.

[65] S. Papaharalabos, M. Papaleo, P.T. Mathiopoulos, M. Neri, A. Vanelli-Coralli, and G.E. Corazza. Dvb-s2 ldpc decoding using robust check node update approximations. *IEEE Trans. Broadcasting*, 54(1):120–126, March 2008.

[66] T. Mladenov, S. Nooshabadi, and Kiseon Kim. Efficient incremental raptor decoding over bec for 3gpp mbms and dvb ip-datacast services. *IEEE Trans. Broadcasting*, 57(2):313–318, June 2011.

[67] Michael Luby, Tiago Gasiba, Thomas Stockhammer, and Mark Watson. Reliable multimedia download delivery in cellular broadcast networks. *IEEE Trans. Broadcasting*, 53(1):235–246, March 2007.

[68] T. Ho, M. Medard, R. Koetter, D.R. Karger, M. Effros, Jun Shi, and B. Leong. A random linear network coding approach to multicast. *IEEE Trans. Inf. Theory*, 52(10):4413–4430, Oct. 2006.

[69] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *IEEE Trans. Inf. Theory*, 46(4):1204–1216, Jul 2000.

[70] S.-Y.R. Li, R.W. Yeung, and Ning Cai. Linear network coding. *IEEE Trans. Inf. Theory*, 49(2):371–381, Feb. 2003.

[71] R. Koetter and M. Medard. An algebraic approach to network coding. *IEEE/ACM Trans. Networking*, 11(5):782–795, Oct. 2003.

[72] S. Jaggi, P. Sanders, P.A. Chou, M. Effros, S. Egner, K. Jain, and L.M.G.M. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Trans. Inf. Theory*, 51(6):1973–1982, June 2005.

[73] C. Fragouli and E. Soljanin. Information flow decomposition for network coding. *IEEE Trans. Inf. Theory*, 52(3):829–848, March 2006.

[74] April Rasala Lehman and Eric Lehman. Complexity classification of network information flow problems. In *Proc. Fifteenth ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 142–150, Philadelphia, PA, USA, 2004.

[75] Muriel Medard, Michelle Effros, David Karger, and Tracey Ho. On coding for non-multicast networks. In *Proc. 41st Allerton Conf. on Communication, Control and Computing*, May 2003.

[76] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow. *IEEE Trans. Inf. Theory*, 51(8):2745–2759, Aug. 2005.

[77] N. Ratnakar, D. Traskov, and R. Koetter. Approaches to network coding for multiple unicasts. In *Proc. Int. Zurich Seminar on Communications*, pages 70–73, 2006.

[78] T. Ho, Y. Chang, and K. J. Han. On constructive network coding for multiple unicasts. In *Proc. 44th Allerton Conf. on Communication, Control and Computing*, Sept. 2006.

[79] M. Langberg, A. Sprintson, and J. Bruck. Network coding: A computational perspective. *IEEE Trans. Inf. Theory*, 55(1):147–157, Jan. 2009.

[80] Minkyu Kim, M. Medard, V. Aggarwal, U.-M. O'Reilly, Wonsik Kim, Chang Wook Ahn, and M. Effros. Evolutionary approaches to minimizing network coding resources. In *Proc. 26th IEEE Int. Conf. on Computer Communications (INFOCOM)*, pages 1991–1999, May 2007.

[81] D.S. Lun, N. Ratnakar, M. Medard, R. Koetter, D.R. Karger, T. Ho, E. Ahmed, and F. Zhao. Minimum-cost multicast over coded packet networks. *IEEE Trans. Inf. Theory*, 52(6):2608–2623, June 2006.

[82] K. Bhattad, N. Ratnakar, R. Koetter, and K.R. Narayanan. Minimal network coding for multicast. In *Proc. IEEE Int. Symp. Info. Theory (ISIT)*, pages 1730–1734, Sept. 2005.

[83] T. Cover and A.E. Gamal. Capacity theorems for the relay channel. *IEEE Trans. Inf. Theory*, 25(5):572–584, Sep. 1979.

[84] G. Kramer and A.J. van Wijngaarden. On the white gaussian multiple-access relay channel. In *Proc. IEEE Int. Symp. Info. Theory (ISIT)*, page 40, 2000.

[85] G. Kramer, M. Gastpar, and P. Gupta. Cooperative strategies and capacity theorems for relay networks. *IEEE Trans. Inf. Theory*, 51(9):3037–3063, Sept. 2005.

[86] L. Sankaranarayanan, G. Kramer, and N. B. Mandayam. Capacity theorems for the multiple-access relay channel. In *Proc. 42nd Allerton Conf. on Communication, Control, and Computing*, Sept. 2004.

[87] L. Sankaranarayanan, G. Kramer, and N.B. Mandayam. Hierarchical sensor networks: capacity bounds and cooperative strategies using the multiple-access relay channel model. In *Proc. First IEEE Commun. Society Conf. on Sensor and Ad Hoc Commun. and Networks (SECON)*, pages 191–199, Oct. 2004.

[88] M. Luby. Lt codes. In *Proc. 43rd IEEE Symp. Foundations of Computer Science (FOCS)*, pages 271–280, Vancouver, BC, Canada, Nov. 2002.

[89] A. Shokrollahi. Raptor codes. *IEEE Trans. Inf. Theory*, 52(6):2551–2567, June 2006.

[90] T. D. Nguyen, L. L. Yang, and L. Hanzo. Systematic luby transform codes and their soft decoding. In *Proc. IEEE Workshop on Signal Processing Systems*, pages 67–72, Shanghai, China, Oct. 2007.

[91] Hrvoje Jenkac and Timo Mayer. Soft decoding of lt-codes for wireless broadcast. In *Proc. IST Mobile Summit 2005*, Dresden, Germany, June 2005.

[92] O. Etesami and A. Shokrollahi. Raptor codes on binary memoryless symmetric channels. *IEEE Trans. Inf. Theory*, 52(5):2033–2051, May 2006.

[93] J. Hamkins. Performance of low-density parity-check coded modulation. In *Proc. IEEE Aerospace Conf.*, pages 1–14, Big Sky, MT, March 2010.

[94] Etsi en 302 307 - digital video broadcasting (dvb); second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications (dvb-s2), August 2009.

[95] Etsi en 301 790 - digital video broadcasting (dvb); interaction channel for satellite distribution systems (dvb-rcs), May 2009.

[96] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[97] Jin Lu and J.M.F. Moura. Linear time encoding of ldpc codes. *IEEE Trans. Inf. Theory*, 56(1):233 –249, Jan. 2010.

[98] J. Chen, A. Dholakia, E. Eleftheriou, M.P.C. Fossorier, and X.-Y. Hu. Reduced-complexity decoding of ldpc codes. *IEEE Trans. Commun.*, 53(8):1288 – 1299, Aug. 2005.

[99] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 2nd edition, 1991.

[100] J. Harshan and B. S. Rajan. On two-user gaussian multiple access channels with finite input constellations. *IEEE Trans. Inf. Theory*, 57(3):1299–1327, March 2011.

[101] T. Kasami and Shu Lin. Coding for a multiple-access channel. *IEEE Trans. Inf. Theory*, 22(2):129–137, March 1976.

[102] Masoud Salehi John G. Proakis. *Digital Communications*. McGraw-Hill, New York, NY, USA, 5 edition, 2008.

[103] Anthanasios Papoulis and Pillai S. Unnikrishna. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, New York, NY, 4th edition, 2002.

[104] Mohammad Torabi Konjin. *Multicarrier Systems with Antenna Diversity for Wireless Communications*. PhD thesis, Concordia University, Montreal, Canada, 2004.

[105] Thang Nguyen. An investigation of decoding complexity and coding rate performance of raptor codes. Master's thesis, Concordia University, Montreal, Canada, 2011.

[106] I.A. Chatzigeorgiou, M.R.D. Rodrigues, I.J. Wassell, and R.A. Carrasco. Comparison of convolutional and turbo coding for broadband fwa systems. *IEEE Trans. Broadcasting*, 53(2):494 –503, June 2007.